

UNIVERSIDAD DE CÓRDOBA



**ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE INGENIERÍA RURAL**

TESIS DOCTORAL

OPTIMIZACIÓN DE PROBLEMAS DE DISTRIBUCIÓN EN PLANTA MEDIANTE ALGORITMOS EVOLUTIVOS

Directores:

**Laura García Hernández
Lorenzo Salas Morera**

Autor:

José Antonio García Hernández

Córdoba, Mayo de 2020

TITULO: *OPTIMIZACIÓN DE PROBLEMAS DE DISTRIBUCIÓN EN PLANTA
MEDIANTE ALGORITMOS EVOLUTIVOS*

AUTOR: *José Antonio García Hernández*

© 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

UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE INGENIERÍA RURAL

OPTIMIZACIÓN DE PROBLEMAS DE DISTRIBUCIÓN EN PLANTA MEDIANTE ALGORITMOS EVOLUTIVOS

Tesis Doctoral presentada por José Antonio García Hernández, en satisfacción de los requisitos necesarios para optar al grado de Doctor, y dirigida por los Profesores Doctores: Laura García Hernández y Lorenzo Salas Morera, de la Universidad de Córdoba, España.

El Doctorando:

Fdo.: José Antonio García Hernández

Córdoba, Mayo de 2020

Informe justificado de los Directores de la Tesis:

En su tesis, José Antonio García Hernández, ha abordado el diseño de distribución en planta, en particular, se ha centrado en la resolución del problema de distribución en planta de áreas desiguales. Para solucionar este problema, José Antonio ha estudiado y analizado el estado del arte de la multitud de propuestas existentes que resuelven el problema de distribución en planta de áreas desiguales. Después, de hacer un estudio exhaustivo de la literatura del problema, ha identificado que la mayoría de las propuestas existentes buscan el mejor diseño de la planta industrial considerando el coste del flujo de material como principal criterio a optimizar, ya que, está directamente relacionado con la reducción de los costes totales. Además, para hacer que el diseño sea factible, hay que considerar que cumple con ciertas restricciones como son que no está permitido el solapamiento entre los diferentes departamentos que componen la planta industrial, y, además, que todos los departamentos que integran la planta tienen que cumplir con una relación de aspecto. Por todo lo expuesto, en una primera instancia, José Antonio se ha centrado en buscar diseños industriales que permitan reducir el coste del flujo de material. Para ello, ha aplicado al problema de distribución en planta de áreas desiguales, una novedosa y reciente metaheurística que se basa en el comportamiento que tienen los arrecifes de corales marinos para buscar soluciones del problema. Su propuesta ha sido probada de forma extensiva con problemas de referencia tomados de la bibliografía. Con la experimentación realizada, se ha demostrado que el uso de algoritmos de arrecifes de coral son muy efectivos cuando se aplican a la resolución del problema de distribución en planta de áreas desiguales, ya que, mediante esta estrategia es posible encontrar diseños que ofrecen mejores resultados que las propuestas existentes hasta el momento. En una segunda instancia, José Antonio ha ido un paso más allá en la estrategia de optimización de arrecifes de coral, proponiendo un modelo de islas de arrecifes de coral que realiza una paralelización del algoritmo inicial y permite evolucionar diferentes poblaciones de arrecifes de coral al mismo tiempo. Mediante esta mejora se consigue por un lado, aumentar la diversidad de las soluciones del problema, y por otro, encontrar soluciones que mejoran aun más la aptitud en términos de coste de flujo de material.

Bajo nuestro mejor saber y conocimiento, ninguna de estas propuestas han sido aplicadas anteriormente al problema de distribución en planta, siendo el modelo de islas de arrecifes de coral, una nueva metaheurística creada totalmente original. Ya que hasta ese momento, no existía ninguna propuesta paralelizada del algoritmo de optimización basado en arrecifes de coral. Por lo que el trabajo de José Antonio ha contribuido de una manera muy considerable en el estado del arte del problema de distribución en planta de áreas desiguales y también en el campo de la computación evolutiva y las metaheurísticas. Como resultado de esta exitosa investigación, se han publicado sus resultados en tres artículos de investigación publicados en dos revistas de impacto altamente posicionadas en su área (cuartil 1) y una conferencia internacional, obteniendo en ella un premio a la mejor contribución.

Cabe destacar que, José Antonio ha demostrado una gran capacidad de trabajo y cualidades científicas en labores de investigación, proponiendo ideas nuevas y formas

diferentes y efectivas de resolver las problemáticas encontradas. Ha demostrado grandes habilidades en la programación y computación de los algoritmos de optimización diseñados y desarrollados. José Antonio, siempre ha tenido una gran disposición y motivación para afrontar el trabajo de investigación realizado en esta tesis, contribuyendo de una manera muy positiva y fructífera en la línea de investigación desarrollada por sus directores.

Por todo lo expuesto anteriormente, José Antonio es autorizado a presentar su trabajo de investigación con la finalidad de obtener el título de Doctor por la Universidad de Córdoba.

En Córdoba, Abril de 2020.

Fdo. Dra. Laura García Hernández

Fdo. Dr. Lorenzo Salas Morera

A mis padres, mi hermana y mi sobrino.

Agradecimientos

En primer lugar deseo expresar mi más sincero agradecimiento a mis directores de tesis. Gracias Lorenzo, por tus sabios consejos y tu apoyo. Gracias Laura, por tu esfuerzo, tu dedicación, tu magisterio, tu talento, por hacer que esta tesis sea posible. Gracias a los dos.

En segundo lugar deseo agradecer a todos mis profesores, jefes y compañeros que me aportaron cosas buenas y por haber contribuido a mi formación académica, laboral y humana. Gracias.

En tercer lugar, agradecer a mis amigos, los presentes y a los que ya se fueron, por haber compartido conmigo todo tipo de momentos, buenos y malos, por su apoyo y sabios consejos. Gracias especialmente a los que ya no están.

Y en último lugar agradecer a mi familia, por ser el pilar de mi vida. Gracias a mis padres, por haber luchado para que este momento sea posible. Gracias sobre todo a mi hermana, por ser mi mejor amiga, por ser esa compañera inseparable a la que le debo tanto. Gracias por estar ahí siempre. Gracias a mi sobrino, por hacerme sentir padre cada día.

A todos ellos, mil gracias.

*“Converso con el hombre que siempre va conmigo
--quien habla solo espera hablar a Dios un día--;
mi soliloquio es plática con este buen amigo
que me enseñó el secreto de la filantropía.”.*

Antonio Machado

Resumen

Este trabajo de investigación acomete el problema de distribución en planta. De forma resumida, este problema comprende la distribución de los diferentes departamentos que integran una planta industrial de la forma más satisfactoria posible teniendo en cuenta ciertos criterios y restricciones. Dependiendo de las características del problema, pueden originarse multitud de taxonomías o subproblemas de distribución en planta. En esta tesis doctoral, se abordará el problema de distribución en planta de áreas desiguales que ha sido uno de los más estudiados. Para resolver este problema de distribución en planta de áreas desiguales (UA-FLP en inglés), han sido utilizadas multitud de propuestas con el objetivo de obtener el diseño más satisfactorio de la planta industrial. En este sentido, los algoritmos evolutivos han sido ampliamente utilizados en la bibliografía. Por otro lado, dentro de los posibles criterios a considerar cuando se resuelve el problema de distribución en planta, el coste de flujo de material ha sido el más empleado, ya que está directamente relacionado con el coste total de una planta industrial. Es por esta razón que esta tesis doctoral pretende resolver el problema de distribución en planta teniendo en cuenta el criterio del coste de flujo de material, con el objetivo de obtener mejores soluciones que las existentes hasta el momento en la bibliografía de referencia. Para ello, se ha empleado una novedosa y reciente metaheurística que se basa en el comportamiento existente en los arrecifes de corales marinos. Esta nueva metaheurística ha sido empleada con mucho éxito en diferentes problemas complejos de optimización, logrando obtener unos resultados muy satisfactorios en diferentes ámbitos y áreas. Este algoritmo de optimización basado en algoritmos de arrecifes de coral ha sido aplicado al problema de distribución en planta de áreas desiguales considerando el coste de flujo de material como

criterio de optimización. La aplicación de esta propuesta es una contribución totalmente original al problema de distribución en planta, ya que, hasta el momento no había sido probado en este campo. La propuesta de optimización basada en los arrecifes de coral ha sido probada de forma empírica con multitud de problemas de referencia de la bibliografía de diferente complejidad. Como resultado se ha mejorado las soluciones existentes hasta el momento en la mayoría de los casos probados.

Por otro lado, con el objetivo de dar más diversidad a la población y para evitar que el algoritmo caiga en óptimos locales, se ha propuesto una mejora sobre esta metaheurística que se basa en un modelo de islas de arrecifes de coral, lo que permite realizar una paralelización del algoritmo inicial y así, evolucionar diferentes poblaciones de arrecifes de coral al mismo tiempo. Se ha realizado una experimentación empírica con multitud de problemas de referencia de la bibliografía que ha permitido validar este nuevo enfoque bioinspirado, ofreciendo como resultado mejoras sobre las soluciones existentes hasta el momento en referencia a la mayoría de los casos probados (incluso mejores soluciones que las obtenidas por la propuesta inicial de algoritmo de arrecifes de coral). Mediante este nuevo modelo de islas de arrecifes de coral, se consigue también aumentar la diversidad de las soluciones del problema, lo que permite encontrar nuevas soluciones con mejores aptitudes en términos de coste de flujo de material y en menor tiempo de cómputo.

Este nuevo modelo de islas de arrecifes de coral, es una nueva metaheurística que ha sido creada en esta investigación y es totalmente original. Ya que hasta ese momento, no existía ninguna propuesta paralelizada del algoritmo de optimización basado en arrecifes de coral. Por lo que, este nuevo modelo ha contribuido de

una manera muy considerable en el estado del arte del problema de distribución en planta de áreas desiguales y también en el ámbito de la computación evolutiva y las metaheurísticas.

Abstract

This research work tackles the facility layout problem, in summary, this problem includes the distribution of the different departments that make up an industrial plant in the most satisfactory way possible, taking into account certain criteria y restrictions. Depending on the characteristics of the problem, a multitude of facility layout taxonomies or subproblems can arise. In this doctoral thesis, the unequal area facility layout problem is addressed, which has been one of the most studied in the related references. To solve the unequal area facility layout problem (UA-FLP), many proposals have been used to obtain the most satisfactory design of the industrial plant. In this sense, evolutionary algorithms have been the most used in the literature. On the other hand, among the possible criteria to consider when solving the unequal area facility layout problem, the cost of material flow has been the most employed, since it is directly related to the total cost of an industrial plant. This is the reason why this doctoral thesis aims to solve the unequal area facility layout problem taking into account the criterion of the cost of material flow, intending to obtain better solutions than the consequences so far in the reference bibliography. For this, a new y recent metaheuristic has been used that is based on the behaviour existing in the marine coral reefs. This new metaheuristic has been used with great success in different complex optimization problems, achieving very satisfactory results in different fields y areas. This optimization algorithm based on coral algorithms has been applied to the unequal area facility layout problem by

considering the cost of material flow as an optimization criterion. The application of this proposal is a totally original contribution to the facility layout problem, since, until now, it had not been tested in this field. The optimization proposal based on coral reefs has been empirically tested with a multitude of bibliographic reference problems of different complexity. As a result, the solutions improved so far have been improved in the references in most of the cases tested.

Finally, to give more diversity to the population y to avoid the algorithm falling into local optimums, an improvement has been proposed on this metaheuristic that is based on a model of coral reef islands, which allows parallelization of the initial algorithm y thus, evolve different coral reef populations at the same time. Empirical experimentation with a multitude of bibliographic benchmark problems was carried out to validate this new bioinspired approach, y it has resulted in improvements over the solutions that have existed so far in the references in the majority of cases tested (even better solutions than ones obtained by the initial proposal of the coral reefs optimization algorithm). Through this new model of coral reef islands, it is also possible to increase the diversity of the solutions to the problem, allowing to find new designs with better skills in terms of material flow cost y in less computing time.

This new island model of coral reef is a new metaheuristic that has been created in this research y is totally original. Since until then, there was no parallelized proposal for the coral reef-based optimization algorithm. Therefore, this new island model has contributed in a very considerable way in the state of the art of the unequal area facility layout problem, and also, in the evolutionary computation and metaheuristics.

Índice de contenidos

1. Introducción.....	13
2. Objetivos	15
3. Los problemas de distribución en planta y los problemas de distribución en planta de áreas desiguales	17
3.1. <i>Introducción.....</i>	17
3.2. <i>Los problemas de distribución en planta.....</i>	19
3.3. <i>El problema de distribución en planta de áreas desiguales (UA-FLP)..</i>	24
3.3.1 Representación continua en el UA-FLP	25
3.3.2 Representación discreta del UA-FLP.....	27
3.4. <i>Propuestas de resolución del problema de distribución en planta de áreas desiguales.....</i>	28
3.4.1 Procedimientos exactos.....	28
3.4.2 Algoritmos aproximados.....	33
3.4.3 Colonia de hormigas	41
3.4.4 Otras propuestas.....	43
4. Metodología	47
4.1. <i>Algoritmos de optimización de arrecifes de coral.....</i>	47
4.2. <i>Funcionamiento del Algoritmo de Optimización de Arrecifes de Coral</i>	49
4.3. <i>Inicialización del algoritmo.....</i>	49
4.4. <i>Evolución del algoritmo de coral</i>	50
5. Algoritmo de optimización de arrecifes de coral para abordar el UA-FLP	53
5.1. <i>Introducción.....</i>	53
5.2. <i>Enfoque propuesto</i>	54
5.2.1 Codificación del individuo.....	54
5.2.2 Función objetivo	55
5.3. <i>Algoritmo de Optimización basado en Arrecifes de Coral.....</i>	56
5.3.1 Inicialización del algoritmo	57
5.3.2 Evolución del algoritmo de coral.....	59

5.3.3	Experimentación y resultados.....	66
5.4.	<i>Resultados</i>	69
6.	Algoritmo de optimización de arrecifes de coral basado en un modelo de islas para el UA-FLP.....	77
6.1.	<i>Introducción</i>	77
6.2.	<i>Problema de UA-FLP. Definición y antecedentes</i>	79
6.2.1	Definición del problema de UA-FLP	79
6.2.2	Algoritmos paralelos evolutivos.....	80
6.3.	<i>Una nueva metaheurística que combina el modelo de islas y CRO (IMCRO)</i>	81
6.3.1	Algoritmo básico de arrecifes de coral CRO.....	81
6.3.2	Codificación de los corales para el UA-FLP	83
6.3.3	El modelo de islas CRO (IMCRO).....	83
6.3.4	El modelo de islas CRO (IMCRO) extendido.....	86
6.4.	<i>Experimentación y resultados</i>	89
6.4.1	Descripción de los UA-FLPs de prueba	90
6.4.2	Metodología experimental.....	90
6.4.3	Resultados.....	92
7.	Conclusiones finales.....	101
8.	Trabajo futuro.....	105
9.	Referencias	107
A.	Publicaciones.....	131

Índice de tablas

<i>Tabla 1. Comparación de categorías FLP según Komarudin (2009).</i>	21
<i>Tabla 2. Características de los problemas testeados bien conocidos.</i>	68
<i>Tabla 3. Parámetros CRO seleccionados.</i>	68
<i>Tabla 4. Estadísticas de los resultados obtenidos por el algoritmo CRO.</i>	68
<i>Tabla 5. Resumen de los problemas conocidos y sus soluciones FBS.</i>	70
<i>Tabla 6. Comparación de los mejores resultados obtenidos por CRO y otros enfoques alternativos de FBS.</i>	70
<i>Tabla 7. Resumen de los resultados obtenidos por la propuesta CRO.</i>	71
<i>Tabla 8. Características de los UA-FLPs testeados.</i>	90
<i>Tabla 9. Valores usados para el ajuste de los parámetros.</i>	92
<i>Tabla 10. Valores usados dependiendo del tamaño del problema.</i>	92
<i>Tabla 11. Resultados obtenidos por las versiones IMCRO básico y extendido.</i>	93
<i>Tabla 12. Resumen de las soluciones obtenidas por el IMCRO básico vs otros resultados en la literatura.</i>	97
<i>Tabla 13. Resumen de las soluciones obtenidas por el IMCRO extendido vs otros resultados en la literatura.</i>	97
<i>Tabla 14. Resumen de resultados y cromosomas obtenidos por el IMCRO básico.</i> .	98
<i>Tabla 15. Resumen de resultados y cromosomas obtenidos por el IMCRO extendido.</i>	98

Índice de figuras

<i>Figura 1. Clasificación de los problemas de distribución en planta tomada de Drira et al. (2007).</i>	23
<i>Figura 2. Ejemplo de estructura de representación de bahías flexibles (FBS.) tomada de Aiello et al. (2006).</i>	26
<i>Figura 3. Ejemplo de estructura de árbol de corte tomado de Tam (1991).</i>	26
<i>Figura 4. Ejemplo de teoría gráfica tomado de Goetschalckx (1992).</i>	29
<i>Figura 5. Imagen de arrecife de coral.</i>	48
<i>Figura 6. Simulación del arrecife de coral según Salcedo-Sanz et al. (2014); (a) rejilla NxM (b) corales y huecos en el arrecife.</i>	49
<i>Figura 7. Diagrama de flujo del algoritmo CRO. Imagen tomada de Salcedo-Sanz et al. (2014).</i>	50
<i>Figura 8. Ejemplo de un diseño basado en FBS.</i>	54
<i>Figura 9. Ejemplo de un cromosoma con el diseño de una planta.</i>	55
<i>Figura 10. Ejemplo de un arrecife de coral de tamaño 4x4.</i>	57
<i>Figura 11. Ejemplo de un arrecife de coral con individuos aleatorios insertados y $p_0 = 0.5$</i>	58
<i>Figura 12. Diagrama de flujo del algoritmo CRO propuesto</i>	58
<i>Figura 13. Diagrama que muestra la fase de Broadcast spawning</i>	61
<i>Figura 14. Diagrama que muestra el proceso de Crossover en la fase de Broadcast spawning</i>	61
<i>Figura 15. Diagrama que muestra la fase de Brooding.</i>	62
<i>Figura 16. Diagrama que muestra la fase de Larvae setting</i>	64
<i>Figura 17. Diagrama que muestra la fase de Budding.</i>	65
<i>Figura 18. Diagrama que muestra la fase de Depredation.</i>	66
<i>Figura 19. Diagrama que muestra el proceso de mutación y Brooding.</i>	66
<i>Figura 20. El mejor diseño alcanzado por el enfoque CRO propuesto en el Slaughterhouse UA-FLP.</i>	73
<i>Figura 21. El mejor diseño alcanzado por el enfoque CRO propuesto en el CartonPacks UA-FLP.</i>	73
<i>Figura 22. El mejor diseño alcanzado por el enfoque CRO propuesto en el ChoppedPlastic UA-FLP.</i>	73
<i>Figura 23. El mejor diseño alcanzado por el enfoque CRO propuesto en el O9 UA-</i>	

<i>FLP</i>	74
<i>Figura 24. El mejor diseño alcanzado por el enfoque CRO propuesto en el Ba14 UA-FLP</i>	74
<i>Figura 25. Proceso de evolución en algoritmo CRO</i>	82
<i>Figura 26. Ejemplo de estructura de bahías flexibles FBS</i>	83
<i>Figura 27. Proceso de evolución en algoritmo IMCRO</i>	84
<i>Figura 28. Las mejores soluciones obtenidas por el IMCRO extendido para algunas de las instancias probadas de UA-FLP</i>	94
<i>Figura 29. Las mejores soluciones obtenidas por el IMCRO básico para algunas de las instancias probadas de UA-FLP</i>	95
<i>Figura 30. Las mejores soluciones obtenidas por el IMCRO básico para algunas de las instancias probadas de UA-FLP</i>	96

Índice de Símbolos

A_i	-	área de la instalación i .
c_{ij}	-	costo asociado de mover una unidad de flujo de manejo de material entre las instalaciones i y j .
$C(i,j)$	-	medida de separación espacial entre i y j .
$d_{i,j}$	-	distancia Manhattan entre las instalaciones i y j .
D_{inf}	-	es la cantidad de instalaciones que no son factibles.
$f_{i,j}$	-	flujo de cantidad de material entre las instalaciones i y j .
f_r	-	fracción de corales seleccionados en el arrecife r .
F_a	-	fracción de los corales que se eligen en función de fitness que poseen para duplicarse a sí mismos e instalarse de nuevo en el arrecife.
F_b	-	cantidad total de corales seleccionados en el arrecife.
H	-	alto de la planta.
k	-	es un parámetro de penalización.
n	-	numero de instalaciones en la planta.
N	-	alto del arrecife.
M	-	largo del arrecife.
P_d	-	probabilidad de depredación en el arrecife.
$Q(i,j)$	-	medida de interacción entre i y j .
r_i	-	arrecife i .
V_{feas}	-	es el mejor valor de la función objetivo factible que se haya logrado hasta ahora.
V_{all}	-	es en general el mejor valor de la función objetivo (sea factible o no).
W	-	ancho de la planta.
X_k^r	-	coral k en el arrecife r .
Λ_{ij}	-	modelo de arrecife, donde i y j son las coordenadas de un coral
$\Xi(i,j)$	-	coral donde i y j son las coordenadas de la posición en el arrecife.

ρ_0 - proporción entre cuadrados vacíos y en uso que existen en el arrecife que se denota com.

ρ_k - cantidad inicial de corales en el arrecife k .

1. Introducción

El diseño de distribución en planta trata de definir la disposición física más efectiva de las instalaciones (máquinas, equipos de procesamiento y departamentos de servicio...) de una planta y sus diversas partes para lograr la mejor coordinación y eficiencia en el uso del hombre, la máquina y materiales. Mediante el diseño de distribución en planta se pretende cubrir los siguientes objetivos:

- Reducción del costo de manejo de materiales.
- Utilizar el espacio de manera efectiva y eficiente.
- Reducción de la congestión en el proceso productivo.
- Mejorar la capacidad de producción.
- Minimizar el número de accidentes.
- Facilitar la supervisión.

El problema de distribución en planta es un factor clave para maximizar los beneficios de una industria (García-Hernández, 2011). Se sabe que un buen diseño de distribución de las instalaciones en una industria es muy eficaz para reducir los costos de fabricación, los plazos de entrega, el trabajo en progreso y la productividad. Contribuye a la eficiencia general de las operaciones y puede reducir significativamente los costos operativos totales, hasta un 50% (Tompkins et. al., 2010). Existen muchas taxonomías dentro del diseño de distribución en planta. Una buena clasificación de los distintos problemas de distribución en planta puede encontrarse en el trabajo realizado por Drira et al (2007). Uno de los problemas más estudiados en el diseño de distribución en planta es el problema de distribución en planta de área desiguales (en inglés: Unequal Area Facility Layout Problem, UA-FLP) (Palomo-Romero et al., 2017). Éste fue propuesto por Armour y Buffa (1963) y es definido con un conjunto de instalaciones rectangulares de área desigual que han de ser distribuidas en base

a una función objetivo en un planta también rectangular de dimensiones conocidas, de forma que no haya solapamiento entre las instalaciones y que la suma del área de todas las instalaciones no sea superior al área total de la planta. De acuerdo con (Gonçalves y Resende, 2015), la función objetivo más frecuente a optimizar se basa en la minimización del flujo de materiales, ya que está directamente relacionado con los costes totales en un planta industrial.

2. Objetivos

El objetivo fundamental que se busca con el desarrollo de esta tesis doctoral se basa en mejorar los resultados obtenidos previamente por las propuestas existentes en el problema de distribución en planta de áreas desiguales (UA-FLP). Este objetivo principal puede ser a su vez subdividido en varios subobjetivos.

1. Revisión, estudio y análisis de los trabajos previos existentes en la bibliografía que han abordado el problema de distribución en planta de áreas desiguales. A través de ello, se realizará el estado del arte del problema de distribución en planta en general y, posteriormente, será enfocado el problema de distribución en planta de áreas desiguales, en particular.

2. Revisión, estudio y análisis de las estrategias de optimización más novedosas y efectivas en problemas complejos, teniendo como meta seleccionar propuestas prometedoras para abordar el problema de distribución en planta de una forma exitosa que permita mejorar el rendimiento obtenido mediante otros métodos del estado del arte.

3. Diseño e implementación de una nueva estrategia que se base en el método de optimización previamente seleccionado y aborde la optimización de flujo de material existente en el problema de distribución en planta de áreas desiguales.

4. Diseño e implementación de una nueva estrategia paralela que permita mejorar aún más el rendimiento de las estrategias seleccionada previamente, mejorando la diversidad de los diseños industriales y ampliando el rango de búsqueda de soluciones. Esta nueva propuesta se aplicará al problema de distribución en planta de áreas desiguales considerando el coste de flujo de material como función objetivo a minimizar.

5. Revisión, estudio y selección de los problemas de distribución en planta de instalaciones de áreas desiguales de referencia para realizar la experimentación de las estrategias sugeridas, teniendo en cuenta, que se evalúen todos los rangos y tamaños de problemas.

6. Experimentación empírica de las propuestas sugeridas. Extracción de los resultados obtenidos y realización de comparativas con las mejores soluciones obtenidas en la bibliografía.

7. Extracción y análisis de las conclusiones de la investigación realizada a través de los resultados alcanzados por las técnicas propuestas, así como, el establecimiento de las posibles líneas futuras y prometedoras de investigación.

3. Los problemas de distribución en planta y los problemas de distribución en planta de áreas desiguales

3.1. *Introducción*

Los problemas de distribución en planta comprenden desde la asignación de actividades a ciudades, sitios, campus o edificios, hasta la ubicación de equipos y grupos de personal en un edificio (Liggett, 2000). Puede englobar el diseño y la asignación de espacio en un edificio nuevo o, incluso, la reasignación de espacio en un edificio existente. En un edificio existente, se pueden usar herramientas de diseño para el problema continuo de la gestión del espacio. Por ejemplo, a medida que los grupos de proyectos aumentan o disminuyen de tamaño, ¿cómo deben ubicarse los empleados dentro de una oficina para que la contigüidad del grupo se mantenga con un número mínimo de movimientos en el espacio de trabajo? ¿Cómo se puede consolidar efectivamente el espacio no utilizado para minimizar los costos? (Meller et al., 2010). Desde principios de la década de 1960, se han desarrollado numerosas aplicaciones informáticas para la solución automatizada de los problemas de diseño espacial. El interés en esta área proviene de los investigadores de ingeniería en ciencias de la computación que analizan principalmente los problemas de diseño de plantas o instalaciones de producción, así como de arquitectos y diseñadores de interiores interesados en el diseño de grandes instalaciones, tales como edificios de oficinas, universidades, hospitales o grandes almacenes (Heragu, 2008). Los enfoques para realizar el diseño de distribución en planta difieren en términos del tipo de problema abordado, así como los criterios utilizados para generar, evaluar y comparar soluciones (Drira et al., 2007).

Todos los problemas de planificación del espacio consisten en un conjunto de actividades a ubicar y un espacio en el que ubicarlas. El espacio se puede representar de diferentes maneras, proporcionando así un método para clasificar tipos alternativos de problemas de diseño (Thürmer y Wüthrich, 1997):

- Espacio como objetos discretos.
- Espacio como área.
- Espacio como área y forma.

Tanto la formulación del problema como las técnicas de solución se ven afectadas por la forma en que se representan la actividad y el espacio físico. El problema de diseño más simple es la asignación de un conjunto de actividades discretas a un conjunto de ubicaciones discretas de tal manera que cada actividad se asigne a una única ubicación. Esto se llama un problema de asignación uno a uno (también conocido como un problema de diseño de área igual) y tiene algunas aplicaciones muy interesantes. En general, las aplicaciones de planificación espacial no son tan sencillas como las asignaciones individuales. Las áreas requeridas por las actividades no son necesariamente iguales, por lo que no es factible combinar actividades y ubicaciones de forma individual.

Los problemas más difíciles de representar son aquellos en los que una actividad se representa como un polígono en la planta. Este polígono debería poder adoptar cualquier forma y ubicación mientras se mantiene el área de actividad requerida. El método usado para resolver el problema de distribución en planta de áreas desiguales tiene un impacto significativo en el enfoque de solución adoptado.

En general, el diseño de la distribución de las instalaciones es una disposición de espacio y recursos en la organización (Roslin et al., 2010). La solución obtenida del problema de distribución de las instalaciones es un diseño que especifica la ubicación relativa de cada departamento. Luego, se puede realizar más trabajo para obtener el diseño detallado, que especifica las ubicaciones exactas del departamento, las estructuras del pasillo y el diseño dentro de cada departamento (Meller y Gau, 1996).

Una buena distribución de las instalaciones es crucial para lograr la eficiencia de la producción porque influye directamente en los costos de fabricación, los plazos de entrega, el trabajo en proceso y la productividad (García-Hernández et al., 2013). Según Francis y White (1974), un diseño eficiente puede reducir entre el 10% y el 30% el costo de flujo de materiales. El objetivo del diseño de distribución en planta es lograr la disposición más efectiva de acuerdo con algunos criterios y/u objetivos establecidos, al tiempo que admite algunas limitaciones. En la literatura, la eficiencia del diseño se mide comúnmente en términos de costos de flujo de material, pero otros objetivos o criterios pueden considerarse como: el aprovechamiento del espacio, la seguridad, satisfacción de distancia/lejanía entre instalaciones, etc. Muther (1973) enumeró los siguientes objetivos a considerar en el problema de distribución de una planta:

1. Reducir las distancias de desplazamiento de los materiales.
2. Tener un flujo regular de piezas y productos evitando cuellos de botella en la producción.
3. Utilizar eficazmente el espacio ocupado por las instalaciones.
4. Mejorar la satisfacción y seguridad de los trabajadores.
5. Reducir el tiempo de entrega en la producción.
6. Obtener flexibilidad en el diseño con la finalidad de que se pueda reajustar fácilmente para condiciones cambiantes de producción.

3.2. Los problemas de distribución en planta

Uno de los enfoques más populares para el diseño de distribución en planta de instalaciones fue formulado por primera vez por Koopmans y Beckmann en 1967, para problemas relacionados con la asignación de instalaciones de fabricación a posiciones de manera que el costo de transporte del flujo de material entre instalaciones se minimiza. Este problema fue conocido como el Problema de Asignación Cuadrática (QAP), y, se ocupa de encontrar ubicaciones óptimas para un conjunto de objetos que están interrelacionados. El problema

puede describirse como sigue:

Considerese la asignación de N actividades a N o más sitios, cada uno de los cuales puede acomodar una y solo una actividad. Asociado con cada par de actividades (i,j) hay una medida de interacción $Q(i,j)$ (por ejemplo: intensidad de comunicación, nivel de tráfico, etc.) que se asocia con cada par de posiciones (k,l) . Además, existe una medida de separación espacial $C(k,l)$ (ejemplo: distancia, tiempo de viaje, etc.), y un costo fijo, $F(I,k)$, que puede estar asociado con la ubicación de la actividad i en el sitio k . Si $A(i)$ denota el sitio al que se asigna la actividad i en un mapeo A de actividades a los sitios, el costo total de un mapeo (solución) puede ser dado por:

$$\text{cost}(A) = \sum_{\text{activity } i} F(i, A(i)) + \sum_{\text{activity } i} \sum_{\text{activity } j} [Q(i,j)C(A(i), A(j))]$$

El objetivo es encontrar un mapeo A , de modo que esta función de costo se minimice.

Armour y Buffa formuló por primera vez el problema de distribución en planta de áreas desiguales (en inglés, UA-FLP) en 1963. Consideró el diseño de una planta de fabricación donde el criterio a minimizar era el costo del flujo de material entre departamentos. Su trabajo quedó plasmado en un programa de ordenador llamado CRAFT (técnica de asignación relativa computarizada de instalaciones) (Armour y Buffa, 1963).

En la formulación de asignación cuadrática, el diseño de distribución en planta puede verse como un problema combinatorio en el que las actividades indivisibles (por ejemplo, departamentos o estaciones de trabajo de empleados individuales) deben asignarse a ubicaciones fijas en una planta industrial. En principio, es posible resolver este problema mediante una enumeración exhaustiva de todas las formas posibles de asignar actividades a las ubicaciones, y mediante la selección de un planta que satisfaga las limitaciones dadas y/u (como en el caso de la formulación de asignación cuadrática) obtener el valor

mínimo para la función objetivo. En la práctica, esto resulta inviable para problemas de tamaño realista (problemas de más de 15 instalaciones) ya que el número de combinaciones de actividad/ubicación involucradas puede ser enorme.

Se puede demostrar que los problemas de asignación cuadrática pertenecen a una clase de problemas matemáticos conocidos como NP-completos. En general, se acepta que la solución eficiente de los problemas NP-completos es imposible en principio. Sin embargo, existen varias buenas estrategias de solución aproximada que producen soluciones de alta calidad para problemas de tamaño realista a un costo aceptable.

Komarudin (2009) propuso una comparación entre las principales categorías de FLP:

Problema	Tamaño de la instalación	Localización candidata	Área de la planta
QAP	Iguales con dimensiones fijas o ignoradas	Fijas	Fijas o ignoradas
UA-FLP	Desiguales	Variables	Mayor o igual al área total de las instalaciones
MLP	Desiguales con dimensiones fijas	Variables	Mayor o igual al área total de las instalaciones o dimensiones libres

Tabla 1. Comparación de categorías FLP según Komarudin (2009).

El QAP fue presentado originalmente por Koopmans y Beckman (1957) que intentaban modelar un problema de ubicación de instalaciones (Çela, 1998). El objetivo de QAP es asignar todas las instalaciones a todas las ubicaciones de manera que se minimice el costo total de la asignación. Cabe destacar que las

instalaciones y ubicaciones tienen áreas iguales, que son fijas y conocidas de antemano.

El UA-FLP fue presentado originalmente por Armor y Buffa (1963). El objetivo del problema UA-FLP es dividir la región total de la planta en subregiones, con área apropiada, para minimizar el costo total del flujo de material. En este problema teniendo en cuenta que la suma total de las áreas de las subregiones debe ser menor o igual al área total de la planta. La forma de la región (diseño de la planta) y las subregiones (instalaciones) son regulares y desiguales (Tate y Smith, 1995a).

El MLP se discute en Heragu y Kusiak (1986). En este problema, el diseño involucra el diseño de máquinas y las estaciones de trabajo (Kusiak y Heragu, 1987). El objetivo a alcanzar en MLP es asignar las máquinas en el diseño de la planta minimizando los costos de manejo de materiales (Tompkins et al., 2003). En este problema, las máquinas son de diferentes tamaños y su ubicación depende de procesos específicos que son necesarios para obtener los productos finales.

Existen multitud de trabajos de investigación que abordan los diferentes problemas de distribución en planta. Para enfatizar las características esenciales que pueden caracterizar los diferentes problemas de distribución en planta, se han propuesto algunas clasificaciones. Un ejemplo de una de ellas pueden verse en la Figura 1 propuesta por Drira et al. (2007). Por lo tanto, los problemas de distribución en planta se distinguen entre sí dependiendo de factores específicos como: las características de la planta industrial, el problema abordado y los enfoques utilizados para resolverlo.

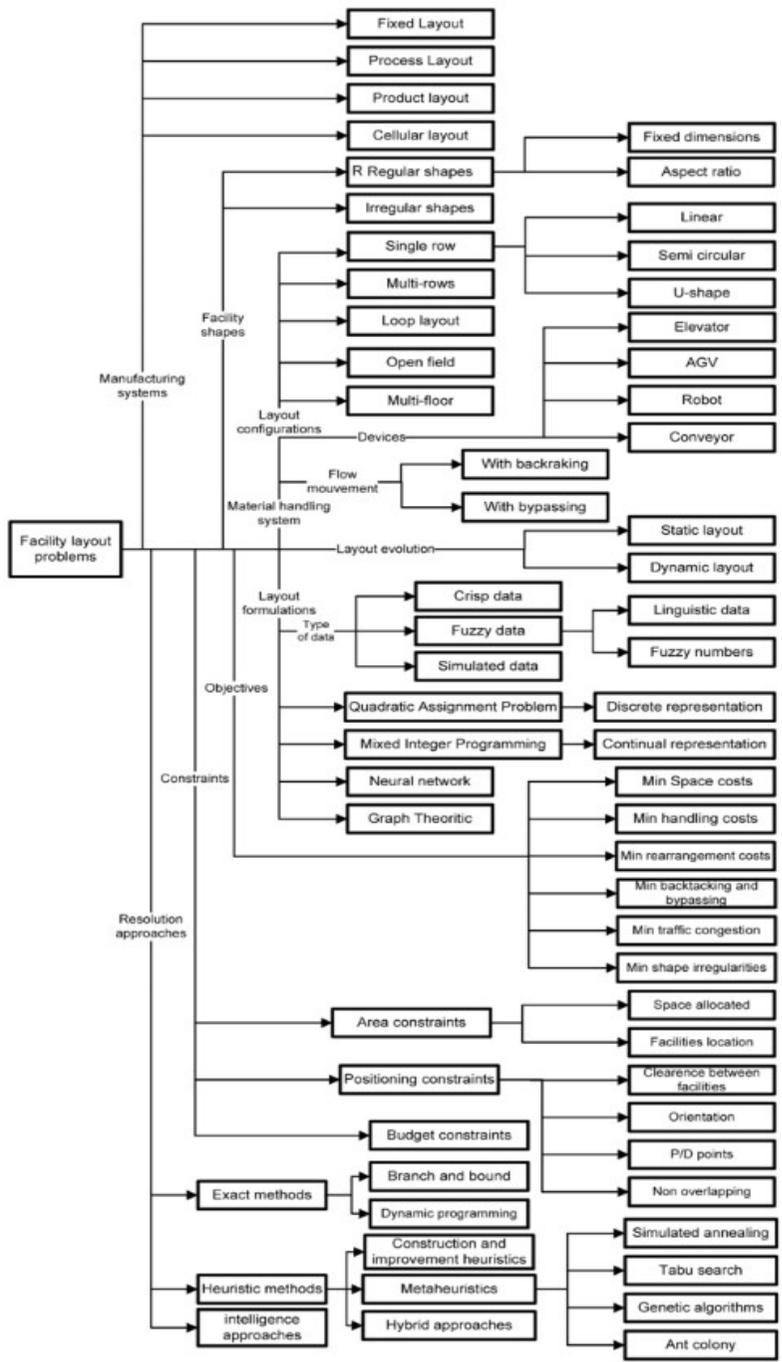


Figura 1. Clasificación de los problemas de distribución en planta tomada de Drira et al. (2007).

3.3. El problema de distribución en planta de áreas desiguales (UA-FLP)

El problema de distribución en planta de áreas desiguales, conocido como UA-FLP en inglés, considera un diseño de planta rectangular que está compuesta por instalaciones rectangulares desiguales que deben colocarse efectivamente en el diseño de la planta industrial. De acuerdo con Aiello et al. (2012) y Gonçalves y Resende (2015), el UA-FLP se define como una región rectangular, A , con dimensiones fijas H y W , y una colección de n departamentos rectangulares. Normalmente, la función objetivo utilizada para minimizar el costo total del flujo de material es el siguiente:

$$\text{Material Flow Cost } F(\mathbf{x}) = \sum_i^n \sum_j^n f_{ij} c_{ij} d_{ij}$$

donde n es el número de instalaciones en la planta de fabricación, f_{ij} es la cantidad de flujo de manejo de material entre las instalaciones i y j , c_{ij} es el costo asociado de mover una unidad de flujo de manejo de material entre instalaciones i y j , y d_{ij} es la distancia medida entre las instalaciones i y j .

Normalmente, el UA-FLP tiene en cuenta características que son cuantitativas como por ejemplo: el costo de manejo de materiales, las relaciones de cercanía o la distancia, los requisitos de adyacencia, la relación de aspecto (Babbar-Sebens y Minsker, 2012). Estas características se pueden abordar con diferentes enfoques de optimización. Sin embargo, a veces, las características cualitativas deben tenerse en cuenta (Tuzkaya y Ertay, 2004), éstas pueden ser: preferencias de ubicación de las instalaciones, distribución de los espacios restantes, preferencias de ubicación relativa o cualquier otra preferencia subjetiva que el DM (*Decision Maker*) pueda considerar importante (Garcia-Hernandez et al., 2013).

Para representar una planta industrial existen dos tipos básicos de estructuras de representación. De ellas, la estructura de representación más usada en la

literatura ha sido la estructura de representación continua, que engloba tanto la estructura de bahías flexibles como la estructura de árboles de corte. Por otro lado, existe también la estructura de representación discreta.

3.3.1 Representación continua en el UA-FLP

Dentro de la categoría de representación continua, hay dos posibles estructuras de representación: la estructura de bahías flexibles y estructura de árbol de cortes.

Estructura de bahías flexibles

La estructura de bahías flexibles (FBS) fue propuesta por (Tong, 1991). Divide la instalación en bahías verticales u horizontales. El ancho de la bahía es flexible, dependiendo de los departamentos que contiene. Además, el número de bahías y el número de departamentos en cada bahía también son flexibles. De esta manera, el problema se vuelve más simple y fácil de resolver, y la complejidad del problema se reduce a la determinación del orden de ubicación de la instalación y el número total de instalaciones que contiene cada bahía (Komarudin, 2009). Además, la representación de la estructura de bahías flexibles tiene la ventaja de que las bahías ofrecen la posibilidad de incorporar pasillos fácilmente (Aiello et al., 2012). Esto favorece al diseñador para traducir el modelo en un diseño de instalación real (Konak et al., 2006). Existen muchos trabajos publicados que usan esta estructura de representación para resolver el problema de distribución en planta de áreas desiguales (UA-FLP), como por ejemplo: Tate y Smith (1995a), Mak et al. (1998), Lee y Lee (2002), Lee et al. (2003, 2005), Enea et al. (2005), Gómez et al. (2005), Norman y Smith (2006), y Chae y Peters (2006). La Figura 2 ofrece un ejemplo de la estructura de representación por bahías flexibles.

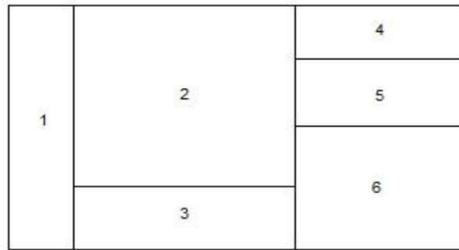


Figura 2. Ejemplo de estructura de representación de bahías flexibles (FBS.) tomada de Aiello et al. (2006).

Estructura de árbol de cortes

Tam (1991, 1992) propuso una nueva forma de representación continua mediante el diseño de la estructura de representación de la planta como una estructura de árbol de cortes que se construye recursivamente al dividir la planta rectangular. Cada partición rectangular en la estructura de cortes corresponde al espacio asignado a una actividad. Tam utiliza el análisis de conglomerados para generar un árbol de cortes que coloca las actividades muy cerca unas de otras. Las hojas del árbol representan las actividades que se insertarán, y los nodos interiores representan la operación de corte (corte izquierdo, corte derecho, corte inferior o corte superior como se muestra en la Figura 3). En su propuesta (Tam, 1991; Tam, 1992), se utiliza un procedimiento de *Simulated Annealing* para intercambiar operadores de cortes en el árbol, generando así diferentes esquemas de partición rectangular. La función objetivo utilizada para ejecutar el proceso de *Simulated Annealing* incluye el coste de flujo de material y un término de penalización para restricciones geométricas. Tam muestra buenos resultados con problemas de distribución en planta de 20 y 30 actividades.

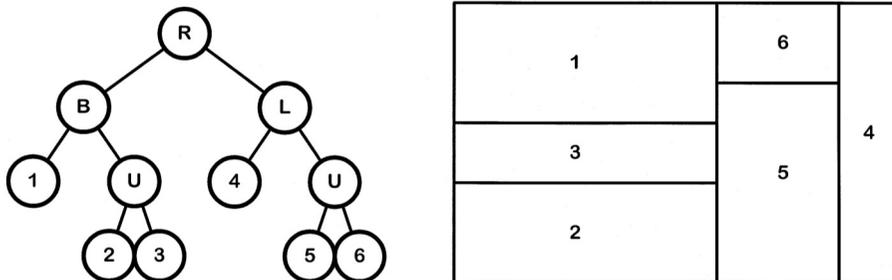


Figura 3. Ejemplo de estructura de árbol de corte tomado de Tam (1991).

Aunque la representación mediante la estructura bahías flexibles es ligeramente más restrictiva que la formulación de árbol de cortes, permite que pueda obtenerse una estructura de pasillos de forma natural en el diseño de planta industrial, una propiedad que puede ser útil (Aiello et al., 2012). Cabe destacar que el número de diseños en planta posibles es mucho mayor cuando se emplea la estructura de representación mediante árbol de cortes que cuando se usa la estructura flexible de bahías como método de representación, sin embargo, usar la estructura de árbol de cortes, aumenta la complejidad a la hora de resolver el problema (Garcia-Hernandez et al., 2020a, 2020b).

3.3.2 Representación discreta del UA-FLP

En la categoría de representación discreta de la planta industrial, se divide toda la planta en cuadrados iguales (mismas área y dimensiones) a modo de rejilla. En este punto pueden ocurrir dos casos. El primero hace referencia al problema QAP donde cada una de las celdas se asocia a una instalación debido a que las instalaciones son iguales también en cuanto a área y dimensiones. En el segundo caso, si las instalaciones no son iguales en área y/o forma, se requiere una estructura adicional para asociar cada celda a su correspondiente instalación. En este sentido, Balakrishnan et al. (2003) propuso una estructura de interconexión llamada *Space Filling Curve (SFC)*. Mediante ella, es posible identificar a cada cuadrado de la rejilla con su correspondiente instalación.

Muchos trabajos de investigación han usado esta estructura de representación cuando han abordado el problema de distribución en planta de áreas desiguales, entre ellos podemos listar los siguientes: Armour y Buffa (1963), Chan y Tansri (1994), Conway y Ventakaramanan (1994), Tate y Smith (1995b), Mak et al. (1998), Singh et al. (1998), Tavakkoli-Moghaddain y Shayan (1998), Balakrishnan et al. (2003a, 2003b), El-Baz (2004), Hu y Wang (2004), Wang et al. (2005), y Rankumar et al. (2008), etc.

3.4. Propuestas de resolución del problema de distribución en planta de áreas desiguales

Existen multitud de propuestas y enfoques para abordar el problema de distribución en planta con instalaciones de áreas desiguales. Entre ellos podemos identificar dos categorías principales: los procedimientos exactos y los enfoques aproximados. En la primera categoría se engloba la teoría gráfica, la programación entera, la programación dinámica y ramificación y poda. Dentro de la última categoría, es decir, los enfoques aproximados, podemos subdividir a su vez entre los siguientes enfoques: métodos heurísticos, métodos meta-heurísticos y otro tipo de enfoques.

3.4.1 Procedimientos exactos

Teoría gráfica

Los enfoques de teoría gráfica también abordan el problema de distribución en planta de áreas desiguales. En estos enfoques, el diseño de la planta se construye mediante una estructura de un grafo plano donde los nodos representan espacios (instalaciones) y los enlaces representan las adyacencias requeridas (Hassan y Hogg, 1987). Este tipo de enfoques tienen como desventaja que aunque siempre es posible construir un diseño de distribución en planta de las instalaciones a partir de un grafo plano que cumpla con los requisitos de adyacencia dados entre las instalaciones y el área exterior, el diseño resultante puede no cumplir con los requisitos de tamaño y forma impuestos en cada instalación. Por lo que, el diseño de la distribución en planta que cumpla con los requisitos de tamaño y forma es un problema no trivial (Meller y Gau, 1996). En este enfoque, se supone que se conoce las preferencias de adyacencia entre las instalaciones (Foulds y Robinson, 1978). Dentro de este tipo de propuestas, Goetschalckx (1992) desarrolló SPIRAL que era un método eficiente para generar un diseño de distribución en planta de instalaciones rectangulares que cumplía con los requisitos de área del grafo plano. La Figura 4, muestra un ejemplo de aplicación del método SPIRAL

propuesto por Goetschalckx (1992) sobre el diseño de distribución en planta de un problema de áreas desiguales rectangulares. Por su parte, Montreuil et al. (1993), utilizó un modelo de programación lineal para generar un diseño de la distribución en planta de espacios rectangulares a partir de un grafo de adyacencia plano. Ambos métodos tienen como limitación que el perímetro del edificio debe ser rectangular.

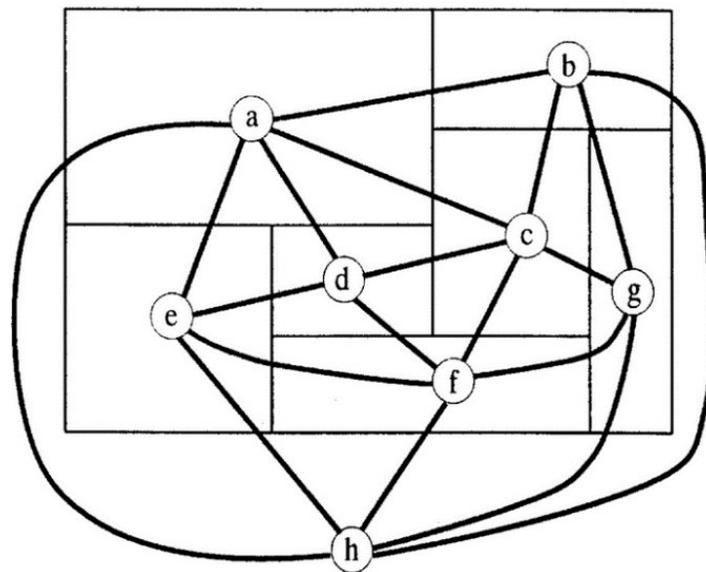


Figura 4. Ejemplo de teoría gráfica tomado de Goetschalckx (1992).

La formulación teórica del gráfico difiere del enfoque tradicional de asignación cuadrática (QAP) del problema de distribución en planta en varias cosas. La primera diferencia fundamental es que el enfoque de la teoría de grafos considera solo los requisitos de adyacencia directa. La segunda diferencia es que no se tienen en cuenta los pares de instalaciones no adyacentes con respecto a los costos de comunicación, incluso si están relativamente cerca en el diseño. Además, la tercera diferencia es que los costos fijos no se incluyen en la formulación teórica del grafo ni se ajustan instalaciones asignadas previamente. En este sentido, Foulds (1991) considera que el enfoque teórico gráfico es más apropiado para el diseño de una nueva instalación donde hay más libertad en el diseño de la distribución en planta de las instalaciones, mientras que la

formulación de QAP es más útil en una situación previamente estructurada y diseñada.

Programación entera

Un problema de programación de enteros es un programa de optimización matemática en el que algunas o todas las variables están restringidas a ser enteros. En muchos entornos, el término se refiere a la programación lineal entera (ILP).

Montreuil (1990) introdujo una programación entera mixta en UA-FLP, cuyo objetivo se basaba en minimizar los costos de flujo de materiales considerando la distancia rectilínea entre los centroides de las instalaciones que conformaban la planta industrial. Aunque su enfoque tuvo mucho éxito, sólo conseguía resolver el problema de distribución en planta de áreas desiguales con seis o menos instalaciones de forma óptima (Meller y Gau, 1996). Más adelante, Sherali et al. (2003) presentaron una mejora de este modelo usando la idea del artículo de Meller et al. (1999). Con este nuevo trabajo fueron capaces de obtener soluciones óptimas para el problema de distribución en planta de áreas desiguales con hasta nueve instalaciones. Konak et al. (2006) utilizaron un enfoque de programación de enteros mixtos para el UA-FLP considerando la estructura de bahías flexibles para la representación de la planta industrial. Su método fue extensivamente testeado y obtuvieron soluciones óptimas a problemas con hasta quince instalaciones.

Programación dinámica

En matemáticas, ciencias de la computación y economía, la programación dinámica es un método para resolver problemas complejos dividiéndolos en subproblemas más simples. La idea clave detrás de la programación dinámica es bastante simple. En general, para resolver un problema dado, necesitamos resolver diferentes partes del problema (subproblemas), luego combinar las soluciones de los subproblemas para llegar a una solución general. A menudo,

muchos de estos subproblemas son realmente los mismos. El enfoque de programación dinámica busca resolver cada subproblema solo una vez, reduciendo así el número de cálculos. Una vez que la solución a un subproblema dado ha sido calculada, se almacena o "memoriza", la próxima vez que se necesita la misma solución, simplemente se mira en memoria. Este enfoque es especialmente útil cuando el número de subproblemas repetitivos crece exponencialmente en función del tamaño de la entrada. Rosenblatt (1986) fue el primer autor en usar la programación dinámica en el problema de distribución en planta. Planteó el problema con instalaciones de igual tamaño y con necesidades dinámicas en la producción (es decir, que cambian en cortos plazos de tiempo). El objetivo era minimizar la suma de costos de manejo de materiales y los costos de reorganización en todos los períodos. Con su propuesta, pudo resolver el problema para diseños con menos de seis instalaciones y cinco períodos de tiempo (Kulturel-Konak, 2007).

Ramificación y poda

Ramificación y poda (en inglés, *Branch & bound*) es un algoritmo general para encontrar soluciones óptimas de diversos problemas de optimización, especialmente la optimización indiscreta y combinatoria. Fue propuesto por primera vez por Land y Doig en 1960. Este algoritmo consiste en una enumeración sistemática de todas las soluciones candidatas, donde grandes subconjuntos de candidatos infructuosos se descartan de forma masiva, utilizando límites estimados superiores e inferiores de la cantidad que se optimiza.

Para facilitar una descripción concreta, suponemos que el objetivo es encontrar el valor mínimo de una función $f(x)$, donde x se extiende sobre un conjunto S de soluciones admisibles o candidatas (el espacio de búsqueda o región factible). Hay que tener en cuenta que uno puede encontrar el valor máximo de $f(x)$ al encontrar el mínimo de $g(x) - f(x)$.

El proceso de ramificación y poda requiere de dos herramientas. Lo primero es

un procedimiento de división que, dado un conjunto de candidatos S , devuelve dos o más conjuntos más pequeños $S_1, S_2, S_3 \dots$ cuya unión cubre S . Hay que tener en cuenta que el mínimo de $f(x)$ de S es $\min \{v_1, v_2 \dots\}$, donde cada v_i es el mínimo de $f(x)$ dentro de S_i . Este paso se llama ramificación, ya que su aplicación recursiva define una estructura de árbol (el árbol de búsqueda) cuyos nodos son los subconjuntos de S .

La segunda herramienta es un procedimiento que calcula los límites superior e inferior para el valor mínimo de $f(x)$ dentro de un subconjunto dado de S . Este paso se llama *bounding (poda)*.

La idea clave del algoritmo ramificación y poda es: si el límite inferior para algún nodo de árbol (conjunto de candidatos) A es mayor que el límite superior para algún otro nodo B , entonces A puede descartarse de forma segura de la búsqueda. Este paso se llama poda, y generalmente se implementa manteniendo una variable global m (compartida entre todos los nodos del árbol) que registra el límite superior mínimo visto entre todas las subregiones examinadas hasta ahora. Cualquier nodo, cuyo límite inferior es mayor que m , puede descartarse. El proceso se detiene cuando el conjunto candidato S actual se reduce a un solo elemento, o cuando el límite superior del conjunto S coincide con el límite inferior. De cualquier manera, cualquier elemento de S será un mínimo de la función dentro de S .

Kouvelis y Kim (1992) utilizaron un algoritmo de ramificación y poda para el problema de distribución en planta en bucle. Meller et al. (1999) combinaron la estructura de grafos con el algoritmo de ramificación y poda para resolver el problema de distribución en planta de áreas desiguales. Mediante esa estrategia lograron resolver el UA-FLP de manera óptima con hasta ocho instalaciones. Kim y Kim (1998) propusieron un enfoque de ramificación y poda para encontrar la ubicación óptima para establecer los puntos de la recogida y devolución en cada instalación.

3.4.2 Algoritmos aproximados

Como los métodos óptimos están limitados por el número de instalaciones, se han desarrollado métodos subóptimos para abordar problemas más complejos. Así, multitud de autores han utilizado entre otros métodos los algoritmos heurísticos y metaheurísticos para resolver el problema de distribución en planta.

Algoritmos heurísticos

Los algoritmos heurísticos se pueden clasificar como algoritmos de tipo de construcción (donde se construye una solución desde cero) y algoritmos de tipo de mejora (donde se mejora una solución inicial) (Singh y Sharma, 2006).

Los algoritmos de tipo de construcción se consideran el enfoque heurístico más simple, pero la solución obtenida se considera en términos de calidad como no satisfactoria. Existen algunos enfoques aplicados al problema de distribución en planta que utilizan este tipo de algoritmos. Estos se detallan a continuación:

ALDEP fue desarrollado por Seehof y Evans (1967). Este enfoque funciona de la siguiente manera: primero, se selecciona una instalación al azar y se asigna a la esquina superior izquierda del diseño. La siguiente instalación seleccionada para ser asignada es la que tiene una relación que es mayor o igual a una relación especificada por el usuario, con una primera instalación seleccionada al azar. Este procedimiento se repite hasta que todas las instalaciones se asignen en el diseño.

CORELAP fue creado por Lee y Moore (1967). Utilizaron el índice de cercanía total de cada instalación para determinar un diseño. De esta forma, se selecciona la instalación con el índice de cercanía total más alto y se asigna al centro del diseño. Luego, las instalaciones posteriores se agregan al diseño según sus relaciones con las instalaciones ya asignadas.

MAT fue propuesto por Edwards et al. (1970). Este enfoque clasifica el par de instalaciones según sus valores de flujo y los pares de ubicación según sus valores de distancia. Luego, esta información se utiliza para determinar un diseño. Permite al usuario asignar instalaciones a cualquier ubicación deseada.

PLANET fue creado por Deisenroth y Apple (1972). Este algoritmo asigna las instalaciones en tres etapas. En primer lugar, se determina el costo de los flujos unitarios entre cada par de instalaciones. En segundo lugar, se selecciona el orden de instalación. Finalmente, las instalaciones se colocan en el diseño en el orden en que se seleccionaron en la segunda etapa.

LSP fue desarrollado por Zoller y Adendorff (1972). Este algoritmo consiste en un simulador que genera la secuencia en la que se asignarán las instalaciones en un diseño, y un modo de construcción que determina un diseño bidimensional para la secuencia generada previamente.

COFAD fue formulado por Tompkins y Reed (1976). Este algoritmo es una modificación de CRAFT (explicado a continuación) que incluye los costos de traslado para todos los sistemas alternativos de manejo de materiales, por lo tanto, este algoritmo integra el problema de selección del sistema de manejo de materiales con el problema de diseño. Más tarde, Shore y Tompkins (1980) modificaron COFAD para incorporar flexibilidad en el proceso de diseño; esta nueva versión se llamaba COFAD-F.

SHAPE es un algoritmo de tipo de construcción que fue propuesto por Hassan et al. (1986). Utiliza una representación discreta y un objetivo basado en la distancia rectilínea entre los centroides de la instalación. La selección de la instalación depende de una clasificación, que se basa en el flujo de cada instalación y un valor de flujo crítico definido por el usuario. Sin embargo, debido a que la forma de la instalación está controlada por la función objetivo, la forma de las instalaciones puede deteriorarse hacia el final.

NLT fue desarrollado por Camp et al. (1991). Este algoritmo se basa en la programación no lineal y utiliza la distancia Euclidiana como métrica de distancia. En NLT hay tres conjuntos de restricciones. Los autores transformaron este modelo restringido en uno sin restricciones mediante el uso del método de función de penalización cuadrática de punto exterior. Con un enfoque de tres etapas, sucesivamente, los problemas más difíciles se resuelven utilizando la

solución de la etapa anterior como un punto de solución inicial.

En cuanto a los algoritmos de mejora, en ellos se realizan iteraciones para mejorar la solución inicial. Estos métodos se pueden combinar con métodos de construcción fácilmente. A este respecto, CRAFT (Armor y Buffa, 1963) parece ser el primer enfoque de tipo de mejora. Comienza por determinar el centroide de cada instalación. Luego, realiza intercambios bidireccionales o tridireccionales de los centroides de instalaciones no fijas que también son iguales en área o adyacentes en el diseño actual. Para cada intercambio, CRAFT calcula y reduce el costo estimado y elige el intercambio con la mayor reducción estimada. Luego intercambia las instalaciones exactamente y continúa hasta que no exista una reducción estimada debido a los intercambios de dos o tres vías. Hicks y Cowen (1976) criticaron el procedimiento de intercambio porque puede conducir a instalaciones con forma irregular.

H63 fue desarrollado por Hillier (1963). Este algoritmo heurístico se basa en una tabla de deseabilidad de movimiento que consta de valores que representan los cambios de costo que resultarían al mover una instalación de su ubicación actual a una adyacente. H63 consideró solo intercambios por parejas entre instalaciones adyacentes, que tienen áreas iguales.

FRAT fue implementado por Khalil (1973). Este algoritmo utiliza principios de otros algoritmos conocidos como CRAFT y H63. Primero, determina la diferencia entre la distancia más larga y la más corta, y luego, el algoritmo lleva a cabo dos procedimientos, es decir, el procedimiento de determinación del costo total y el procedimiento de intercambio. Este algoritmo solo se puede aplicar a problemas cuyas instalaciones tienen áreas iguales.

DISCON fue desarrollado por Drezner (1980). Este enfoque modeló FLP como un problema de programación matemática no convexo. Para resolver este problema, se utiliza un algoritmo de dos fases llamado dispersión-concentración. En la primera fase, se encuentran buenas condiciones iniciales utilizando el método de gradiente diferencial de Lagrange. La segunda fase consiste en

concentrar las instalaciones para que estén lo más cerca posible. En este enfoque, aunque la fase de dispersión proporciona buenos puntos de partida, es difícil justificar este resultado.

MULTIPLE es un enfoque de tipo de mejora de múltiples pisos que fue propuesto por Bozer et al. (1994). Para representar un diseño, utilizaron una representación discreta. MULTIPLE extiende el método CRAFT aplicando curvas de relleno de espacio. Este enfoque mejoró a CRAFT al aumentar el número de intercambios considerados en cada iteración y, además, puede restringir la irregularidad de las formas de las instalaciones mediante el uso de una medida de irregularidad. Sin embargo, debido a que utiliza la representación discreta, las formas de las instalaciones pueden no ser rectangulares.

Algoritmos metaheurísticos

Por otro lado, los enfoques metaheurísticos han ganado importancia durante los últimos años, ya que en estos enfoques se obtienen en general mejores resultados que los métodos deterministas, especialmente en problemas de distribución en planta de tamaño mediano y grande. Dentro de las propuestas metaheurísticas más comunes podemos hablar de las siguientes:

- Algoritmos genéticos (GAs).

Estos se basan en una búsqueda heurística que imita el proceso de la evolución natural. Esta metaheurística se usa habitualmente para generar soluciones factibles para la optimización y los problemas de búsqueda. Los algoritmos genéticos pertenecen a la clase más amplia de algoritmos evolutivos (EA), que generan soluciones a los problemas de optimización utilizando técnicas inspiradas en la evolución natural, como la herencia, la mutación, la selección y el cruce.

De acuerdo con Michalewicz et al. (1996), los algoritmos genéticos se basan en el principio de evolución (supervivencia del más apto). En tales algoritmos, una población de individuos (soluciones potenciales)

experimenta una secuencia de transformaciones unarias (tipo de mutación) y de orden superior (tipo cruzado). A medida que estos individuos se esfuerzan por sobrevivir, un esquema de selección, sesgado hacia individuos más aptos, selecciona a los individuos que pasarán a la próxima generación. Después de varias generaciones, el algoritmo converge con el mejor individuo que, con suerte, representa el óptimo.

Los algoritmos genéticos comparten las siguientes características (Tate y Smith, 1995): una población inicial de soluciones (puede generarse aleatoriamente); un mecanismo para generar nuevas soluciones combinando características de soluciones en la población existente (reproducción); un mecanismo para generar una nueva solución al operar en una única solución previamente conocida (mutación); un mecanismo para seleccionar el conjunto de soluciones de la(s) población(es), dando preferencia a aquellos con mejores valores de función objetivo (selección); un mecanismo para eliminar soluciones de la población. Las soluciones se pueden seleccionar para mutar o reproducir. La selección se realiza con un sesgo hacia la elección de las mejores soluciones en la población actual. Para el problema de diseño de las instalaciones, la mutación puede tomar la forma de alguna variante del intercambio por pares. La característica clave del proceso de reproducción utilizado por Tate y Smith (1995) es que cualquier actividad asignada a la misma ubicación en ambos padres ocupará esa ubicación en la descendencia. Para las ubicaciones restantes, las asignaciones de actividades se eligen aleatoriamente de uno u otro padre. Cualquier actividad no asignada se compara con las ubicaciones restantes sin asignar. A medida que se crean niños, se eliminan (descartan) las soluciones con los valores más pobres del objetivo para mantener a la población del mismo tamaño.

Los algoritmos genéticos han sido muy utilizados para abordar el problema de distribución en planta (Pierreval et al., 2003), como por

ejemplo: Tam (1992b), Banerjee y Zhou (1995), Mak et al. (1998) Matsuzaki et al. (1999), Tam y Chan (1998), Azadivar y Wang (2000), Lee y Lee (2002), Lee et al. (2003, 2005), Dunker et al. (2003), Hu y Wang (2004), El-Baz (2004) y Ficko et al. (2004) para el problema de distribución en planta estático; y Conway y Ventakaramanan (1994), Balakrishnan y Cheng (2000), Balakrishnan et al. (2003) y Dunker et al. (2005) para los problemas de diseño de distribución en planta dinámicos.

También, los GAs han sido muy aplicados al problema de distribución en planta de áreas desiguales. En este sentido, por ejemplo, Tate y Smith (1995a) propusieron una AG para resolver el UA-FLP utilizando la representación de estructura de bahías flexibles y una penalización dinámica o adaptativa. La función se utiliza para guiar la búsqueda en regiones de solución factibles. Wu y Appelton (2002) presentaron un método AG para resolver el UA-FLP considerando las estructuras de pasillo mediante una estructura de representación de árbol de cortes. Gómez et al. (2003) también propusieron un AG para resolver el UA-FLP. Ellos se centraron en un caso particular que implica la consideración explícita de pasillos con la posibilidad de que sean de ancho variable. Wang et al. (2005) utilizaron la representación discreta y un sistema experto basado en reglas para crear curvas de relleno de espacio. Enea et al. (2005) implementaron una AG para buscar una solución óptima en un contexto difuso usando una estructura de bahías flexibles como estructura de representación. Aiello et al. (2006, 2012) sugirieron un enfoque multiobjetivo para resolver el UA-FLP que utiliza un algoritmo de búsqueda genética combinado con el método Electre. Norman y Smith (2006) resolvieron el UA-FLP utilizando un AG con una estructura de representación de bahías flexibles. Consideraron la incertidumbre en costos de flujo de materiales en una escala continua mediante el uso de valores esperados y desviaciones estándar de las previsiones de productos. Por su parte, Liu y Meller (2007) propusieron un enfoque para

resolver el UA-FLP representado como pares de secuencias, utilizando un AG combinado con programación entera mixta. Su método pudo lograr soluciones óptimas para problemas con hasta once instalaciones, mejorando las soluciones a problemas con conjuntos de datos más grandes. Todas estas propuestas fueron realizadas para resolver el problema de distribución en planta de áreas desiguales considerando aspectos cuantitativos. Sin embargo, de acuerdo a Tuzkaya y Ertay (2004), los aspectos cualitativos deben tenerse en cuenta para obtener diseños robustos. Algunos de los aspectos cualitativos considerados podrían ser: posición u orientación específica del departamento, asignación y/o distribución de espacio vacío restante, o cualquier otra característica cualitativa que puede considerarse relevante para el tomador de decisiones (en inglés, *Decision Maker, DM*). Cabe destacar que los aspectos cualitativos son difíciles de considerar por medio de un enfoque clásico de optimización heurística o metaheurística (Brintup et al., 2007). De acuerdo con García-Hernández et al. (2013), este hecho se debe a que los aspectos no cuantitativos pueden ser subjetivos, desconocidos de antemano o cambiantes durante el proceso de diseño. Esto hace que la inclusión del diseñador experto en el enfoque de optimización sea primordial para tratar de considerar aspectos cualitativos del diseño. En este sentido, García-Hernández et al. (2013, 2015a), propusieron una AG interactivo que consideraba las preferencias del diseñador experto en el problema de distribución en planta de áreas desiguales. Más adelante, García-Hernández et al. (2015b) presentaron una propuesta mejorada que permitía reducir la fatiga del diseñador en el AG interactivo mediante la inclusión de métodos de niching en el proceso. Con el objetivo de considerar a la vez aspectos cuantitativos y cualitativos, García-Hernández et al. (2015a), sugirieron un AG interactivo y multiobjetivo. Posteriormente, Palomo-Romero et al. (2017), crearon un GA basado en un modelo de islas que paralelizaba el

algoritmo alcanzando excelentes resultados en el problema de distribución en planta de áreas desiguales.

- Búsqueda tabú (en inglés, *Tabu search*)

La búsqueda tabú es un método de búsqueda local utilizado para la optimización matemática. Las búsquedas locales toman una posible solución a un problema y verifican a sus vecinos inmediatos (es decir, soluciones que son similares, excepto por uno o dos detalles menores) con la esperanza de encontrar una solución mejorada. Los métodos de búsqueda locales tienden a atascarse en regiones subóptimas o en mesetas donde muchas soluciones son igualmente adecuadas. La búsqueda tabú mejora el rendimiento de estas técnicas mediante el uso de estructuras de memoria que describen las soluciones visitadas o los conjuntos de reglas proporcionadas por el usuario. Si una posible solución ha sido visitada previamente dentro de un cierto período a corto plazo o si ha violado una regla, se marca como "tabú", para que el algoritmo no considere esa posibilidad repetidamente. Esta metaheurística ha sido aplicada a los FLP por muchos autores. Así, Hasan y Osman (1995) usaron *Tabu Search* con una función hash que permitía encontrar diseños casi óptimos. Chiang y Kouvelis (1996) emplearon un algoritmo de búsqueda tabú para resolver un FLP basado en el intercambio de dos ubicaciones de instalaciones, además, añadió una estructura de memoria a largo plazo, un tamaño de lista tabú dinámico, criterios de intensificación y estrategias de diversificación. Kaku y Mazzola (1997) y McKendall y Jaramillo (2006) aplicaron *Tabu Search* para resolver el problema del diseño dinámico de las instalaciones, Chiang y Chiang (1998) aplicaron esta metaheurística para el QAP. Más recientemente, Scholz et al. (2009) propusieron un algoritmo *Tabu Search* para resolver el problema de distribución en planta de áreas desiguales. Ellos utilizaron una estructura de representación de árbol de cortes y le incorporaron una curva de relleno

de espacios. Mediante su búsqueda tabú incorporaron cuatro tipos de movimientos de vecindario para encontrar buenas soluciones. Su propuesta obtuvo grandes mejoras con respecto a trabajos previos.

- Recocido simulado (en inglés, *Simulated Annealing*)

El algoritmo de recocido simulado se basa en el proceso de recocido de metales sólidos o cerámicas donde la temperatura del material varía con el objetivo de cambiar sus propiedades físicas. Por analogía, este proceso se aplica para resolver problemas de optimización, donde, en cada iteración, algunos vecinos son evaluados con una cierta probabilidad para decidir si se cambia a un nuevo estado o, por el contrario, se permanece en el actual. El recocido simulado ha sido empleado por varios autores para resolver el problema de distribución en planta de áreas desiguales. Así, Burkard y Rendl (1984) y Chiang y Chiang (1998) emplearon esta técnica para resolver el QAP. Tam (1992a) también resolvió el problema de distribución en planta de áreas desiguales usando recocido simulado. Su propuesta llamada LOGIC realizaba el proceso de optimización mediante el corte inducido por guillotina. Meller y Bozer (1996, 1997) desarrollaron SABLE para resolver el problema de distribución en planta de instalaciones de varios pisos. Chwif (1998) usó el recocido simulado para resolver el problema de distribución en planta que tiene en cuenta el tamaño de la proporción de las instalaciones. Esta metaheurística, también fue empleada en el problema de distribución en planta con requerimientos dinámicos de producción. Este es el caso de Baykasoğlu y Gindy (2001), McKendall et al. (2006), y Sahin et al. (2010).

3.4.3 Colonia de hormigas

La técnica de colonias de hormigas fue originada por Marco Dorigo (1992). Esta búsqueda metaheurística imita el comportamiento de las hormigas para encontrar los caminos de la colonia a la comida. En el medio natural, las hormigas

(inicialmente) deambulan al azar, y al encontrar comida regresan a su colonia mientras establecen rastros de feromonas. Si otras hormigas encuentran ese camino, es probable que no sigan viajando al azar, sino que sigan el camino, regresando y reforzándolo si encuentran comida. Con el tiempo, sin embargo, el rastro de feromonas comienza a evaporarse, reduciendo así su fuerza atractiva. Cuanto más tiempo le toma a una hormiga viajar por el camino y regresar, más tiempo tienden las feromonas a evaporarse. Un camino corto, en comparación, se transcorre con mayor frecuencia y, por lo tanto, la densidad de feromonas se vuelve más alta en caminos más cortos que en los más largos. La evaporación de feromonas también tiene la ventaja de evitar la convergencia a una solución localmente óptima. Si no hubiera evaporación en absoluto, los caminos elegidos por las primeras hormigas tenderían a ser excesivamente atractivos para los siguientes. En ese caso, la exploración del espacio de la solución estaría limitada. Por lo tanto, cuando una hormiga encuentra un buen camino (es decir, corto) desde la colonia hasta una fuente de alimento, es más probable que otras hormigas sigan ese camino, y la retroalimentación positiva, eventualmente lleva a todas las hormigas a seguir un solo camino. La idea del algoritmo de colonias de hormigas es imitar este comportamiento con "hormigas simuladas" caminando alrededor del grafo que representa el problema a resolver. La técnica de optimización basada en hormigas se ha aplicado en varias ocasiones al problema de distribución en planta. En este sentido, Bland (1999) fue el primer autor que la aplicó. Por su parte, Talbi et al. (2001), Ramkumar y Ponnambalam (2006), Ramkumar et al. (2009) y Wong y See (2010) han resuelto el QAP mediante la metaheurística de colonia de hormigas. Corry y Kozan (2004) aplicaron la optimización de colonias de hormigas para el problema de distribución de máquinas. Solimanpur y et al. (2005) emplearon un enfoque de hormigas para el problema de distribución en planta en sistemas de fabricación flexibles. Por otro lado, McKendall y Shang (2006), Baykasoğlu et al. (2006) y Rezazadeh et al. (2009) resolvieron el problema de distribución en planta con requerimientos dinámicos de producción mediante esta metaheurística. Singh

(2010) sugirió un sistema de hormigas integrado con búsqueda local para resolver el problema de distribución en planta. Komarudin y Wong (2010) emplearon la estrategia de colonia de hormigas para resolver el problema de distribución en planta de áreas desiguales. Ellos usaron una representación de árbol de cortes para representar la planta industrial. Además, incorporaron un algoritmo que empleaba diferentes tipos de búsqueda local para mejorar la búsqueda. Ese mismo año, Wong y Komarudin (2010b) también aplicaron ésta técnica para abordar el mismo problema pero usando la estructura de bahías flexibles como representación de la planta. En ambas propuestas, (Wong y Komarudin, 2010a) realizaron una comparativa de las dos técnicas de optimización basadas en hormigas para tratar con espacios vacíos en el UA-FLP. Una exitosa propuesta fue diseñada por Kulturel-Konak y Konak (2011), que usaba esta metaheurística de optimización combinada con una nueva representación de bahía flexible, fueron capaces de alcanzar los mejores resultados en problemas de referencia de tamaño grande y no han sido mejorados hasta el momento. Recientemente, Liu y Liu (2019), han alcanzado unos resultados excelentes usando una técnica de optimización basada en colonia de hormigas, ofreciendo mejoras en la mayoría de los problemas de referencia en el UA-FLP cuando se consideran varios objetivos a la vez.

3.4.4 Otras propuestas

- Sistemas Difusos

La lógica difusa hace referencia a la toma de decisiones con criterios múltiples o el análisis de decisiones con criterios múltiples. Es una subdisciplina de la investigación operativa que considera explícitamente criterios múltiples en entornos de toma de decisiones. Ya sea en nuestra vida diaria o en entornos profesionales, generalmente hay múltiples criterios conflictivos que deben evaluarse para tomar decisiones. El costo o precio suele ser uno de los criterios principales, así como, la

satisfacción del cliente. En relación al problema de distribución en planta, Grobelny (1988), Raoot y Rakshit (1991, 1994), Badiru y Arif (1996), Dweiri (1999), Enea et al. (2005), Safarzadeh y Koosha (2017), usaron técnicas difusas para abordar el problema.

- Redes Neuronales

El término “red neuronal” se usaba tradicionalmente para referirse a una red o circuito de neuronas biológicas. El uso moderno del término a menudo se refiere a redes neuronales artificiales, que están compuestas de neuronas o nodos artificiales. Así, el término tiene dos usos distintos: las redes neuronales biológicas están formadas por neuronas biológicas reales que están conectadas o relacionadas funcionalmente en un sistema nervioso. Por otro lado, las redes neuronales artificiales se componen de neuronas artificiales interconectadas (construcciones de programación que imitan las propiedades de las neuronas biológicas). Las redes neuronales artificiales pueden usarse para resolver problemas de inteligencia artificial sin crear necesariamente un modelo de un sistema biológico real. El sistema nervioso biológico real es altamente complejo. Los algoritmos de redes neuronales artificiales intentan abstraer esta complejidad y enfocarse en lo que hipotéticamente más importa desde el punto de vista del procesamiento de la información. El buen rendimiento (por ejemplo, medido por una buena capacidad de predicción, un error de generalización bajo) o un rendimiento que imita patrones de error de animales o humanos, puede usarse como una fuente de evidencia para apoyar la hipótesis de que la abstracción realmente capturó algo importante desde el punto de vista del procesamiento de la información en el cerebro. Otro incentivo para estas abstracciones es reducir la cantidad de cómputo requerida para simular redes neuronales artificiales, a fin de permitir a uno experimentar con redes más grandes y entrenarlas en conjuntos de datos más grandes. La mayoría de las redes neuronales

artificiales empleadas actualmente para la inteligencia artificial se basan en estimaciones estadísticas, optimización de clasificación y teoría de control. Referido al problema de distribución en planta, Tsuchiya et al. (1996) aplicaron una red neural para abordar el problema del QAP. Por su parte, Garcia-Hernandez et al. (2014), emplearon una red neuronal para resolver el problema de distribución en planta de áreas desiguales. En su propuesta, lograron un modelo que predecía cómo de buena o mala era una solución de cara al diseñador experto a través del entrenamiento de una red neuronal. Ésta propuesta, fue mejorada recientemente por Salas-Morera et al. (2020), mediante la incorporación de un dispositivo de seguimiento visual (en inglés, *Eye tracker*) que permitía captar las preferencias del diseñador mediante el análisis y procesado de su mirada.

- Sistemas Expertos

Se conoce como Sistema Experto dentro de la Inteligencia Artificial a un sistema que intentar simular el comportamiento de un experto humano en la resolución de cualquier tipo de problema. El sistema experto tiene cuatro partes: una base de datos (con los datos sin procesar), una base de conocimiento (con las reglas generales para generar el diseño), una base de prioridad (con las prioridades para reglas) y un motor de inferencia (controla la estructura de resolución de problemas). Aplicados al problema de distribución en planta existen varios trabajos que usan sistemas expertos en sus propuestas, como son: Kumar et al. (1988), Malakooti y Tsurushima (1989), Heragu y Kusiak (1990), Abdou y Dutta (1990), Sirinaovakul y Thajchayapong (1994), y Singh y Sharma (2006).

- Enfoques alternativos:

Existen otras propuestas diferentes para abordar el problema de distribución en planta de áreas desiguales. Así, Ulutas y Kulturel-Konak (2012), utilizaron un sistema inmune artificial para resolver el UA-FLP. Por su parte, Gonçalves y Resende (2015) emplearon un AG de clave

aleatoria sesgada. Sikaroudi y Shahanaghi (2016) usaron un sistema de detección de colisión. Paes et al. (2017) sugirieron resolver el UA-FLP mediante un algoritmo genético híbrido y una técnica de descomposición. Finalmente, Kang y Chae (2017) emplearon una mejora del algoritmo de búsqueda armónica sugerido por Shayan y Chittilappilly (2004) para abordar el problema de distribución en planta de áreas desiguales.

4. Metodología

Este capítulo detalla el material y las técnicas empleadas para resolver el problema de distribución en planta de áreas desiguales mediante los nuevos enfoques propuestos en esta tesis doctoral.

Tras el estudio y el análisis realizado en el capítulo anterior, en el estado del arte realizado ha quedado patente que son las estrategias metaheurísticas las técnicas más recientes y usadas cuando se trata de resolver el problema de distribución en planta de áreas desiguales. Esto es debido a que ofrecen la posibilidad de abordar problemas complejos ofreciendo muy buenas soluciones. Es por ello, que se ha realizado una búsqueda exhaustiva y un posterior análisis acerca de qué metaheurísticas recientes están aplicándose a optimización de problemas complejos de una manera exitosa. Como resultado, hemos identificado los algoritmos de optimización de arrecifes de coral como una estrategia evolutiva muy prometedora para acometer el problema de distribución en planta de áreas desiguales.

4.1. Algoritmos de optimización de arrecifes de coral

El algoritmo de arrecifes de coral fue propuesto por Salcedo et al. (2013, 2014a) y es conocido en inglés por *Coral Reefs Optimization* (CRO). El CRO es un algoritmo de tipo evolutivo cuya evolución se guía imitando procesos que ocurren en arrecifes de coral reales, como reproducción, la lucha por el espacio o la depredación. Cabe destacar que en el arrecife coralino, existe tanto reproducción sexual como asexual. La primera se produce por el desove (de huevos y esperma) que realizan los corales al agua. La reproducción asexual, se realizará mediante la fragmentación de un coral existente. Como resultado de ambas reproducciones, se forman nuevas larvas que intentarán instalarse en el arrecife de coral mediante la competición por el espacio contra los corales

previos que existen en el arrecife.



Figura 5. Imagen de arrecife de coral.

El CRO es un algoritmo que da como resultado una especie de algoritmo híbrido entre un Algoritmo Evolutivo y un Recocido Simulado (Salcedo-Sanz et al., 2014a) y se ha demostrado que mejora ambas técnicas en diversas instancias en áreas como Telecomunicaciones (Salcedo-Sanz et al., 2014d; Salcedo-Sanz et al., 2014b), Energía (Salcedo-Sanz et al., 2016; Salcedo-Sanz et al., 2014c), Ingeniería estructural (Salcedo-Sanz et al., 2017; Camacho-Gómez et al., 2018) o Aplicaciones Biomédicas (Bermejo et al., 2018; Yan et al., 2019). Recientemente, el CRO también se ha utilizado para diferentes problemas, como de agrupación (Medeiros et al., 2015), de formación de redes neuronales (Yang et al., 2016), el análisis de series temporales (Durán-Rosal et al., 2018) o problemas de asignación de recursos (Ficco et al., 2018), entre otros. En estos trabajos, el CRO se ha aplicado con éxito al alcanzar un excelente rendimiento (Salcedo-Sanz, 2017).

4.2. Funcionamiento del Algoritmo de Optimización de Arrecifes de Coral

Consideraremos Λ como modelo del arrecife con una cuadrícula cuadrada de tamaño $N \times M$ (ver Figura 6). Cada cuadrado o celda ubicado en $\Lambda(i,j)$, es un lugar que puede albergar un coral $\Xi(i,j)$ donde i y j son las coordenadas de la posición en el arrecife. Cada coral es una representación de una solución al problema, en este caso particular, una solución de diseño de planta para el problema de distribución en planta de áreas desiguales. Una vez que tengamos modelado el arrecife y los corales en sí, el proceso del algoritmo se define usando los pasos que se detallan a continuación.

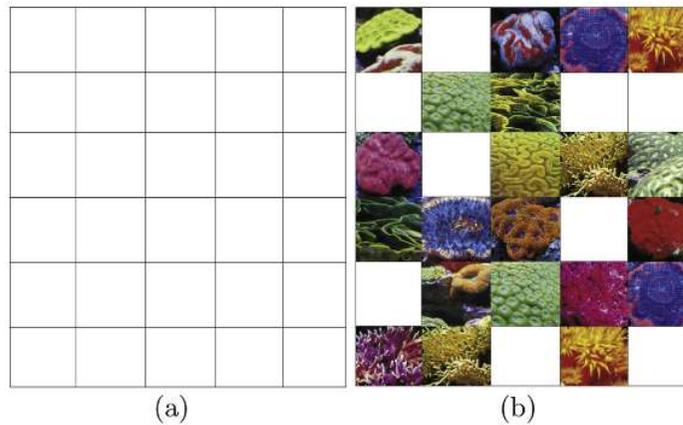


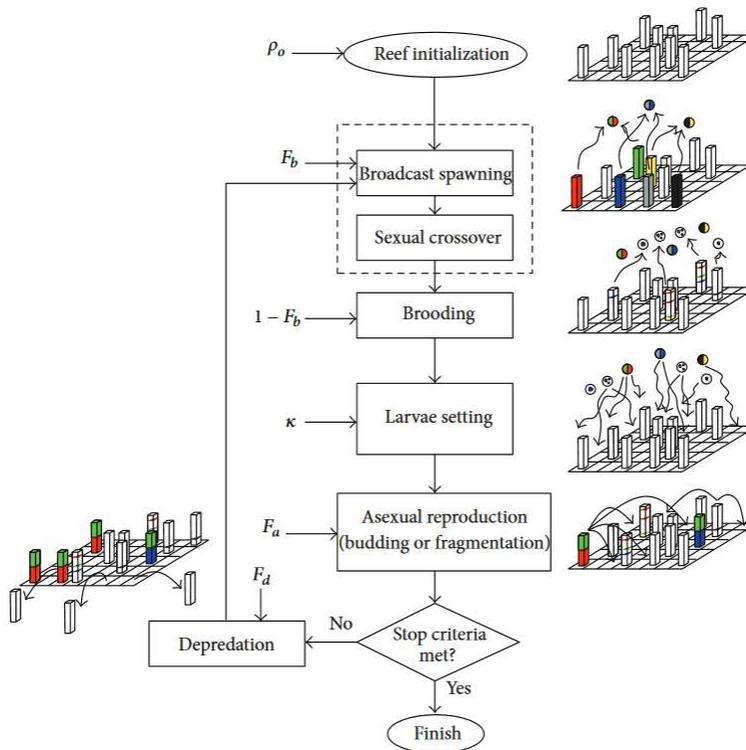
Figura 6. Simulación del arrecife de coral según Salcedo-Sanz et al. (2014); (a) rejilla $N \times M$ (b) corales y huecos en el arrecife.

4.3. Inicialización del algoritmo

Uno de los parámetros más importantes del algoritmo CRO es la cantidad de corales iniciales en el arrecife. Esta tasa está representada por la proporción entre cuadrados vacíos y en uso que existen en el arrecife. Ésta se denota como ρ_0 , de forma que $0 < \rho_0 < 1$. De esta manera, teniendo en cuenta este parámetro, el número inicial de corales se calcula utilizando la siguiente ecuación:

$$\text{Corales Iniciales} = N \times M \times \rho_0$$

Los corales iniciales se generan aleatoriamente y se colocan (también de forma aleatoria) en cuadrados vacíos del arrecife.



 The solution is given by the best coral existing in the reef

Figura 7. Diagrama de flujo del algoritmo CRO. Imagen tomada de Salcedo-Sanz et al. (2014).

Una vez que se inicializan los arrecifes de corales, la simulación de la evolución de los corales comienza con una ejecución iterativa de la reproducción de los corales, que se realiza por medio de diversos operadores hasta que se alcanza el criterio de detención (normalmente, cuando el número requerido de iteraciones han sido satisfechas). Este proceso iterativo ha sido detallado en Salcedo-Sanz et al. (2014a), puede verse en la Figura 7 y se describirá en la siguiente subsección.

4.4. Evolución del algoritmo de coral

La fase de reproducción está definida por diferentes operadores para modelar la

reproducción sexual (que puede ser externa e interna) y la reproducción asexual. Todas estas fases de reproducción generarán nuevos corales a partir de los existentes en el arrecife, que se denominarán larvas. Entre las fases de reproducción sexual y asexual, se encuentra el paso de asentamiento de larvas, donde algunas de las nuevas larvas se fijarán en el arrecife de coral. Finalmente, una fase de depredación eliminará los corales más débiles del arrecife.

Fases del algoritmo

1. *Broadcast spawning* (reproducción sexual externa)

Esta fase se compone de dos pasos. En primer lugar, una serie de corales que existen en el arrecife se seleccionan al azar para ser reproductores. Esta fracción de reproductores se calcula con respecto a la cantidad total de corales existentes en el arrecife. Los corales restantes que no han sido elegidos para ser reproductores serán seleccionados para ser reproducidos en la fase de *Brooding*. En segundo lugar, a partir de los reproductores, el algoritmo seleccionará parejas de corales para reproducirlos.

Esta selección de corales es aleatoria y con reemplazo, una vez que se selecciona una pareja, no se puede volver a seleccionar para reproducirse en el mismo paso. Cada una de las parejas seleccionadas formará dos hijos por cruce sexual. Entonces, una cría será seleccionada al azar como una larva de coral que se libera al agua. El resultado de las larvas se almacena hasta la fase de *Larvae setting*.

2. *Brooding* (reproducción sexual interna)

Los corales restantes de la fase anterior son seleccionados para ser reproducidos por *Brooding*. Esta fase consiste en la formación de una larva de coral mediante una mutación aleatoria en cada uno de los elementos. La larva obtenida es liberada al agua de la misma manera que ha sido realizado en la fase previa. La larva resultante es almacenada hasta la fase de *Larvae Setting*.

3. *Larvae setting* (asentamiento de las larvas en el arrecife)

En este momento, todas las larvas creadas mediante el proceso de *Broadcast spawning* o *Brooding* son almacenadas. El siguiente paso consiste en intentar alojar y hacer crecer esas larvas en el arrecife. Para ello, la función de *fitness* es calculada para todas las larvas y corales que existen en el arrecife (en este caso particular, la función que define es el flujo existente entre los departamentos que componen la planta industrial). Entonces una larva es seleccionada para ser insertada en una posición aleatoria del arrecife. Si esa posición está libre, la larva será instalada ahí. Si no está libre, el *fitness* del coral y la larva serán comparados. De esta forma, si el *fitness* de la larva es mejor que la del coral, el coral será reemplazado por la larva. Si la larva no reemplaza al coral intentará instalarse en el arrecife un número limitado de veces (Salcedo Sanz et al. (2013) sugiere que este número sea 3). Si la larva no se ha podido instalar en esos intentos, será desechada.

4. *Budding or fragmentation* (reproducción asexual)

En esta fase, todos los corales existentes en el arrecife son ordenados en función del nivel de *fitness* que poseen. Entonces una fracción de los mejores de ellos, es duplicada ella misma y se intentará instalar en diferentes posiciones del arrecife. Esto es realizado por el mismo procedimiento que ha sido explicado en el proceso de *Larvae Setting*.

5. *Depredation* (depredación)

Al final de cada iteración del algoritmo, una fracción de los peores corales que existen en el arrecife serán desechados con una probabilidad muy baja. Este proceso, liberará espacio en el arrecife para la siguiente generación del algoritmo de coral.

5. Algoritmo de optimización de arrecifes de coral para abordar el UA-FLP

5.1. Introducción

En este capítulo, se prueba el rendimiento de un algoritmo muy actual y diferente basado en la evolución natural para abordar el problema de distribución en planta de áreas desiguales. Este algoritmo fue propuesto por Salcedo et al. (2013, 2014a) y es conocido como algoritmo de optimización basado en arrecifes de coral (en inglés, *Coral Reefs Optimization*, CRO). El CRO es un algoritmo de tipo evolutivo cuya evolución se guía imitando procesos que ocurren en arrecifes de coral reales, como reproducción, la lucha por el espacio o la depredación. El CRO es un algoritmo que da como resultado una especie de algoritmo híbrido entre un Algoritmo Evolutivo y un Recocido Simulado (Salcedo-Sanz et al., 2014a) y se ha demostrado que mejora ambas técnicas en diversas instancias en áreas como Telecomunicaciones (Salcedo-Sanz et al., 2014d) (Salcedo-Sanz et al., 2014b), Energía (Salcedo-Sanz et al., 2016) (Salcedo-Sanz et al., 2014c), Ingeniería estructural (Salcedo-Sanz et al., 2017) (Camacho-Gómez et al., 2018) o Aplicaciones Biomédicas (Bermejo et al., 2018) (Yan et al., 2019).

Recientemente, el CRO también se ha utilizado para diferentes problemas, como en la categoría de agrupación (Medeiros et al., 2015), la formación de redes neuronales (Yang et al., 2016), el análisis de series temporales (Durán-Rosal et al., 2018) o los problemas de asignación de recursos (Ficco et al., 2018), entre otros. En estos trabajos, el CRO se ha aplicado con éxito al alcanzar un excelente rendimiento en el problema probado (Salcedo-Sanz, 2017). Este capítulo de la tesis trata de investigar el rendimiento obtenido cuando se aplica el algoritmo de optimización de los arrecifes de coral al problema de distribución en planta de áreas desiguales. Hasta el momento, no ha sido aplicada la técnica de

optimización basada en arrecifes de coral para resolver el UA-FLP.

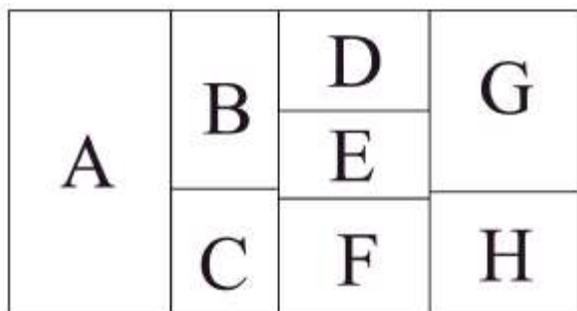


Figura 8. Ejemplo de un diseño basado en FBS.

5.2. Enfoque propuesto

Para abordar el problema de distribución en planta de áreas desiguales, se propone un nuevo enfoque de optimización basado en el algoritmo de optimización de arrecifes de coral que considera el coste de flujo de material como criterio de optimización y, el factor de forma como restricción. A continuación, se describirá la estructura y la implementación del algoritmo.

5.2.1 Codificación del individuo

Para codificar a un individuo del arrecife en el CRO, ha sido utilizada la estructura del cromosoma sugerida por Gomez et al. (2003). Se ilustra en la Figura 9. Esta estructura de codificación está formada por dos segmentos diferentes. El primero ilustra la secuencia de departamentos o instalaciones en el diseño de la planta industrial, el sentido en el que se leen es de arriba a abajo en cada bahía y de izquierda a derecha en la planta industrial. En el primer segmento se emplea una permutación entera de 1 a n (siendo n el número total de departamentos o instalaciones que existen en la planta). El segundo segmento ofrece la información acerca de dónde están los cortes que delimitan las bahías de la planta. Este segmento está compuesto por $(n - 1)$ elementos que tienen valores binarios. De modo que, si aparece el valor "1" en una determinada

posición del segmento, significa que el departamento en la misma posición del primer segmento, es el último elemento de la bahía. De lo contrario, aparecerá el valor '0' en esa posición del segmento. La Figura 9 proporciona el cromosoma de un individuo asociado a la representación de la planta industrial ofrecido en la Figura 8.

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

Figura 9. Ejemplo de un cromosoma con el diseño de una planta.

5.2.2 Función objetivo

Armor y Buffa (1963) establecieron el UA-FLP por primera vez. El problema se define mediante un diseño rectangular de dimensiones ($W \times H$) que son fijas. Además, hay un grupo de instalaciones o departamentos con un área determinada (A_i). La suma de las áreas de los departamentos debe ser menor o igual que el área total de la planta rectangular. Lo podemos ver en la siguiente ecuación:

$$\sum_i^n A_i \leq W \times H$$

El objetivo del problema es colocar todos los departamentos en la planta optimizando un criterio dado y teniendo en cuenta que no se permite la superposición entre departamentos o instalaciones.

En Aiello et al. (2012) se afirma que el problema de distribución en planta de áreas desiguales persigue como objetivo principal la minimización del coste de flujo de material entre los departamentos o instalaciones. La función objetivo utilizada en los algoritmos evolutivos para evaluar las soluciones de los UA-FLPs se basa, por lo tanto, en el flujo de material. Además, para guiar el proceso de búsqueda hacia individuos factibles, ha sido utilizada una función de penalización propuesta por Tate y Smith (1995). De esta manera, para cada solución en el algoritmo, se define un grado de penalización, que es proporcional

al número de instalaciones que conforman la planta y que no satisfacen la restricción de la relación de aspecto (ya sea que no se satisface la relación de aspecto máxima o la longitud mínima requerida del lado). Estas instalaciones se consideran como inviables. La función *fitness* que minimiza el material de flujo es la siguiente:

$$g(\mathbf{x}) = \sum_i^n \sum_j^n f_{ij} d_{ij} + (D_{inf})^k (V_{feas} - V_{all})$$

donde n es el número de departamentos o instalaciones en la planta, f_{ij} es el flujo de material entre los departamentos i y j , d_{ij} es la distancia Manhattan entre las instalaciones i y j , D_{inf} es la cantidad de instalaciones que no son factibles (es decir, no cumplen la relación de aspecto requerida, V_{feas} es el mejor valor de la función objetivo factible que se haya logrado hasta el momento, V_{all} es el mejor valor de la función objetivo (sea factible o no) que se ha logrado hasta el momento, k es un parámetro de penalización que se ajusta al valor de la función de penalización (se ha puesto a '3' como en Tate y Smith, 1995).

5.3. Algoritmo de Optimización basado en Arrecifes de Coral

El Algoritmo de Optimización basado en Arrecifes de Coral (CRO) fue propuesto recientemente por Salcedo-Sanz et al. (2014a). Este enfoque es una especie de algoritmo de tipo evolutivo que imita la evolución de los arrecifes de coral y los diferentes procesos que ocurren en estos ecosistemas. Consideraremos Λ como modelo del arrecife con una cuadrícula cuadrada de tamaño $N \times M$ (ver Figura 10). Cada cuadrado o celda ubicado en $\Lambda(i,j)$, es un lugar que puede albergar un coral $\Xi(i,j)$ donde i y j son las coordenadas de la posición en el arrecife. Cada coral es una representación de una solución al problema, en este caso particular, una solución de diseño de planta para el problema de distribución en planta de áreas desiguales. Una vez que tengamos modelado el arrecife y los

corales en sí, el proceso del algoritmo se define usando los pasos que se detallan a continuación.

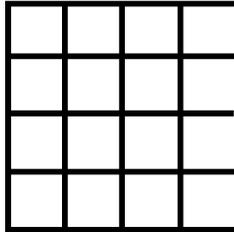


Figura 10. Ejemplo de un arrecife de coral de tamaño 4x4.

5.3.1 Inicialización del algoritmo

Uno de los parámetros más importantes del algoritmo CRO es la cantidad de corales iniciales en el arrecife. Esta tasa está representada por la proporción entre cuadrados vacíos y en uso que existen en el arrecife que se denota como ρ_0 , de forma que $0 < \rho_0 < 1$. De esta manera, teniendo en cuenta este parámetro, el número inicial de corales se calcula utilizando la siguiente ecuación:

$$\text{Corales Iniciales} = N \times M \times \rho_0$$

La Figura 11 ilustra un arrecife de coral inicializado con corales aleatorios en una proporción de '0.5' entre cuadrados vacíos y en uso, esto es con $\rho_0 = 0.5$.

Los corales iniciales se generan aleatoriamente y se colocan (también de forma aleatoria) en cuadrados vacíos del arrecife. Este paso se resume en el siguiente algoritmo:

Reef initialization

Input Reef size (width and height) and occupation rate

Output Initial reef population

```
1: procedure INITIALIZE REEF( $n, m, \rho_0$ )           ▷ Coral Reef initialization
2:    $reef\_size \leftarrow n \times m$ 
3:    $k \leftarrow reef\_size \times \rho_0$              ▷ Number of initial corals
4:   for  $k$  times do
5:     generate random coral
6:     place coral in random empty reef position
7:   end for
8:   return initial reef
9: end procedure
```

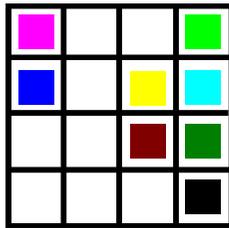


Figura 11. Ejemplo de un arrecife de coral con individuos aleatorios insertados y $\rho_0 = 0.5$

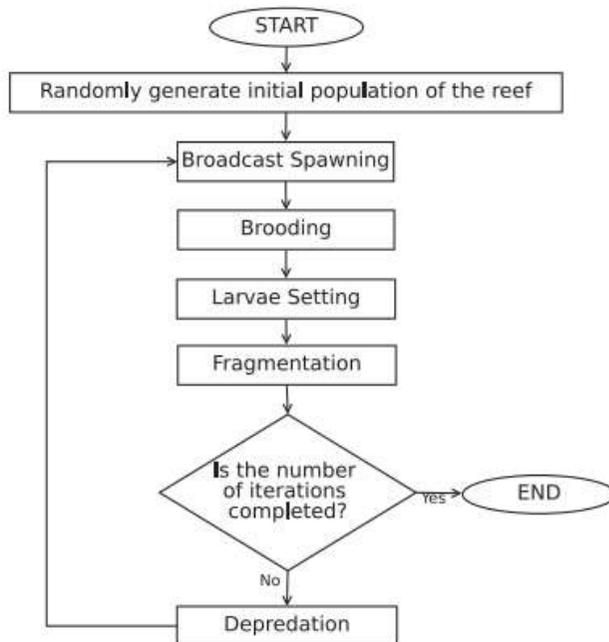


Figura 12. Diagrama de flujo del algoritmo CRO propuesto

Una vez que se inicializan los arrecifes de coral, la simulación de la evolución de los corales comienza con una ejecución iterativa de la reproducción de los corales, que se realiza por medio de diversos operadores hasta que se alcanza el criterio de detención (en este caso particular, cuando el número requerido de iteraciones han sido satisfechas). Este proceso iterativo ha sido detallado en Salcedo-Sanz et al. (2014a) y se describirá en la siguiente subsección.

5.3.2 Evolución del algoritmo de coral

La fase de reproducción está definida por diferentes operadores para modelar la reproducción sexual (que puede ser externa e interna) y la reproducción asexual. Todas estas fases de reproducción generarán nuevos corales a partir de los existentes en el arrecife, que se denominarán larvas. Entre las fases de reproducción sexual y asexual, se encuentra el paso *Larvae Setting*, donde algunos de los nuevos elementos de larvas se fijarán en el arrecife de coral. Finalmente, una fase de *Depredation* eliminará los corales más débiles del arrecife. La Figura 12 resume todo el proceso del algoritmo CRO mediante un diagrama de flujo. Además, el siguiente pseudocódigo muestra todo el proceso:

CRO algorithm.

Input Algorithm's control parameters
Output Feasible solution with best *fitness*

```

1: procedure CRO( $n, m, \rho_0, F_b, F_u, F_d, P_d$ ) ▷ Coral Reef Optimization
   algorithm
2:   initialize reef with size  $n \times m$  and occupation rate  $\rho_0$ 
3:   repeat
4:     reproduce corals fraction  $F_b$  by broadcast spawning
5:     reproduce corals fraction  $1 - F_b$  by brooding
6:     larvae evaluation
7:     larvae setting
8:     reproduce best corals fraction  $F_u$  by asexual reproduc-
   tion
9:     predation of  $F_d$  worst reef corals with  $P_d$  probability
10:  until stop condition
11:  return best feasible solution
12: end procedure

```

El diagrama de flujo del algoritmo CRO con sus diferentes fases, se detalla a continuación:

1. *Broadcast spawning* (reproducción sexual externa)

Esta fase se compone de dos pasos. En primer lugar, una serie de corales que existen en el arrecife, denotados por ρ_k , se seleccionan al azar para ser reproductores. Esta fracción de reproductores se calcula con respecto a la cantidad total de corales existentes en el arrecife y se denota como F_b . Los corales restantes que no han sido elegidos para ser reproductores ($1 - F_b$) serán seleccionados para ser reproducidos en la fase de *Brooding*. En segundo lugar, a partir de los reproductores (ρ_k), el algoritmo seleccionará parejas de corales para reproducirlos.

Esta selección de corales es aleatoria y con reemplazo, una vez que se selecciona una pareja, no se puede volver a seleccionar para reproducirse en el mismo paso. Cada una de las parejas seleccionadas formará dos hijos por cruce sexual. Específicamente en el enfoque propuesto, el operador de cruce *Partial Mapped Crossover* (PMX) propuesto por Goldberg y Robert (1985), se utiliza para el segmento de secuencia de instalaciones, y el operador de cruce *Crossover One Point* (Holland, 1992) se aplica sobre el segmento de cortes. En este punto, una cría será seleccionada al azar y pasará a ser una larva de coral que se libera al agua. Este proceso de cruce es ilustrado en la Figura 14, donde se muestra cómo la representación de la planta cambia durante el algoritmo CRO. El resultado de las larvas se almacena hasta la fase de *Larvae setting*. A continuación, se detalla el pseudocódigo y en la Figura 13 se detalla la fase de *Broadcast spawning*:

Broadcast spawning.

Input Coral reef, External sexual reproduction rate
Output Generated larvae set

```
1: procedure BROADCAST SPAWNING(reef,  $F_b$ )
2:    $\rho_k \leftarrow \text{coral\_num} \times F_b$    ▷ Number of corals to reproduce by
   broadcast spawning
3:   select  $\rho_k$  corals from reef
4:   pair selected corals
5:   for each coral pair do
6:     apply crossover
7:     add generated solution to larvae set
8:   end for
9:   return generated larvae set
10: end procedure
```

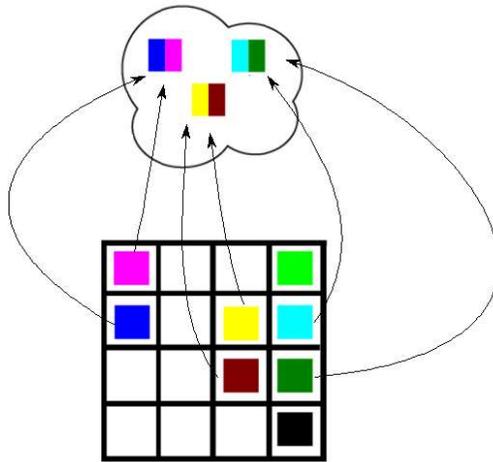


Figura 13. Diagrama que muestra la fase de Broadcast spawning.

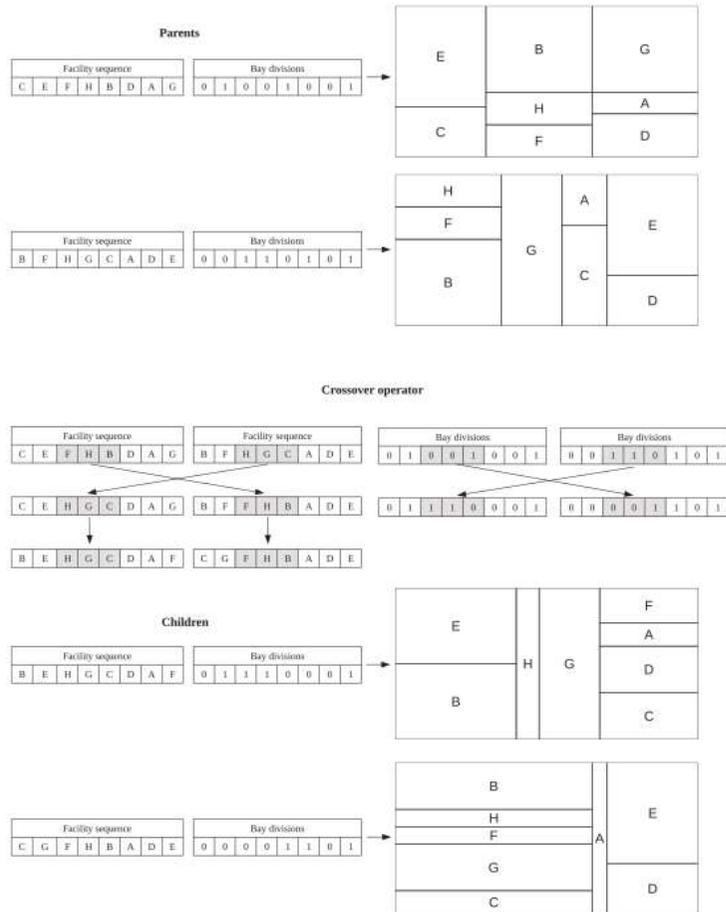


Figura 14. Diagrama que muestra el proceso de Crossover en la fase de Broadcast spawning

2. *Brooding* (reproducción sexual interna)

Los corales restantes de la fase anterior son seleccionados para ser reproducidos por *Brooding*. Esta fase consiste en la formación de una larva de coral mediante una mutación aleatoria en cada uno de los elementos.

La larva obtenida es liberada al agua de la misma forma que en la fase previa. La Figura 15 muestra la reproducción por *Brooding* sobre dos de los corales que no han sido seleccionados para ser reproducidos en la fase previa. En la misma, puede observarse como las representaciones de la planta cambian de nuevo durante el proceso del algoritmo. La larva resultante es almacenada hasta la fase de *Larvae setting*. Esta fase, se detalla en el siguiente algoritmo:

Brooding.

Input Coral reef
Output Generated larvae set

```
1: procedure BROODING(reef)
2:   select all corals not reproduced by broadcast spawning from reef
3:   for each selected coral do
4:     apply mutation
5:     add generated solution to larvae set
6:   end for
7:   return generated larvae
8: end procedure
```

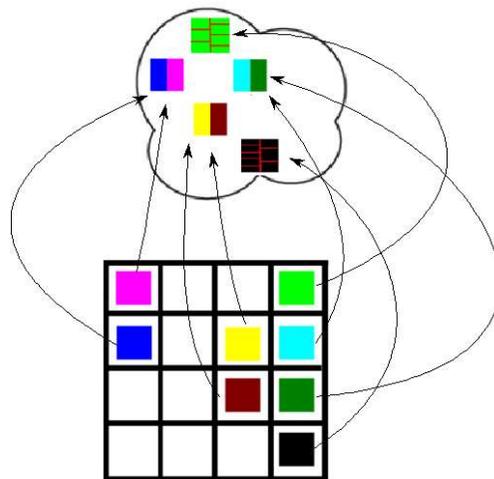


Figura 15. Diagrama que muestra la fase de *Brooding*.

3. Larvae setting (asentamiento de las larvas en el arrecife)

En este momento, todas las larvas creadas mediante el proceso de *Broadcast spawning* o *Brooding* son almacenadas. El siguiente paso consiste en intentar alojar y hacer crecer esas larvas en el arrecife. Para ello, la función de *fitness* es calculada para todas las larvas y corales que existen en el arrecife (en este caso particular, la función que define es el flujo existente entre los departamentos que componen la planta industrial). Entonces una larva es seleccionada para ser insertada en una posición aleatoria del arrecife. Si esa posición está libre, la larva será instalada ahí. Si no está libre, el *fitness* del coral y la larva serán comparados. De esta forma, si el *fitness* de la larva es mejor (es decir, tiene un valor de flujo de material más bajo) que el coral, el coral será reemplazado por la larva. Si la larva no reemplaza al coral (tiene un valor más alto de coste de flujo de material), intentará instalarse en el arrecife k veces (este número es 3 como sugiere Salcedo Sanz et al. (2013)). Si la larva no se ha podido instalar en esos intentos, será desechada. Este mecanismo es explicado mediante la Figura 16 y el siguiente algoritmo:

```
Larvae setting.


---


Input Coral reef, larvae set
Output Updated reef
1: procedure LARVAE_SETTING(reef, larvae)
2:   for each larvae do
3:     placed  $\leftarrow$  False
4:      $k \leftarrow 3$   $\triangleright$  Number of attempts to settle in the reef
5:     while not placed and  $k > 0$  do
6:       pos  $\leftarrow$  random reef position
7:       if pos is empty or larva fitness is better than resi-
         dent's then
8:         larva settles in pos
9:         placed  $\leftarrow$  True
10:      else
11:         $k \leftarrow k - 1$ 
12:      end if
13:    end while
14:  end for
15:  return reef
16: end procedure
```

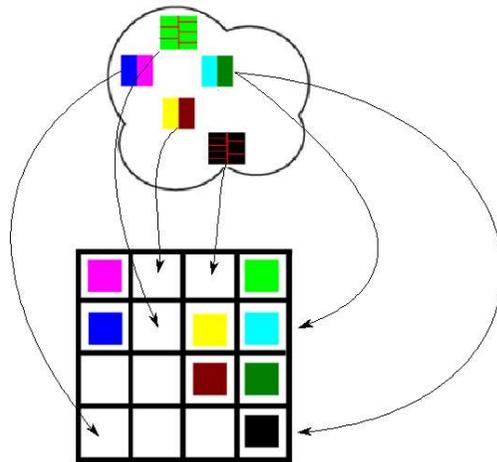


Figura 16. Diagrama que muestra la fase de Larvae setting.

4. Budding or fragmentation (reproducción asexual)

En esta fase, todos los corales existentes en el arrecife son ordenados en función del nivel de *fitness* que poseen. Entonces una fracción de ellos denotada por F_a , es duplicada ella misma y se intentará instalar en diferentes posiciones del arrecife. Esto es realizado por el mismo procedimiento que ha sido explicado en el proceso de *Larvae setting*. La reproducción es asexual y se ilustra por medio de la Figura 17 y el siguiente algoritmo:

Asexual reproduction.

Input Coral reef, Asexual reproduction rate
Output Updated reef

```

1: procedure ASEXUAL_REPRODUCTION(reef,  $F_a$ )
2:    $n_a \leftarrow coral\_num \times F_a$       ▷ Number of corals to duplicate
3:   select the best  $n_a$  corals from reef
4:   for each selected coral do
5:     settle coral in reef      ▷ Same procedure as larvae_setting
6:   end for
7:   return reef
8: end procedure

```

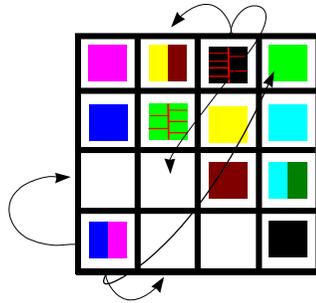


Figura 17. Diagrama que muestra la fase de Budding.

5. Depredation (depredación)

Al final de cada iteración del algoritmo, una fracción de los peores corales que existen en el arrecife denotada por F_d serán desechados con una probabilidad muy baja denotada P_d .

Depredation.

Input Coral reef, depredation fraction, depredation probability

Output Updated reef

```

1: procedure DEPREDATION(reef,  $F_d$ ,  $P_d$ )
2:    $n_d \leftarrow coral\_num \times f_d$       ▷ Number of corals that may be
    predated
3:   select the worst  $n_d$  corals from reef
4:   for each selected coral do
5:     if  $random(0.0, 1.0) \leq P_d$  then
6:       remove coral from reef
7:     end if
8:   end for
9:   return reef
10: end procedure

```

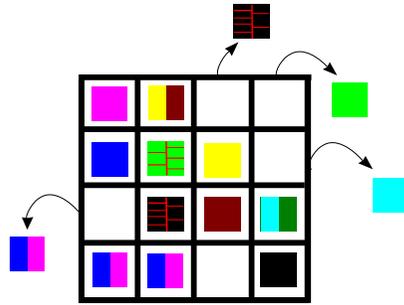


Figura 18. Diagrama que muestra la fase de Depredation.

Este proceso, liberará espacio en el arrecife para la siguiente generación del algoritmo de coral. La fase de *Depredation* es ilustrada en la Figura 18 y el algoritmo arriba expuesto.

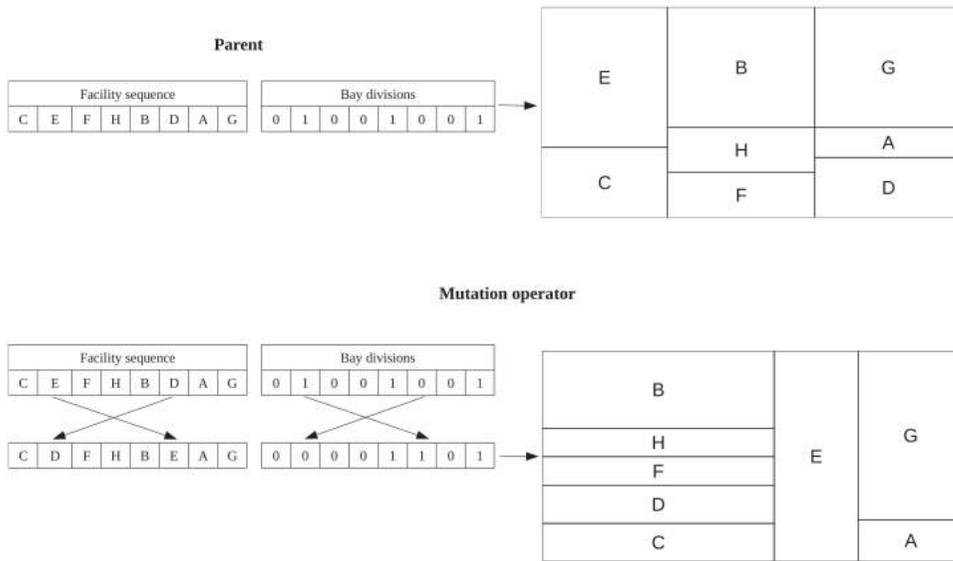


Figura 19. Diagrama que muestra el proceso de mutación y Brooding.

5.3.3 Experimentación y resultados

El rendimiento de la propuesta CRO es evaluado comparándolo con otros algoritmos del estado del arte que abordan el problema UA-FLP en otros artículos de la bibliografía. El conjunto de problemas de distribución en planta

de áreas desiguales que han sido seleccionados son: *Slaughterhouse* abordado en Salas-Morera et al. (1996); *CartonPacks* y *ChoppedPlastic* en García-Hernández et al. (2013a); O7, O8 y O9, descrito por Meller et al. (1998); VC10 (restricciones lado mínimo y de relación de aspecto) ilustrado en Camp et al. (1992); MB12 explicado por Bozer y Meller (1997); Ba12 abordado en Bazaraa (1975); Ba14 presentado en Komarudin y Wong (2010) del problema descrito en Bazaraa (1975); Ma15 (con dos restricciones de dos formas diferentes) en Bozer et al. (1994); AB20 detallado por Armour y Buffa (1963); SC30, una modificación tomada de Komarudin y Wong (2010) del problema descrito por Liu y Meller (2007); SC35 de Liu y Meller (2007); y DU62 descrito por Dunker et al. (2003).

Las características de los problemas UA-FLP seleccionados han sido descritas en la Tabla 2. Esta información hace referencia al nombre del problema de referencia UA-FLP, el número de instalaciones que lo componen, el ancho de la planta industrial, el alto de la planta industrial, el factor de forma (siendo α el máximo valor de factor de forma, y $lmin$ el valor mínimo del lado permitido), y finalmente las referencias de los trabajos fuentes de dónde se han tomado los problemas. Cabe destacar que como medida de distancia se ha usado por defecto la distancia Manhattan. Sin embargo, la distancia Euclidiana también ha sido aplicada en algunos problemas como es el caso de *Slaughterhouse*, *CartonPacks* y *ChoppedPlastic*.

El algoritmo CRO propuesto depende fuertemente de un conjunto de parámetros de entrada. Estos parámetros han sido afinados de una manera empírica. Así, se han realizado diferentes pruebas para determinar la mejor combinación de parámetros posibles del algoritmo CRO cuando resuelve el problema UA-FLP. En la Tabla 3 se ilustran los posibles valores de los parámetros de entrada del algoritmo y cuales de ellos ofrecen los mejores resultados. Teniendo en cuenta los valores que han sido presentados en la Tabla 3, se ha realizado un diseño factorial completo de experimentación, probando diferentes problemas de referencia UA-FLP.

<i>Problem name</i>	<i>Fac.</i>	<i>W × H</i>	<i>Aspect ratio</i>	<i>Reference</i>
Slaughterhouse	12	51.14 × 30.00	$\alpha=4$	Salas-Morera et al. (1996)
CartonPacks	11	20.00 × 14.50	$\alpha=4$	García-Hernández et al. (2013a)
ChoppedPlastic	10	10.00 × 30.00	$\alpha=4$	García-Hernández et al. (2013a)
O7	7	8.54 × 13.00	$\alpha=4$	Meller et al. (1998)
O8	8	11.3 × 13.00	$\alpha=4$	Meller et al. (1998)
O9	9	12.00 × 13.00	$\alpha=4$	Meller et al. (1998)
vC10Ra	10	25.00 × 51.00	$\alpha=5$	van Camp et al. (1992)
Vc10Rs	10	25.00 × 51.00	<i>Min.side=5</i>	van Camp et al. (1992)
Ba12	12	6.00 × 10.00	<i>Min.side=1</i>	Bazaraa (1975)
MB12	12	6.00 × 8.00	$\alpha=4$	Bozer and Meller (1997)
Ba14	14	7.00 × 9.00	<i>Min.side={1,0}</i>	Komarudin and Wong (2010)
Ma15	15	15.00 × 15.00	$\alpha=5$	Bozer et al. (1994)
Ma15s	15	15.00 × 15.00	<i>Min.side=1</i>	Bozer et al. (1994)
AB20	20	2.00 × 3.00	$\alpha=5$	Armour and Buffa (1963)
SC30	30	12.00 × 15.00	$\alpha=5$	Liu and Meller (2007)
SC35	35	16.00 × 15.00	$\alpha=4$	Liu and Meller (2007)
Du62	62	Arbitrary × Arbitrary	$\alpha=4$	Dunker et al. (2003)

Tabla 2. Características de los problemas testados bien conocidos.

UA-FLP	Chosen values			Tested values		
	O9	Ma15s	SC30	Combination of:		
$N \times M$	25 × 25	25 × 25	25 × 25	10 × 10	15 × 15	25 × 25
ρ_0	c0.4	0.4	0.4	0.4	0.5	0.6
F_b	c0.9	0.9	0.9	0.8	0.85	0.9
F_a	c0.1	0.1	0.2	0.1	0.15	0.2
F_d	0.1	0.1	0.1	0.01	0.05	0.1
P_a	c0.1	0.1	0.1	0.01	0.05	0.1

Tabla 3. Parámetros CRO seleccionados.

<i>Problem name</i>	<i>OFV Best</i>	<i>OFV Mean</i>	<i>CPU(s)</i>
Slaughterhouse	3487.12	3487.12	78.00
CartonPacks	80.91	80.91	74.00
ChoppedPlastic	265.77	265.77	65.00
O7	134.16	134.16	4.00
O8	245.48	245.48	24.00
O9	239.44	239.44	49.00
vC10Ra	20142.13	20576.93	61.00
Vc10Rs	22897.65	22898.65	63.00
Ba12	8021.0	8103.96	87.00
MB12	125.00	125.00	81.00
Ba14	4665.93	4731.23	92.00
Ma15	26800.63	26972.95	104.00
Ma15s	22871.97	23034.88	106.00
AB20	5243.95	5250.02	202.00
SC30	3519.44	3566.27	622.00
SC35	4263.3	4409.34	552.00
Du62	713876.55	3719342.03	871.00

Tabla 4. Estadísticas de los resultados obtenidos por el algoritmo CRO.

Específicamente, los problemas seleccionados para ser representativos y hacer las pruebas de afinamiento de parámetros han sido los siguientes: O9 de Meller et al. (1998), Ma15d de Bozer et al. (1994) y SC30 tomado de Liu y Meller (2007). Estos problemas han sido seleccionados como representativos para tener en cuenta los diferentes tamaños que existen en los problemas UA-FLP de referencia (pequeño, medio y grande). Una comparación entre las soluciones que han sido alcanzadas ha sido llevada a cabo para seleccionar qué parámetros se ajustan mejor. La mejor configuración para cada uno de los problemas representativos ha sido mostrada en la columna de valor elegido en la Tabla 3. La experimentación ha sido replicada 5 veces para cada uno de estos problemas como se realiza en Komarudin y Wong (2010) con un criterio de parada de 10000 iteraciones como máximo y 500 iteraciones sin mejora. El algoritmo CRO ha sido codificado usando Python 2.7.13., y todos los experimentos han sido realizados usando un Intel Core i5 6200u.

5.4. Resultados

La Tabla 4 presenta los resultados estadísticos que han sido obtenidos por la propuesta CRO. Para cada problema seleccionado se muestra: el mejor valor obtenido de la función objetivo, la media de la función objetivo y el tiempo en segundos necesario para obtener el mejor valor de la función objetivo. Mirando la Tabla 4, podemos extraer que el algoritmo CRO es robusto porque el porcentaje de error entre el mejor valor de la función objetivo y la media es relativamente bajo. Este error normalmente aumenta en función de cuánto aumente el número de instalaciones en la planta industrial.

Si nos fijamos en el tiempo de CPU, See y Wong (2008) establece que en el diseño de la distribución en planta el tiempo de CPU no es un tema extremadamente importante. En este contexto, nuestra propuesta es capaz de alcanzar soluciones satisfactorias en un tiempo de CPU razonable si lo comparamos con otras propuestas alternativas (como por ejemplo con

Komarudin y Wong, 2010 y Palomo-Romero et al., 2017, entre otros). Con el objetivo de comparar el rendimiento alcanzado por esta propuesta CRO, se realiza una comparación en profundidad considerando los resultados obtenidos por otros autores cuando resuelven el problema de UA-FLP. Además los resultados han sido comparados tanto con trabajos que usan la estructura FBS como la estructura STS para representar la planta industrial. Toda esta información es detallada en las Tablas 5 y 7.

<i>Problem</i>	<i>Best known</i>	<i>Layout represent.</i>	<i>Reference</i>	<i>Best known FBS</i>	<i>Reference</i>
Slaughterhouse	3487.12	FBS	This approach	3487.12	This approach
CartonPacks	89.02	FBS	This approach	89.02	This approach
ChoppedPlastic	265.77	FBS	This approach	265.77	This approach
O7	131.56	STS	Gonçalves and Resende (2015)	134.16	This approach
O8	243.12	STS	Wong and Komarudin (2010)	245.48	This approach
O9	236.14	STS	Kang and Chae (2017)	239.44	This approach
Vc10Ra	18522.79	STS	Kang and Chae (2017)	20142.13	This approach
Vc10Rs	19951.17	STS	Gonçalves and Resende (2015)	22897.65	This approach
Ba12	8021.0	FBS	This approach	8021.0	This approach
MB12	125.00	FBS	This approach	125.00	This approach
Ba14	4628.79	STS	Gonçalves and Resende (2015)	4665.93	This approach
Ma15a	26800.63	FBS	This approach	26800.63	This approach
Ma15s	22871.97	FBS	This approach	22871.97	This approach
AB20	4959.11	STS	Kang and Chae (2017)	5243.95	This approach
SC30	3352.70	STS	Kang and Chae (2017)	3443.34	Kulturel-Konak and Konak (2011)
SC35	3316.77	STS	Gonçalves and Resende (2015)	3613.11	Kulturel-Konak and Konak (2011)
Du62	3635307.0	STS	Kang and Chae (2017)	3641497.00	Kulturel-Konak and Konak (2011)

Tabla 5. Resumen de los problemas conocidos y sus soluciones FBS.

<i>Problem</i>	<i>CRO</i>	<i>Palomo(2017)</i>	<i>Kulturel-Konak (2011)</i>	<i>Kulturel-Konak (2012)</i>	<i>Wong (2010)</i>	<i>Enea (2005)</i>
Slaughterhouse	3487.12	-	-	-	-	3854.00
CartonPacks	89.02	-	-	-	-	94.10
ChoppedPlastic	265.77	-	-	-	-	377.18
O7	134.16	134.19	-	-	-	-
O8	245.48	245.51	-	-	-	-
O9	239.44	241.06	-	-	241.06	-
Vc10Ra	20142.13	20142.13	20142.13	21463.07	21463.1	-
Vc10Rs	22897.65	22899.65	22899.65	22899.65	22899.65	-
Ba12	8021.0	8435.83	8129.00	8021.0	8786.00	-
MB12	125.00	125.00	-	-	-	-
Ba14	4665.93	4665.93	4780.91	4739.74	5004.55	-
Ma15a	26800.63	-	27545.27	-	27545.30	-
Ma15s	22871.97	-	23197.80	-	23197.80	-
AB20	5243.95	5256.10	5336.36	5297.6	5677.83	-
SC30	3519.44	3613.11	3443.34	3563.95	-	-
SC35	4263.3	3885.29	3700.75	-	-	-
Du62	3713876.55	-	3641497.00	-	-	-

Tabla 6. Comparación de los mejores resultados obtenidos por CRO y otros enfoques alternativos de FBS.

<i>Problem name</i>	<i>Best sol.</i>	<i>FBS Diff(%)</i>	<i>STS Diff(%)</i>	<i>Solution by CRO</i>
Slaughterhouse	3487.12	10.52	10.52	1 8-2 4-5 12-7-6 11-3-10-9
CartonPacks	89.02	5.70	5.70	2-6-11 9-10-1-8 5-4-7-3
ChoppedPlastic	265.77	41.61	41.61	10-2-3-4-5-6-7 1-9-8
O7	134.16	0.02	- 1.93	3-5-7-8 1-4-6-2
O8	245.48	0.02	- 0.96	5-8-6-3 2-1-4-7
O9	239.44	0.69	- 1.37	5-9-6-2-3 8-1-4-7
Vc10Ra	20142.13	0.00	- 8.03	5-8-10-9-2-6-1 7-3
Vc10Rs	22871.97	1.43	- 12.86	7 5-10-9 3 11 12 8 6 4-2 1
Ba12	8021.0	0.00	0.00	4-10 9-5-7 3 2-12 1 11-8-6
MB12	125.00	0.00	0.00	12 10-7-3-4-2-8-6-5-1-9 11
Ba14	4665.93	0.00	- 0.79	7-11-5 10 1 3 9 4-2 13-1-4-12-8-6
Ma15a	26800.63	2.78	2.78	6-11 2-1-8-7-13 4-15-3 5-14-12-10-9
Ma15s	22871.97	1.43	1.43	9-10-12-15-6-8-11-7 14-4-3-13 5-2 1
AB20	5243.95	0.23	- 11.20	1-16-11 17-13 12-9-15 3-14 19-10 6-4-2-7-20 18-5
SC30	3519.44	- 2.16	- 4.73	19-34-30-10 2-6-22-26 17-25-29-35-28-21 3-4-1-20
SC35	3885.29	- 4.74	- 14.63	19-34-30-10 2-6-22-26 17-25-29-35-28-21 3-4-1-20
Du62	3713876.55	- 1.9	- 2.11	19-34-30-10 2-6-22-26 17-25-29-35-28-21 3-4-1-20 23-33-18-24-32 13-15-7-11-8 12-34-9 14-31-5-27-16

Tabla 7. Resumen de los resultados obtenidos por la propuesta CRO.

La Tabla 5 ofrece para cada problema probado la siguiente información: el mejor resultado conocido y su representación en planta asociada, así como, la referencia del trabajo donde ha sido obtenida. De forma adicional, teniendo en cuenta que se ha usado la estructura de representación de bahías flexibles en la propuesta, la Tabla 6 también presenta para cada uno de los problemas mostrados el mejor resultado conocido y su referencia asociada considerando exclusivamente la estructura de bahías flexibles como estructura de representación. En esta tabla, se ha resaltado en negrita que los resultados alcanzados por la propuesta CRO son los mejores resultados conocidos hasta el momento. De esta manera, mirando las Tablas 5 y 7, puede verse que la propuesta CRO alcanza o mejora la mejor solución obtenida en 7 casos de los 17 problemas que han sido probados cuando consideramos ambas estructuras es decir, estructura de representación de árbol de cortes y estructura de bahías flexibles. Esto mismo ocurre en 14 casos de los 17 problemas de referencia probados cuando se considera exclusivamente la estructura de bahías flexibles como estructura de representación de la planta industrial. En los casos restantes, la propuesta CRO es capaz de alcanzar soluciones muy próximas a las mejores soluciones obtenidas hasta el momento.

De acuerdo con Kang y Chae (2017) la estructura de representación de árbol de cortes no puede compararse con las soluciones que se obtienen por medio de la

estructura de bahías flexibles. Esa es la razón por la que en la mayoría de los casos las soluciones obtenidas usando la estructura de representación de árbol de cortes obtienen mejores resultados que la estructura de representación de bahías flexibles. Así, se considera interesante analizar los resultados comparándolos en profundidad con otros trabajos que también usen la estructura de bahías flexibles en sus propuestas como estructura de representación. En particular, estas propuestas son tomadas de Palomo-Romero et al. (2017), Kulturel-Konak y Konak (2011), Kulturel-Konak (2012), Wong y Komarudin (2010) y Enea et al. (2005). La Tabla 6 muestra los resultados obtenidos por la propuesta CRO y las propuestas previamente enunciadas. Para cada problema probado, se han resaltado en negrita las mejores soluciones. Primero, la Tabla 6 nos muestra como la propuesta CRO es capaz de alcanzar mejores resultados en la mayoría de los casos UA-FLP de referencia probados. Como se ha dicho previamente, el algoritmo CRO mejora los resultados en 14 de los problemas probados. Específicamente la propuesta CRO mejora a Enea et al. (2005) en todos los casos comparados: *Slaughterhouse*, *CartonPacks* y *ChoppedPlastic*. El algoritmo CRO también es capaz de obtener mejores resultados que el algoritmo de Wong y Komarudin (2010) en todos los casos si lo comparamos con el algoritmo propuesto. También comparando la propuesta con el algoritmo de Kulturel-Konak (2012), el algoritmo CRO alcanza mejores soluciones en todos y cada uno de los seis problemas UA-FLP probados. Adicionalmente, esta propuesta es capaz de alcanzar los mismos o mejores resultados que la propuesta de Kulturel-Konak y Konak (2011) en 7 de los 10 problemas UA-FLP analizados. Finalmente, la propuesta CRO es capaz de igualar o mejorar las soluciones obtenidas por Palomo-Romero et al. (2017) en 10 de los 11 casos UA-FLP probados.

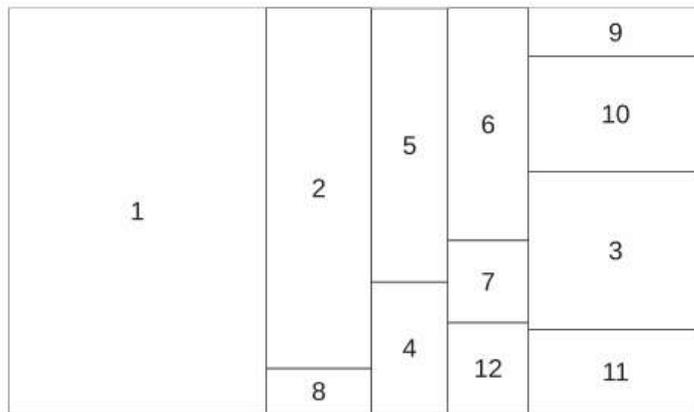


Figura 20. El mejor diseño alcanzado por el enfoque CRO propuesto en el Slaughterhouse UA-FLP.

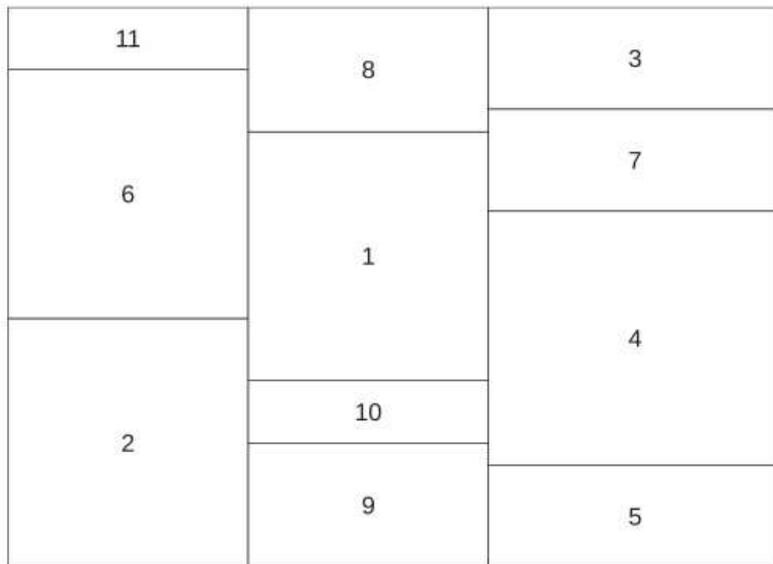


Figura 21. El mejor diseño alcanzado por el enfoque CRO propuesto en el CartonPacks UA-FLP.

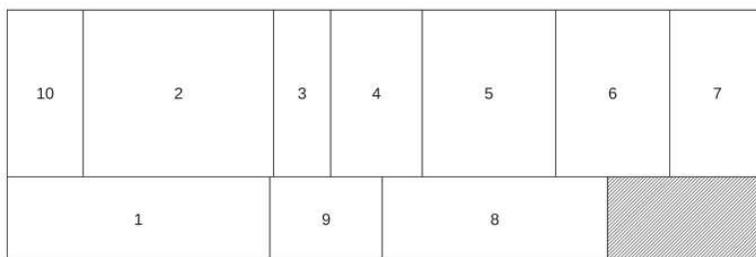


Figura 22. El mejor diseño alcanzado por el enfoque CRO propuesto en el ChoppedPlastic UA-FLP.

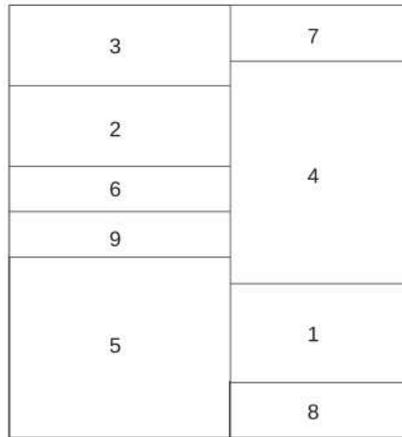


Figura 23. El mejor diseño alcanzado por el enfoque CRO propuesto en el O9 UA-FLP.

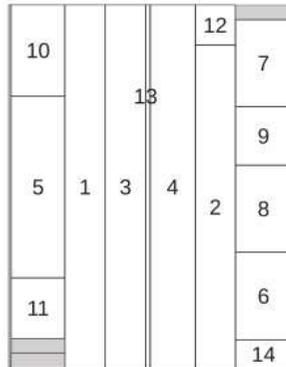


Figura 24. El mejor diseño alcanzado por el enfoque CRO propuesto en el Ba14 UA-FLP.

Considerando la estructura de bahías flexibles como estructura de representación, el algoritmo CRO es capaz de igualar o ganar a las propuestas previas en la mayoría de los casos. El algoritmo CRO ha igualado el mejor resultado para tres problemas y ha mejorado la mejor solución para otros 11 problemas UA-FLP de referencia (considerando un total de 17 UA-FLPs de prueba). Cabe destacar que se ha demostrado la eficacia y efectividad del algoritmo CRO cuando ha sido aplicado a problemas pequeños, (como *Slaughterhouse*, *CartonPacks*, *ChoppedPlastic*, O7, O8, O9, Vc10Ra, Vc10rs, Ba12, MB12, Ba14, Ma15), y resolviendo problemas de tamaño mediano, ya que la propuesta de CRO es capaz de alcanzar el mejor resultado para el problema AB20 y muy cercano al mejor en el caso del problema SC30. Cuando se

resuelven problemas grandes esta propuesta es capaz de alcanzar valores muy próximos a los mejores soluciones, éste es el caso del problema SC35 y DU62. Es en esta categoría de problemas UA-FLP cuando exclusivamente en tres problemas la propuesta CRO no es capaz de alcanzar la mejor solución pero si alcanza valores muy cercanos a los mejores obtenidos por otros trabajos previos.

Además, la Tabla 6 compara detalladamente los resultados alcanzados por la propuesta CRO y los mejores resultados obtenidos por otros autores. La Tabla 7 muestra la solución con el mejor valor de *fitness* producido por el algoritmo CRO, la diferencia (en porcentaje) entre la solución con el mejor *fitness* alcanzado por la propuesta CRO y el mejor resultado conocido considerando la estructura de bahías flexibles como estructura de representación alcanzado por anteriores trabajos, y finalmente, esta tabla presenta los mejores diseños de instalaciones obtenidos por el algoritmo CRO. Para completar esta tabla, se incluyen imágenes de ejemplos de las soluciones de la distribución en planta de los siguientes problemas de referencia UA-FLP: *Slaughterhouse*, *CartonPacks*, *ChoppedPlastic*, O9, Ba14, Ma15a, Ma15s. Todas estas soluciones mejoran las soluciones obtenidas previamente y se muestran en las Figuras desde la 20 a la 24.

Es bien conocido que una correcta distribución en planta puede incrementar la eficiencia y reducir el coste de la producción industrial en una forma muy considerable. En este sentido, los resultados que han sido obtenidos por esta propuesta mejoran de una manera significativa el rendimiento y la eficiencia en las plantas industriales.

6. Algoritmo de optimización de arrecifes de coral basado en un modelo de islas para el UA-FLP

6.1. *Introducción*

Hoy en día, el diseño industrial inteligente tiene como objetivo mejorar eficiencia y productividad de los recursos industriales, con la finalidad de aumentar la calidad del producto final (Abraham et al., 2020). La forma en la que se distribuyen los elementos que conforman una planta industrial debe ser considerada como un tema muy importante para tratar ya que, es bien sabido, que el buen diseño físico industrial de una planta puede afectar de una manera sustancial a los costes de producción y a la eficiencia de la industria (Singh y Sharma, 2006). Esta afirmación también fue formulada por Drira et al. (2017), quienes también identificaban y describían un gran número de diferentes problemas de distribución en planta. Considerando las diferentes taxonomías de los problemas de distribución en planta, el problema de distribución en planta de áreas desiguales (conocido en inglés como UA-FLP), propuesto por Armour y Buffa (1963) ha sido identificado como uno de los problemas más útiles y estudiados dentro de todas las clasificaciones de problemas de distribución en planta (Palomo-Romero et al. 2017). En el UA-FLP, los elementos que serán distribuidos son departamentos rectangulares, dentro de un área rectangular que es conocida. Además, el solapamiento de los elementos no está permitido. Normalmente, el objetivo principal que se tiene en cuenta es el de minimizar el flujo de material que existe entre los elementos del sistema productivo (Gonçalves y Resende, 2015). Para trabajar con el problema UA-FLP, han sido propuestos diferentes algoritmos de la literatura y también fuertes técnicas metaheurísticas en los mismos, como por ejemplo los que se detallan a continuación: (Garcia-Hernandez et al., 2013b; Gonçalves y Resende, 2015;

Wong y Komarudin, 2010; Garcia-Hernandez et al., 2019). Para codificar el problema del UF-FLP como una metaheurística, han sido usadas tres estructuras básicas de representación: Diseño de Bloque (BL), Estructura de Árbol de Cortes (STS) y Estructura de Bahías Flexibles (FBS). La última fue sugerida por (Tong 1991), y ha sido la más usada en la literatura ya que tiene una estructura muy simple que no añade una excesiva complejidad computacional a la propuesta, y además permite la posibilidad de incorporar pasillos en el diseño (Wong y Komarudin 2010).

Con respecto al buen rendimiento, los métodos metaheurísticos han mostrado alguna desventaja, tales como el estancamiento del óptimo local (Kurdi, 2016) y la gran cantidad de tiempo computacional para alcanzar soluciones satisfactorias (Sitkoff, 1995). En este sentido, la computación paralela surge para intentar solucionar esta problemática, ya que ofrece la capacidad de preservar la diversidad de la población gracias a una evolución simultánea de diferentes poblaciones al mismo tiempo (Whitley et al., 1998). Esto produce en general mejores soluciones que las metaheurísticas clásicas (Cantu-Paz et al., 1998), además de menos iteraciones del algoritmo (Alba y Troya, 1999).

El buen rendimiento obtenido por las metaheurísticas paralelas resolviendo duras tareas de optimización ha motivado la creación de una nueva propuesta para resolver el problema de UA-FLP. Específicamente, en este capítulo, se propone y evalúa dos nuevos modelos de islas basados en la optimización de arrecifes de coral (IMCRO), para resolver el problema de UA-FLP.

Estos algoritmos combinan la idea de muchas poblaciones independientes como movimientos migracionales (modelo de islas) con la exploración y explotación de las estrategias de optimización del algoritmo de coral, las cuales están inspiradas en la forma de reproducción y supervivencia de los corales en el mundo real (Salcedo-Sanz et al., 2014a; Salcedo-Sanz, 2017). Ambas estrategias han sido aplicadas separadamente para resolver el problema de UA-FLP con resultados positivos (Palomo-Romero et al., 2017; Garcia-Hernandez et al.,

2019). Desde nuestro conocimiento, no existe ninguna propuesta, que haya combinado ambos algoritmos (modelos de islas y optimización basados en arrecifes de coral) para crear una nueva técnica para resolver el problema de UA-FLP. Adicionalmente, se han descrito e implementado dos nuevas versiones de la nueva propuesta IMCRO.

6.2. Problema de UA-FLP. Definición y antecedentes

6.2.1 Definición del problema de UA-FLP

Armor y Buffa establecieron el UA-FLP en 1963 por primera vez. El problema se define mediante un diseño rectangular de dimensiones ($W \times H$) que son fijas. Además, hay un grupo de instalaciones o departamentos con un área determinada (A_i). La suma de las áreas del departamento debe ser menor o igual que el área total del diseño rectangular. El objetivo del problema es colocar todos los departamentos en el diseño optimizando un criterio dado y teniendo en cuenta que no se permite la superposición entre departamentos. La siguiente función define lo expuesto anteriormente:

$$\sum_i^n A_i \leq W \times H$$

En Aiello et al. (2012) se afirma que el UA-FLP implica como objetivo principal la minimización del flujo de material entre departamentos. La puntuación de aptitud utilizada en los algoritmos evolutivos para evaluar los problemas del caso de UA-FLP se basa, por lo tanto, en el coste de flujo de material. Además, para guiar el proceso de búsqueda a individuos factibles, ha sido utilizada una función de penalización propuesta por Tate y Smith (1995). De esta manera, para cada solución en el algoritmo, se define un punto de penalización, que es proporcional al número de instalaciones que conforman el diseño y que no satisfacen la restricción de la relación de aspecto (ya sea la relación de aspecto máxima o la longitud mínima del lado). Estas instalaciones se consideran como inviables. La

función *fitness* que minimiza el material de flujo es la siguiente:

$$OF_z = \sum_i^n \sum_j^n f_{ij} d_{ij} + (D_{inf})^k (V_{feas} - V_{all})$$

donde n es el número de departamentos en el diseño, f_{ij} es el flujo de material entre los departamentos i y j , d_{ij} es la distancia de Manhattan entre i y j , D_{inf} es la cantidad de instalaciones que no son factibles, V_{feas} es el mejor valor de aptitud física posible que se haya logrado hasta ahora, V_{all} es el mejor valor de condición física que se ha logrado hasta el momento y k es un parámetro de penalización que se ajusta al valor de la función de penalización (se ha incorporado con un valor de '3' como en Tate y Smith, 1995).

6.2.2 Algoritmos paralelos evolutivos

Dado el incremento de la complejidad de algoritmos UA-FLP, es necesaria una nueva forma de resolver estos problemas, ya que, la cantidad de tiempo y la capacidad computacional para alcanzar una solución aceptable, empieza a ser prohibitiva, incluso de la aplicación de búsquedas metaheurísticas. En este escenario, la paralelización parece una buena forma para resolver el problema de UA-FLP en problemas con un gran número de departamentos o instancias. De hecho, existen propuestas previas que usan estrategias paralelas para abordar este problema, específicamente, Palomo-Romero et al. (2017) usó un modelo de islas de algoritmos genéticos para resolver el UA-FLP. Esta estrategia permitía obtener buenos resultados en comparación con otras propuestas metaheurísticas no paralelas. De acuerdo con Cantu-Paz (1998), los algoritmos paralelos evolutivos pueden ser clasificados en tres grandes grupos, dependiendo del tratamiento que se le dé a la población total y las relaciones establecidas entre las unidades de computación:

- **Global single-population master-slave:** para una única población existen muchos procesos para computar el *fitness* de los individuos.

- **Single-population fine-grained:** la población total es dividida en regiones más pequeñas que sólo se relacionan con las poblaciones vecinas mediante cruces.
- **Multiple-population coarse-grained:** en este tipo de algoritmo hay muchas poblaciones que evolucionan de manera independiente una de las otras, y la única interacción entre ellas consiste en un intercambio de individuos (migración), realizado con una cierta frecuencia.

6.3. Una nueva metaheurística que combina el modelo de islas y CRO (IMCRO)

En esta sección se presenta el modelo IMCRO propuesto, detallando sus características, diferentes componentes y versiones del algoritmo. En primer lugar, se describe una breve introducción del algoritmo básico CRO. En la siguiente subsección se describe cómo puede construirse un modelo de islas para resolver el modelo de UA-FLP. Finalmente se incluye un modelo de islas extendido que incluye diferentes procedimientos de búsqueda en cada isla.

6.3.1 Algoritmo básico de arrecifes de coral CRO

El algoritmo CRO de Salcedo-Sanz et al. (2014b) es un algoritmo de tipo evolutivo, en donde se define una superficie de tamaño $M \times N$, en la cual, pueden establecerse diferentes corales X_k (soluciones potenciales al problema en cuestión). Inicialmente, el arrecife no está completamente ocupado, es decir, existen agujeros (zonas vacías) en el arrecife. Durante el proceso evolutivo, los corales se reproducen de dos maneras diferentes: sexual y asexual. Puede haber dos tipos de relaciones sexuales: reproducción externa (*Broadcast spawning*), donde dos corales intercambian información genética para crear un nuevo individuo (larva), o interna (*Brooding*), donde se crea la larva resultante con una pequeña alteración de un coral monoparental. A su vez, la reproducción asexual (o *Budding*) la realizan los mejores individuos del arrecife, que producen larvas

por medio de la autoduplicación.

Las soluciones producidas por el proceso de reproducción intentarán encontrar un lugar en el arrecife para instalarse y sobrevivir durante las siguientes generaciones. Una larva (nueva solución al problema) puede establecerse en el arrecife si el lugar que ha seleccionado para asentarse está vacío, o si el coral actual en el arrecife tiene peor salud (función *fitness*) que la larva que lo intenta. Si después de tres intentos una larva no ha encontrado un lugar para prosperar, se descarta. Al final de cada generación, algunos de los peores corales también pueden ser depredados, formando nuevos agujeros en el arrecife. La Figura 25 ilustra el proceso de evolución para una generación del arrecife.

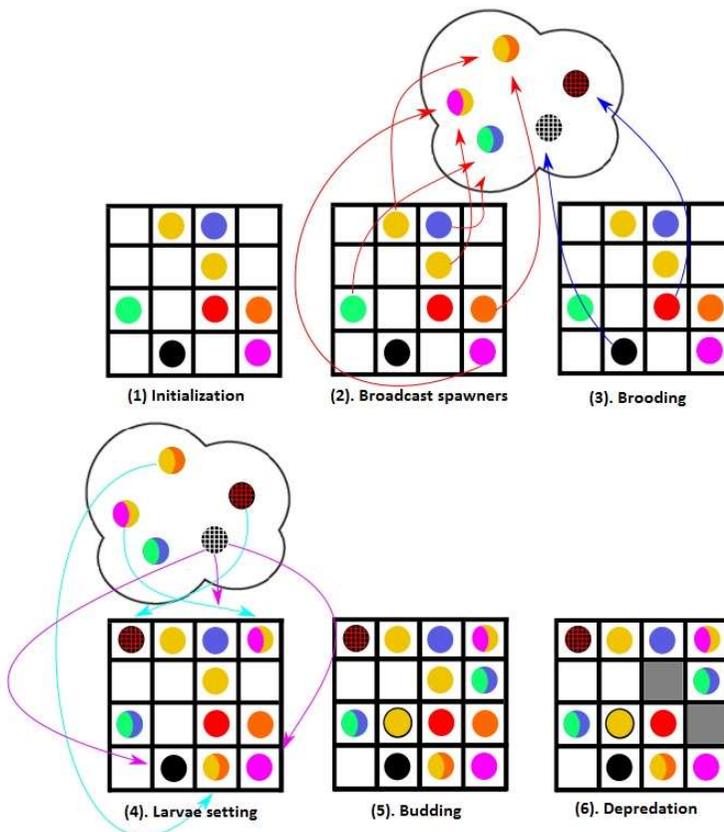


Figura 25. Proceso de evolución en algoritmo CRO.

6.3.2 Codificación de los corales para el UA-FLP

De acuerdo con Gómez et al. (2003), un cromosoma válido (coral en este caso) para representar una solución particular de UA-FLP con FBS consta de dos partes o fragmentos: el primero se refiere al orden en que las n instalaciones aparecen en el diseño y la segunda marca el final de cada bahía. El orden en que aparecen los departamentos en el primer fragmento del cromosoma corresponde a una lectura de los departamentos en el diseño, de arriba a abajo y de izquierda a derecha. El segundo fragmento es un vector codificado en binario donde el valor "1" en una posición i representa que la instalación ubicada en la posición i en el primer fragmento es la última de la bahía. De esta manera, el siguiente cromosoma se asocia con la Figura 26.

Estructura del cromosoma:

1. Fragmento 1: {I,C,J,G,L,F,K,E,D,H,B,A}

1. Fragmento 2: {0,1,0,0,1,0,1,0,1,0,1,1}

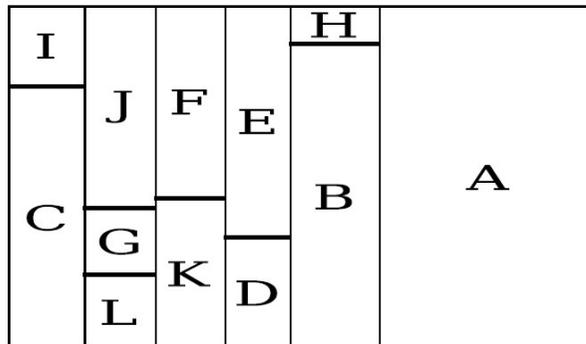


Figura 26. Ejemplo de estructura de bahías flexibles FBS.

6.3.3 El modelo de islas CRO (IMCRO)

El algoritmo IMCRO se puede describir como una versión con múltiples poblaciones del algoritmo básico de coral CRO. Donde $K = \{\Lambda_0, \Lambda_1, \dots, \Lambda_{r-1}\}$ es

una matriz de r arrecifes independientes de dimensiones $\{M_0 \times N_0, M_1 \times N_1, \dots, M_{r-1} \times N_{r-1}\}$. Cada arrecife Λ evoluciona como un CRO básico, y durante cada generación de frecuencia de migración (*migr-freq*) los mejores corales (*num_migr*) de cada arrecife se redistribuyen entre el resto de los arrecifes, lo que garantiza que ningún coral termine en su arrecife original. El esquema de migración seguido para la reorganización de los corales es algo parecido a la topología de malla de las redes: cada arrecife está "conectado" a todos los demás arrecifes en el sentido de que un coral X_k^r puede moverse a cualquier otro arrecife $s \neq r; s \in K$, pero los movimientos de los corales son independientes entre sí. La Figura 27 muestra un ejemplo de migración para un sistema de 5 arrecifes con 4 corales migratorios. Cada coral de un determinado arrecife puede (o no) tener diferentes destinos. Para explicar mejor la propuesta de IMCRO, a continuación se muestra el pseudocódigo del algoritmo, y se explica paso a paso:

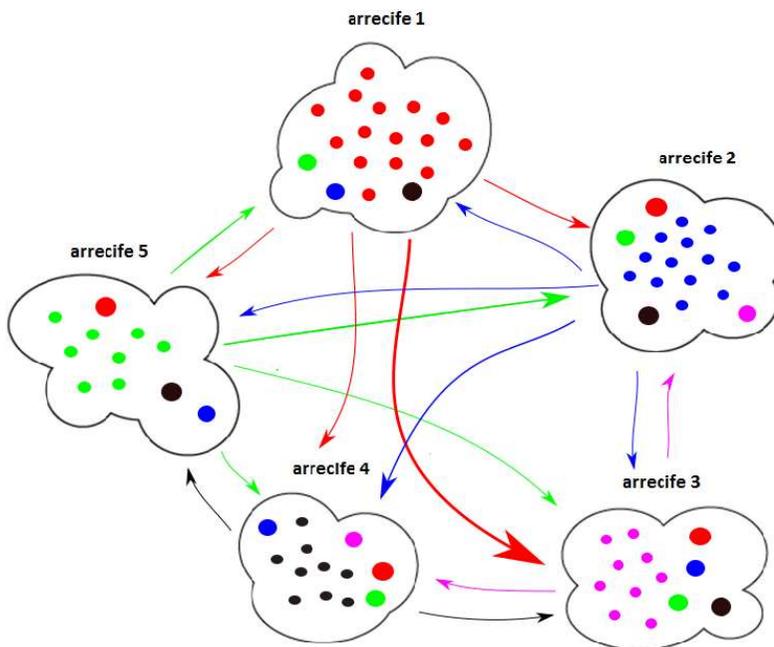


Figura 27. Proceso de evolución en algoritmo IMCRO.

IMCRO algorithm**Input** Algorithm's control parameters**Output** Feasible solution with best *fitness*

```
1: procedure IMCRO(num_reefs, m, n,  $\rho_0$ ,  $f_b$ ,  $f_a$ ,  $p_d$ , migr_freq, num_migr)
2:    $r \leftarrow 0$ 
3:   for num_reefs times do
4:     initialize reef  $r$  with size  $m_r \times n_r$  and occupation rate  $\rho'_0$ 
5:      $r \leftarrow r + 1$ 
6:   end for
7:   num_it  $\leftarrow 0$ 
8:   repeat
9:     for each reef  $r$  do
10:      reproduce corals fraction  $f_b^r$  by broadcast spawning
11:      reproduce corals fraction  $1 - f_b^r$  by brooding
12:      larvae evaluation
13:      larvae setting
14:      reproduce best corals fraction  $f_a^r$  by asexual reproduction
15:      depredation of  $f_d^r$  worst reef corals with  $p_d^r$  probability
16:    end for
17:    num_it  $\leftarrow$  num_it + 1
18:    if num_it = migr_freq then
19:      migrate top num_migr corals from each reef
20:      num_it  $\leftarrow 0$ 
21:    end if
22:  until stop condition
23:  return best feasible solution
24: end procedure
```

1. **Inicialización:** cada arrecife r se llena aleatoriamente con corales generados aleatoriamente hasta que se alcanza la fracción de ocupación ρ'_0 de la superficie total.

2. **Para cada arrecife:**

(a) *Broadcast spawning:* se selecciona una fracción X_k (soluciones potenciales al problema en cuestión. Un coral puede reproducirse una vez por generación). Cada pareja produce una larva por medio de operadores genéticos.

(b) *Brooding:* a partir de los corales restantes ($1 - f_b^r$) se generan varias larvas con una pequeña alteración aplicada a cada una de ellas.

(c) *Larvae setting:* todas las larvas creadas previamente intentan encontrar un lugar en el arrecife para asentarse siguiendo el siguiente procedimiento:

i) Selecciona un lugar al azar.

ii) Si el lugar está vacío: colocarlo o

iii) Si la larva tiene mejor salud que el coral que ocupa el lugar: sustituye el coral por la nueva larva creada.

iv. De lo contrario, vuelta al paso 2. (c).i.

Estos pasos se pueden repetir hasta tres veces. Después de eso, la larva se

descarta.

(d) *Budding*: una fracción f^r_a de los corales más saludables del arrecife se duplica, muta y trata de establecerse como en el paso anterior (*Larvae setting*).

(e) *Depredation*: La peor fracción de coral f^r_d puede ser depredada con una baja probabilidad de producción p^r_d .

3. **Migrar los mejores corales**: el proceso de migración se realiza con una frecuencia de (*migr_freq*) generaciones, donde el (*num_migr*) de los mejores corales se desplazan a otro arrecife.

4. Vuelta al paso 2 hasta que se cumpla una condición de detención. En la presente implementación, el algoritmo se detiene después de un número predeterminado de generaciones.

6.3.4 El modelo de islas CRO (IMCRO) extendido

Sin embargo, el enfoque descrito anteriormente puede producir un efecto no deseado: si cada arrecife explora el espacio de búsqueda de la misma manera (es decir, utilizando los mismos operadores de cruce y mutación), existe una alta probabilidad de que todos los arrecifes converjan al mismo espacio de búsqueda, en lugar de explotar diferentes zonas de ese espacio. Para evitar esto, una posible solución sería la implementación de diferentes estrategias de evolución por arrecife (combinación de operadores de cruce y de mutación). De esta manera, la implementación de los operadores de *Broadcast spawning* y *Brooding* depende del arrecife del que se recogen los padres. Esta idea se implementó previamente en una versión avanzada del algoritmo CRO, el llamado enfoque CRO-SL (Optimización de Arrecifes de Coral con Capas de Sustrato), que es un enfoque de métodos múltiples en el que diferentes operadores de búsqueda se aplican conjuntamente en una sola población (Salcedo-Sanz et al., 2016; Camacho-Gómez et al., 2018; Bermejo et al., 2018; Salcedo-Sanz, 2017). En este caso, el modelo de islas extendido propuesto trae esta idea y la extiende para paralelizar el algoritmo, al incluir cada operador de búsqueda en una isla diferente.

Los operadores implementados para este enfoque se enumeran y describen brevemente en las siguientes subsecciones.

Operadores de cruce

Esta sección describe los operadores de cruce considerados en el algoritmo IMCRO extendido, tanto para el orden de las instalaciones (primer fragmento de cromosoma) como para la estructura de la bahía (segundo fragmento).

- Operadores de cruce para el orden de instalaciones:

1) *Partially Mapped Crossover (PMX)* (Goldberg y Lingle, 1985). El cruce funciona siguiendo los siguientes pasos:

(a) Son seleccionados dos puntos de corte al azar.

(b) Es copiado el segmento definido por los puntos de corte de cada padre y se crea un mapa de equivalencia para la recombinación (el valor n -ésimo de un padre corresponde al valor n -ésimo del otro padre). Este mapa se utiliza para garantizar que ambos hijos representen permutaciones válidas.

(c) Los elementos restantes (no presentes en el segmento) son copiados del otro padre utilizando el mapa de equivalencia para aquellos elementos que ya están presentes porque se han copiado en el paso anterior.

2) *Order Crossover (OX)* (Eiben y Smith, 2003). La recombinación realizada por este operador es similar a PMX, pero no se utiliza un mapa de equivalencia. En cambio, los elementos fuera de los puntos de corte aparecen en el mismo orden que en el otro padre a partir del segundo punto de corte.

3) *Cycle* (Eiben y Smith, 2003). La forma de combinar padres es la siguiente:

(a) Se encuentran ciclos que alinean los cromosomas parentales y los valores de emparejamiento en una posición determinada.

(b) Es seleccionado uno de los ciclos encontrados (si los hay).

(c) Se crean hijos manteniendo los elementos del ciclo encontrado de uno de los

padres y llenando los vacíos restantes con los elementos del otro padre.

4) **Edge** (Eiben y Smith, 2003). Este operador se basa en la idea de que el hijo obtenido debe ser muy diferente de ambos padres. Para lograr esto, se construye una tabla frontera que almacena qué enlaces tiene cada elemento en ambos padres. Después de eso, se crea un hijo siguiendo este procedimiento:

(a) Se elige el elemento de la tabla al azar y se agrega a la descendencia.

(b) Se eliminan todas las referencias de la tabla.

(c) Se examina el contenido de la tabla para la entrada correspondiente y se selecciona, en orden de preferencia:

i) Elemento frontera común.

ii) Entrada de tabla con menos elementos. En caso de empate, se selecciona al azar de ese conjunto.

iii) En el caso de una lista vacía de límites frontera, la siguiente entrada que se examinará será la correspondiente al último elemento actual de la descendencia; de lo contrario, se aplica una selección aleatoria.

(d) Actualizar el valor de acuerdo con el valor seleccionado en el paso anterior.

(e) Regresar a (b) hasta que se complete la descendencia.

• Operadores de cruce para el fragmento de los cortes de bahía:

1) **One Point** (Holland, 1992). Se selecciona aleatoriamente un punto de corte, y ambos segmentos creados después se combinan alternativamente para crear dos hijos.

2) **N-point** (Eiben y Smith, 2003). Este operador es una generalización del anterior. El número de puntos de corte se elige al azar y los hijos crean segmentos alternos de ambos padres.

3) **Uniform** (Eiben y Smith, 2003). Cada alelo se selecciona aleatoriamente de uno u otro progenitor con la misma probabilidad.

Operadores de mutación

Los operadores de mutación considerados en el IMCRO extendido se presentan a continuación.

- Operadores de mutación para el fragmento del orden de instalaciones:
 - 1) *Towers* (**TWORS**) (Eiben y Smith, 2003). Se seleccionan dos posiciones al azar y se intercambia su contenido.
 - 2) *Inversion* (Eiben y Smith, 2003). Como en el operador anterior, se seleccionan dos posiciones al azar. Después de eso, la sección delimitada por ambas posiciones se invierte.
 - 3) *Scramble* (Eiben y Smith, 2003). Los elementos contenidos en una sección definida por dos posiciones aleatorias están codificados.
 - 4) *Insert* (Eiben y Smith, 2003). Se seleccionan dos elementos al azar. El segundo se posiciona después del primero, y todos los elementos después del primero se mueven para hacer espacio.
- Operadores de mutación para la estructura de la bahía:
 - 1) *Bit-swap* (Eiben y Smith, 2003). Para cada alelo, cada cierto tiempo calculado con una probabilidad, cambia su valor (de 0 a 1 o viceversa).

El algoritmo continúa igual que la versión básica explicada anteriormente pero utilizando diferentes operadores en las fases de *Brooding* y *Broadcast spawning*, dependiendo del arrecife (isla) en el que se realizan estas operaciones.

6.4. Experimentación y resultados

En esta sección se evaluará el rendimiento del enfoque IMCRO propuesto en las dos versiones sugeridas: IMCRO básico e IMCRO extendido. Para llevarlo a cabo, se ha seleccionado un conjunto común de problemas de UA-FLP para analizar los resultados alcanzados. Más adelante, se ha realizado una

comparación con enfoques alternativos y una discusión completa sobre los resultados obtenidos.

6.4.1 Descripción de los UA-FLPs de prueba

Se ha elegido un gran conjunto de UA-FLPs teniendo en cuenta diferentes tamaños del problema (pequeño, mediano y grande). La descripción relacionada con cada UA-FLP se expresa en la Tabla 8. Se detalla la siguiente información: nombre del problema, número de instalaciones, dimensiones de la planta (en metros cuadrados), restricción de factor de forma (mediante una relación de aspecto o una medida de lado mínimo), distancia medida y referencia del problema.

<i>Problem name</i>	<i>Fac.</i>	<i>W × H (m²)</i>	<i>Shape constr.</i>	<i>Dist.</i>	<i>Reference</i>
Slaughterhouse12	12	51.14 × 30.00	$\alpha = 4$		Eucl.Salas-Morera et al. (1996)
CartonPacks	11	20.00 × 14.50	$\alpha = 4$		Eucl.Garcia-Hernandez et al. (2013a)
ChoppedPlastic10	10	10.00 × 30.00	$\alpha = 4$		Eucl.Garcia-Hernandez et al. (2013a)
O7	7	8.54 × 13.00	$\alpha = 4$		Rect.Meller et al. (1998)
O8	8	11.3 × 13.00	$\alpha = 4$		Rect.Meller et al. (1998)
O9	9	12.00 × 13.00	$\alpha = 5$		Rect.Wong and Komarudin (2010)
vC10Ra	10	25.00 × 51.00	$\alpha = 5$		Rect.van Camp et al. (1992)
Vc10Rs	10	25.00 × 51.00	$side = 5$		Rect.van Camp et al. (1992)
vC10Ea	10	25.00 × 51.00	$\alpha = 5$		Eucl.van Camp et al. (1992)
Vc10Es	10	25.00 × 51.00	$side = 5$		Eucl.van Camp et al. (1992)
Ba12	12	6.00 × 10.00	$side = 1$		Rect.Bazaraa (1975)
MB12	12	6.00 × 8.00	$\alpha = 4$		Rect.Bozer and Meller (1997)
Ba14	14	7.00 × 9.00	$side = \{1, 0\}$		Rect.Komarudin and Wong (2010)
AB20_ar3	20	2.00 × 3.00	$\alpha = 3$		Rect.Armour and Buffa (1963)
AB20_ar5	20	2.00 × 3.00	$\alpha = 5$		Rect.Armour and Buffa (1963)
AB20_ar7	20	2.00 × 3.00	$\alpha = 7$		Rect.Armour and Buffa (1963)
AB20_ar10	20	2.00 × 3.00	$\alpha = 10$		Rect.Armour and Buffa (1963)
AB20_ar15	20	2.00 × 3.00	$\alpha = 15$		Rect.Armour and Buffa (1963)
AB20_ar50	20	2.00 × 3.00	$\alpha = 50$		Rect.Armour and Buffa (1963)
SC30	30	12.00 × 15.00	$\alpha = 5$		Rect.Liu and Meller (2007)
SC35	35	16.00 × 15.00	$\alpha = 4$		Rect.Liu and Meller (2007)
Du62	62	Arbitrary	$\alpha = 4$		Rect.Dunker et al. (2003)

Tabla 8. Características de los UA-FLPs testados.

6.4.2 Metodología experimental

Para probar el enfoque IMCRO propuesto considerando las dos versiones propuestas (básica y extendida), se realizó una prueba empírica completa para encontrar los parámetros que mejor se ajustan al resolver el UA-FLP, es decir,

que combinando los parámetros produce las mejores soluciones para un determinado problema. La Tabla 9 muestra los posibles valores que se han considerado para cada parámetro del algoritmo. No todos los valores posibles se han probado, por supuesto, debido al número inalcanzable de combinaciones posibles, sino un conjunto de los más recomendados. Por ejemplo, la fracción de depredación y la probabilidad de depredación no deben tomar valores altos, ya que eso conduciría a un comportamiento excesivamente elitista, por lo que no se han considerado valores superiores a 0.1. El ajuste de parámetros se realizó seleccionando uno de los tres valores posibles considerados para cada parámetro y usando esa configuración para un problema representativo de cierto tamaño (O9 para las instancias más pequeñas, BA14 para las instalaciones de tamaño mediano y Du62 para las instancias grandes). Para problemas pequeños, el número de generaciones realizadas por el algoritmo ha sido 1000, mientras que para instancias medianas y grandes este parámetro se ha establecido en 5000. El algoritmo también puede detenerse antes de alcanzar ese número de generaciones si después de 500 iteraciones no ha encontrado una solución mejor. El número de generaciones se ha establecido muy alto porque durante la búsqueda no queremos dejar áreas buenas potenciales del espacio de búsqueda sin explorar, pero la segunda condición de detención asegurará que si el algoritmo converge, no se desperdicie tiempo de cálculo. Cada una de estas combinaciones de parámetros ha sido clonada 5 veces como en la propuesta de Komarudin y Wong (2010). Los parámetros que mejor se adaptan a cada categoría de UA-FLP se muestran en la Tabla 10, donde S representa problemas pequeños (hasta 12 instalaciones), M para medianos (13-25) y L para grandes (26 o más). En el algoritmo IMCRO básico, hemos utilizado PMX, *crossover* de un punto, mutación TWORS e intercambio de bits como operadores de búsqueda. A su vez, para el IMCRO extendido hemos definido cinco combinaciones de operadores de cruce y mutación:

- (a) **Cruce:** *PMX, 1-point* | **Mutación:** *TWORS, Bit-swap.*
- (b) **Cruce:** *Cycle, N-point* | **Mutación:** *Inversion, Bit-swap.*
- (c) **Cruce:** *OX, Uniform* | **Mutación:** *Scramble, Bit-swap.*
- (d) **Cruce:** *PMX, Uniform* | **Mutación:** *Insert, Bit-swap.*
- (e) **Cruce:** *Edge, N-point* | **Mutación:** *Scramble, Bit-swap.*

La propuesta de IMCRO se ha desarrollado utilizando Python 3. Como se ha llevado a cabo una comparación de tiempo con los enfoques básicos de CRO anteriores, es necesario especificar que la experimentación de prueba se ha ejecutado utilizando un Intel Core i5 6200U (2.30 GHz), 8 GB de RAM y un sistema operativo Linux.

Parameter	$N \times M$	r	ρ_0	F_b	F_a	F_d	P_d	$migr_freq$	num_migr
	10×10	5	0.6	0.7	0.1	0.01	0.01	5	5
Tested values	15×15	10	0.7	0.8	0.15	0.05	0.05	10	10
	25×25	20	0.8	0.9	0.2	0.1	0.1	20	20

Tabla 9. Valores usados para el ajuste de los parámetros.

Parameter	$N \times M$	r	ρ_0	F_b	F_a	F_d	P_d	$migr_freq$	num_migr
Selected values (S)	10×10	5	0.7	0.8	0.1	0.1	0.1	5	5
Selected values (M)	15×15	10	0.7	0.8	0.2	0.1	0.1	5	10
Selected values (L)	25×25	10	0.7	0.8	0.2	0.1	0.1	5	10

Tabla 10. Valores usados dependiendo del tamaño del problema.

6.4.3 Resultados

Esta subsección presenta los resultados obtenidos por ambas versiones (IMCRO básico e IMCRO extendido) del enfoque IMCRO propuesto. Inicialmente, un resumen de los mejores resultados (*OF Best*) y su media asociada (*OF mean*) alcanzada por las dos versiones de la propuesta IMCRO para cada uno de los UA-FLPs probados se enumeran en la Tabla 11. Además, esta tabla muestra la diferencia (en porcentaje) entre las mejores soluciones obtenidas por las dos versiones de IMCRO ($100 \times$ (la solución más conocida encontrada en la literatura

- la mejor solución IMCRO) / (la mejor solución IMCRO)). Además, esta tabla se completa ofreciendo el tiempo empleado por la propuesta IMCRO para alcanzar el mejor resultado en cada versión implementada, y también, el tiempo de CPU necesario en la propuesta CRO básica (García-Hernández et al., 2019). Mirando la Tabla 11, es posible confirmar que las dos versiones desarrolladas de los enfoques IMCRO son robustas, ya que la diferencia entre la aptitud de la mejor solución y la aptitud promedio es muy pequeña. Esta diferencia aumenta a medida que aumenta el número de departamentos en el problema. Con respecto a las unidades de tiempo necesarias para lograr las mejores soluciones para cada problema, el enfoque IMCRO es más eficiente en términos de tiempo de CPU, ya que supera el enfoque CRO básico en todos los casos probados. Sin embargo, el tiempo de CPU no se considera algo crucial en el UA-FLP (See y Wong, 2008). Finalmente, la Tabla 14 y Tabla 15 muestran el rendimiento de las versiones básica y extendida del enfoque IMCRO al resolver el UA-FLP. A partir de este análisis, podemos afirmar que la versión extendida alcanza mejores resultados que la básica, ya que es capaz de igualar los resultados obtenidos por su contraparte básica en 18 de todas las instancias probadas y superarlas en 4 de ellas (*Slaughterhouse*, *CartonPacks*, *AB20_AR5* y *AB20_AR7*). Las soluciones logradas para los problemas anteriores por el IMCRO extendido se ilustran en la Figura 28.

UA-FLP	B_OF Best	B_OF Mean	E_OF Best	E_OF Mean	Dif(%)	B_CPU(s)	E_CPU(s)	CRO CPU(s)
Slaughterhouse	3440.49	3490.49	3439.96	3487.11	0.02	72.00	74.00	75.00
Carton Packs	86.08	86.08	85.99	86.03	0.1	69.00	73.00	74.00
Chopped Plastic	257.94	257.94	257.94	257.94	0.00	64.00	63.00	65.00
O7	134.16	134.16	134.16	134.16	0.00	4.00	4.00	4.00
O8	245.48	245.48	245.48	245.48	0.00	22.00	22.00	24.00
O9	238.73	238.73	238.73	238.73	0.00	35.00	39.00	39.00
vC10Ra	20142.13	20142.13	20142.13	20142.13	0.00	60.00	59.00	61.00
Vc10Rs	22897.65	22897.65	22897.65	22897.65	0.00	61.00	60.00	63.00
vC10Ea	18554.59	18554.59	18554.59	18554.59	0.00	86.00	84.00	-
Vc10Es	18823.74	18823.74	18823.74	18823.74	0.00	82.00	81.00	-
Ba12	8021.0	8021.96	8021.0	8099.96	0.00	86.00	85.00	87.00
MB12	125.00	125.00	125.00	125.00	0.00	77.00	80.00	81.00
Ba14	4649.22	4740.23	4649.22	4739.15	0.00	89.00	90.00	92.00
AB20_ar3	5396.37	5455.11	5396.37	5475.77	0.00	131.00	137	-
AB20_ar5	5252.98	5371.04	5252.98	5360.55	0.00	182.00	198	202
AB20_ar7	4789.24	4881.19	4785.96	4802.24	0.07	160.00	145	-
AB20_ar10	4367.56	4425.32	4367.56	4403.21	0.00	147.00	160	-
AB20_ar15	4099.63	4202.67	4099.38	4152.33	0.01	143.00	137	-
AB20_ar50	2382.73	2419.90	2382.73	2423.84	0.00	136.00	135	-
SC30	3714.74	4002.54	3714.74	3999.13	0.00	617.00	611.00	622.00
SC35	4263.3	4899.35	4263.3	4714.14	0.00	545.00	550.00	552.00
Du62	3654022.27	3674086.53	3646062.45	3667124.28	0.22	861.00	860	871.00

Tabla 11. Resultados obtenidos por las versiones IMCRO básico y extendido.

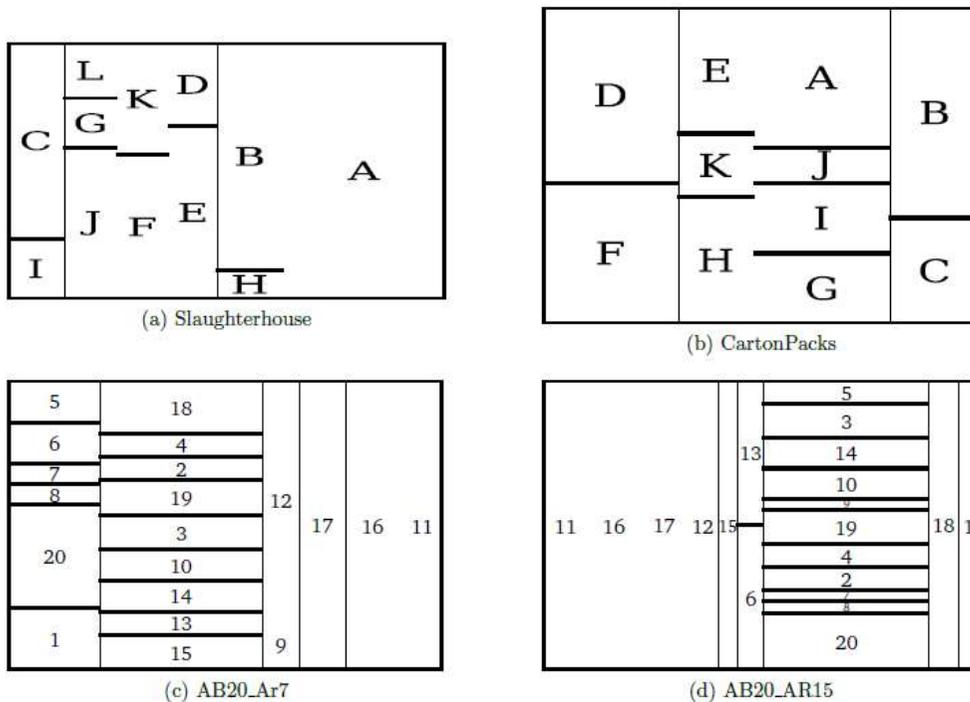


Figura 28. Las mejores soluciones obtenidas por el IMCRO extendido para algunas de las instancias probadas de UA-FLP.

Para mostrar los resultados alcanzados por nuestra propuesta se han usado las Tablas 12 y 14 para la versión básica de IMCRO, y las Tablas 13 y 15 para la versión de IMCRO extendida. En ellos, se muestra una comparación con las otras propuestas que también han utilizado FBS como estructuras de representación de la planta. En particular, se muestran los resultados obtenidos por los siguientes trabajos: García-Hernández et al. (2019), Palomo-Romero et al. (2017), Kulturel-Konak y Konak (2011), Kulturel-Konak (2012), Wong y Komarudin (2010) y Enea et al. (2005). Para cada UA-FLP, el mejor valor de la solución se ha resaltado en negrita. Echando un vistazo a la Tabla 12 y la Tabla 13, se puede ver que nuestra propuesta es capaz de alcanzar la mejor solución en la mayoría de los casos probados. En particular, las dos versiones de esta nueva propuesta son capaces de encontrar el mejor resultado en 19 UA-FLPs de los 22 casos probados. En los otros tres UA-FLPs, las dos versiones del sistema IMCRO pueden obtener resultados muy cercanos a las mejores soluciones obtenidas

mediante enfoques alternativos. Las versiones básica y extendida del enfoque IMCRO coinciden con las mejores soluciones previas en 10 de los problemas probados. Estos son los casos de los problemas *Slaughterhouse*, *CartonPacks*, *ChoppedPlastic*, O9, Ba14, AB20_AR3, AB20_AR5, AB20_AR7, AB20_AR15 y AB20_AR50. Para una mejor ilustración de los diseños de sus instalaciones asociadas, hemos incluido la representación del diseño físico de todos estos UA-FLPs donde el enfoque IMCRO supera las soluciones más conocidas (Figuras 28, 29 y 30). En todas las soluciones mostradas, el número de departamentos inviiables es cero, por lo que el valor de OF es el mismo que el valor del flujo de material, que es el mínimo encontrado para cumplir las restricciones de relación de aspecto. Además, las versiones básica y extendida de IMCRO pueden alcanzar las soluciones más conocidas en los siguientes UA-FLPs: O7, O8, Vc10Ra, Vc10Rs, Vc10Ea, Vc10Es, Ba12, MB12 y AB20_AR10.

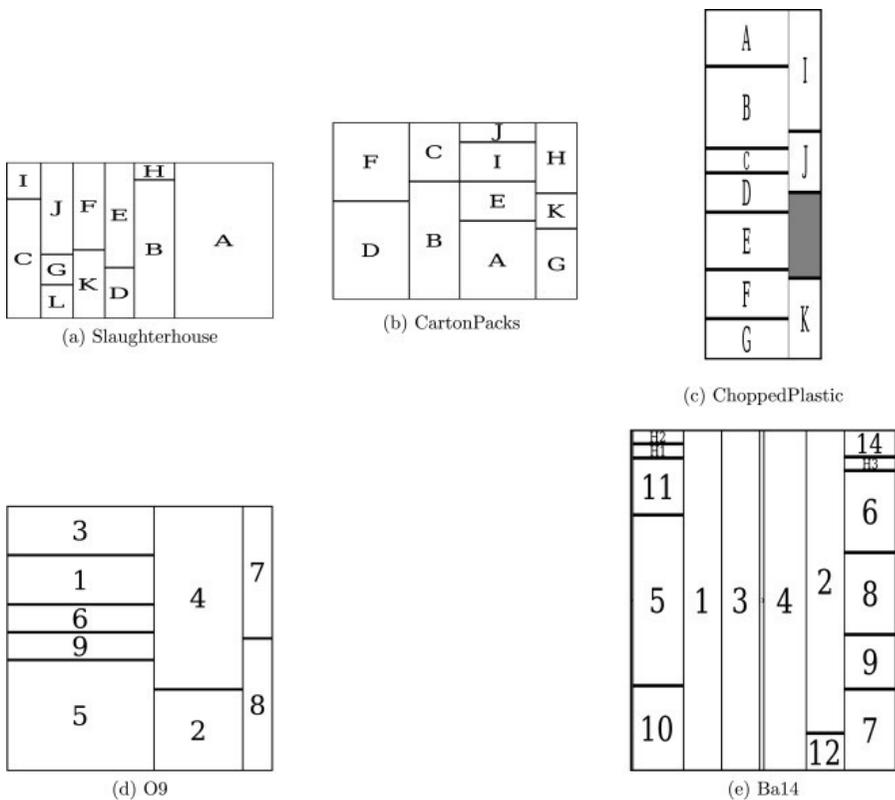


Figura 29. Las mejores soluciones obtenidas por el IMCRO básico para algunas de las instancias probadas de UA-

La Tabla 14 y la Tabla 15 agregan información relevante adicional al comparar las soluciones logradas por la versión básica y extendida del IMCRO y las mejores soluciones conocidas alcanzadas por enfoques previos, respectivamente. Por lo tanto, para cada UA-FLP probado, se muestra el mejor resultado logrado para cada versión del IMCRO propuesto, y también su estructura de diseño asociada y el porcentaje de diferencia entre el mejor valor de resultado obtenido por nuestro enfoque y la mejor solución encontrada en trabajos anteriores de la literatura relacionada.

<i>FBS approaches</i>							
<i>UA-FLP</i>	<i>E IMCRO</i>	García-Hernandez et al. (2019)	Palomo-Romero et al. (2017)	Kulturel-Konak and Konak (2011)	Kulturel-Konak (2012)	Wong and Komarudin (2010)	Enea et al. (2005)
Slaughterhouse	3440.49	3487.12	-	-	-	-	3854.00
CartonPacks	86.08	89.02	-	-	-	-	94.10
ChoppedPlastic	257.94	265.77	-	-	-	-	377.18
O7	134.16	134.16	134.19	-	-	-	-
O8	245.48	245.48	245.51	-	-	-	-
O9	238.73	239.44	241.06	-	-	241.06	-
Vc10Ra	20142.13	20142.13	20142.13	20142.13	21463.07	21463.1	-
Vc10Rs	22899.65	22899.65	22899.65	22899.65	22899.65	22899.65	-
Vc10Ea	18554.59	-	18554.59	-	-	-	-
Vc10Es	18823.74	-	18823.74	-	20320.52	-	-
Ba12	8021.00	8021.00	8435.83	8129.00	8021.0	8786.00	-
MB12	125.00	125.00	125.00	-	-	-	-
Ba14	4649.22	4665.93	4665.93	4780.91	4739.74	5004.55	-
AB20_ar3	5396.37	-	5419.49	5594.30	5594.29	-	-
AB20_ar5	5252.98	5256.95	5256.10	5336.36	5297.6	5677.83	-
AB20_ar7	4789.24	-	4844.49	4793.5	4793.47	-	-
AB20_ar10	4367.56	-	4367.56	4440.70	4440.68	-	-
AB20_ar15	4099.63	-	4100.17	4119.8	4119.8	-	-
AB20_ar50	2382.73	-	2382.74	3526.5	2457.79	-	-
SC30	3714.74	3519.44	3613.11	3443.34	3563.95	-	-
SC35	4272.21	4263.30	3885.29	3700.75	-	-	-
Du62	3654022.27	3713876.55	-	3641497.00	-	-	-

Tabla 12. Resumen de las soluciones obtenidas por el IMCRO básico vs otros resultados en la literatura.

<i>FBS approaches</i>							
<i>UA-FLP</i>	<i>E IMCRO</i>	García-Hernandez et al. (2019)	Palomo-Romero et al. (2017)	Kulturel-Konak and Konak (2011)	Kulturel-Konak (2012)	Wong and Komarudin (2010)	Enea et al. (2005)
Slaughterhouse	3439.96	3487.12	-	-	-	-	3854.00
CartonPacks	85.99	89.02	-	-	-	-	94.10
ChoppedPlastic	257.94	265.77	-	-	-	-	377.18
O7	134.16	134.16	134.19	-	-	-	-
O8	245.48	245.48	245.51	-	-	-	-
O9	238.73	239.44	241.06	-	-	241.06	-
Vc10Ra	20142.13	20142.13	20142.13	20142.13	21463.07	21463.1	-
Vc10Rs	22899.65	22899.65	22899.65	22899.65	22899.65	22899.65	-
Vc10Ea	18554.59	-	18554.59	-	-	-	-
Vc10Es	18823.74	-	18823.74	-	20320.52	-	-
Ba12	8021.00	8021.00	8435.83	8129.00	8021.0	8786.00	-
MB12	125.00	125.00	125.00	-	-	-	-
Ba14	4649.22	4665.93	4665.93	4780.91	4739.74	5004.55	-
AB20_ar3	5396.37	-	5419.49	5594.30	5594.29	-	-
AB20_ar5	5252.98	5256.95	5256.10	5336.36	5297.6	5677.83	-
AB20_ar7	4785.96	-	4844.49	4793.5	4793.47	-	-
AB20_ar10	4367.56	-	4367.56	4440.70	4440.68	-	-
AB20_ar15	4099.38	-	4100.17	4119.8	4119.8	-	-
AB20_ar50	2382.73	-	2382.74	3526.5	2457.79	-	-
SC30	3714.74	3519.44	3613.11	3443.34	3563.95	-	-
SC35	4272.21	4263.3	3885.29	3700.75	-	-	-
Du62	3646062.45	3713876.55	-	3641497.00	-	-	-

Tabla 13. Resumen de las soluciones obtenidas por el IMCRO extendido vs otros resultados en la literatura.

UA-FLP	B_IMCRO	Diff.(%)	B_IMCRO chromosome
Slaughterhouse	3440.49	1.35	9, 3 10, 7, 12 6, 11 5, 4, 13 8, 2 1
CartonPacks	86.08	3.41	6, 4 3, 2 10, 9, 5, 1 8, 11, 7
ChoppedPlastic	257.94	3.03	1, 2, 3, 4, 5, 6, 7 8, 9, 10
O7	134.16	0.00	3, 5, 7 1, 4, 6, 2
O8	245.48	0.00	5, 8, 6, 3 2, 1, 4, 7
O9	238.73	0.29	3, 1, 6, 9, 5 4, 2 7, 8
Vc10Ra	20142.13	0.00	5, 8, 10, 9, 2, 6, 1 4, 7, 3
Vc10Rs	22899.65	0.00	3, 5 9, 10, 8 2, 4 6, 7 1
Vc10Ea	18554.59	0.00	7, 4, 3 1 8, 5, 9, 2, 6, 10
Vc10Es	18823.74	0.00	10, 9 8, 4, 2 5, 7, 6 3, 1
Ba12	8021.0	0.00	4, 10 9, 5, 7 3 2, 12 1 11, 8, 6
MB12	125.00	0.00	12 9, 1, 5, 6, 8, 2, 4, 3, 7, 10 11
Ba14	4649.22	0.35	11, 5, 10 1 3 13 4 2, 12 14, 6, 8, 9, 7
AB20_ar3	5396.37	0.42	18, 20 6, 4, 2, 7, 8, 5 1, 19, 3, 10 12, 9, 14 17, 13, 15 16, 11
AB20_ar5	5252.98	0.05	11, 16 15, 17 13, 12 5, 14, 10, 9, 19, 3 20, 8, 7, 2, 4, 6 1, 18
AB20_ar7	4789.24	0.08	1, 20, 8, 7, 6, 5 13, 15, 14, 10, 3, 19, 2, 4, 18 9, 12 17 16 11
AB20_ar10	4367.56	0.00	5, 20, 8, 7, 2, 4, 6, 18 15, 13, 14, 10, 9, 19, 3, 1 12 17 16 11
AB20_ar15	4099.38	0.02	11 11 16 17 12 15 13, 6 5, 3, 14, 10, 9, 19, 4, 2, 7, 8, 20 18 1
AB20_ar50	2382.73	0.0004	11 16 13 17 12 15 9 14 10 3 19 4 2 6 7 8 20 5 18 1
SC30	3714.74	-7.3	20, 30, 19, 26 29, 27 25, 3, 1 4, 24, 17 18, 28, 23 15, 11, 9, 8 12, 7 16, 14, 13, 10, 5, 6, 2, 21, 22
SC35	4272.21	-13.37	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 20, 34, 22, 4, 26 3
Du62	3654022.27	-0.34	45, 26, 6 39, 35, 22, 4, 41 33, 30, 60, 50, 36, 23, 13 19, 51, 8, 18, 32, 20, 25, 53

Tabla 14. Resumen de resultados y cromosomas obtenidos por el IMCRO básico.

UA-FLP	E_IMCRO	Diff.(%)	E_IMCRO chromosome
Slaughterhouse	3439.96	1.37	3, 9 12, 7, 10 11, 6, 13 4, 5 2, 8 1
CartonPacks	85.99	3.52	4, 6 5, 11, 8 1, 10, 9, 7 2, 3
ChoppedPlastic	257.94	3.03	1, 2, 3, 4, 5, 6, 7 8, 9, 10
O7	134.16	0.00	3, 5, 7 1, 4, 6, 2
O8	245.48	0.00	5, 8, 6, 3 2, 1, 4, 7
O9	238.73	0.29	3, 1, 6, 9, 5 4, 2 7, 8
Vc10Ra	20142.13	0.00	5, 8, 10, 9, 2, 6, 1 4, 7, 3
Vc10Rs	22899.65	0.00	3, 5 9, 10, 8 2, 4 6, 7 1
Vc10Ea	18554.59	0.00	7, 4, 3 1 8, 5, 9, 2, 6, 10
Vc10Es	18823.74	0.00	10, 9 8, 4, 2 5, 7, 6 3, 1
Ba12	8021.0	0.00	4, 10 9, 5, 7 3 2, 12 1 11, 8, 6
MB12	125.00	0.00	12 9, 1, 5, 6, 8, 2, 4, 3, 7, 10 11
Ba14	4649.22	0.35	11, 5, 10 1 3 13 4 2, 12 14, 6, 8, 9, 7
AB20_ar3	5396.37	0.42	18, 20 6, 4, 2, 7, 8, 5 1, 19, 3, 10 12, 9, 14 17, 13, 15 16, 11
AB20_ar5	5252.98	0.05	11, 16 15, 17 13, 12 5, 14, 10, 9, 19, 3 20, 8, 7, 2, 4, 6 1, 18
AB20_ar7	4785.96	0.15	5, 6, 7, 8, 20, 1 18, 4, 2, 19, 3, 10, 14, 13, 15 12, 9 17 16 11
AB20_ar10	4367.56	0.00	5, 20, 8, 7, 2, 4, 6, 18 15, 13, 14, 10, 9, 19, 3, 1 12 17 16 11
AB20_ar15	4099.63	0.01	11 16 17 12 15 13, 6 5, 3, 14, 10, 9, 19, 4, 2, 7, 8, 20 1 18
AB20_ar50	2382.73	0.0004	11 16 13 17 12 15 9 14 10 3 19 4 2 6 7 8 20 5 18 1
SC30	3714.74	-7.3	20, 30, 19, 26 29, 27 25, 3, 1 4, 24, 17 18, 28, 23 15, 11, 9, 8 12, 7 16, 14, 13, 10, 5, 6, 2, 21, 22
SC35	4272.21	-13.37	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 20, 34, 22, 4, 26 3
Du62	3646062.45	-0.12	45, 39, 6 26, 35, 22, 4, 41 33, 30, 60, 36, 50, 23, 13 19, 51, 8, 18, 32, 20, 25, 53
			34, 55, 5, 24, 56, 21, 16, 3, 43, 29, 49, 7 31, 47, 12, 57, 38, 61, 10
			9, 52, 28, 59, 11, 1, 44, 14 15, 37, 27, 2, 40, 42, 48 46, 58, 54 17, 62

Tabla 15. Resumen de resultados y cromosomas obtenidos por el IMCRO extendido.

Después de analizar la información mostrada anteriormente, podemos decir que el algoritmo IMCRO propuesto es efectivo al abordar el UA-FLP de diferentes categorías de acuerdo con su tamaño. Por lo tanto, el algoritmo IMCRO puede ganar o alcanzar las mejores soluciones conocidas en todos los UA-FLPs pequeños. Estos son los casos de *Slaughterhouse*, *CartonPacks*, *ChoppedPlastic*, *O7*, *O8*, *O9*, *Vc10Ra*, *Vc10Rs*, *Vc10Ea*, *Vc10Es*, *Ba12*, *MB12*, *Ba14*. Al

resolver problemas medios como AB20_AR3, AB20_AR5, AB20_AR7, AB20_AR10, AB20_AR15 y AB20_AR50, el IMCRO propuesto también demuestra su efectividad para lograr las mejores soluciones. En esta categoría de tamaño, refiriéndose a SC30, el enfoque propuesto llega a soluciones muy cercanas a los resultados más conocidos. Esto también ocurre en los UA-FLPs grandes como son SC35 y DU62, donde, nuevamente, IMCRO puede aproximarse a las soluciones más conocidas para estos casos.

7. Conclusiones finales

En esta tesis, se ha presentado un estado del arte exhaustivo y reciente, relacionado con el problema de distribución en planta, en general, y en el problema de distribución en planta de áreas desiguales, en particular. El estudio y análisis realizado se basa en una gran cantidad de referencias bibliográficas. A partir de este análisis, podemos confirmar que los artículos relacionados con el problema de distribución en planta siguen publicándose de forma regular en las revistas de investigación de prestigio internacionales y altamente posicionadas en los mejores rankings de calidad (*Journal Citation Reports*, por ejemplo). Por lo tanto, puede decirse que el problema de distribución en planta es un área de investigación que sigue estando activa y viva.

En el estado del arte realizado, se ha estudiado el problema de distribución en planta en detalle, aportado las distintas estructuras de representación usadas y los diferentes enfoques considerados para abordar este interesante problema. En este sentido, en términos de métodos utilizados para resolver los problemas de distribución en planta, obviamente se puede ver que el uso de las metaheurísticas es el enfoque más usado en las investigaciones, para hacer frente a problemas de mayor tamaño y tener en cuenta restricciones más realistas, siendo los algoritmos evolutivos los enfoques más populares. Los métodos de resolución también se combinan formando algoritmos híbridos, ya sea para resolver problemas complejos o para proporcionar soluciones más realistas.

Por un lado, en esta tesis doctoral, se ha realizado una evaluación en profundidad del rendimiento de aplicar el algoritmo de arrecifes de coral para optimizar los problemas de distribución en planta de áreas desiguales considerando como estructura de representación la estructura de bahías flexibles. Tras un análisis exhaustivo de la bibliografía, parece que es la primera propuesta que usa un algoritmo de optimización de arrecifes de coral para resolver el

problema de distribución en planta de áreas desiguales. Esta propuesta ha sido probada con 17 instancias de problemas de distribución en planta de áreas desiguales tomados de las referencias, y el rendimiento ha sido analizado y comparado con las diferentes propuestas del estado del arte publicados previamente. Del estudio empírico realizado, puede decirse que la propuesta de optimización basada en arrecifes de coral es capaz de mejorar o alcanzar la mejor solución en 14 de los 17 problemas probados si consideramos la estructura de representación de bahías flexibles. Además cuando consideramos esta propuesta frente a la estructura de representación de árboles de cortes, podemos decir que el algoritmo de optimización basado en arrecifes de coral mejora o alcanza la mejor solución en 7 de los 17 casos probados. En los casos restantes, esta propuesta es capaz de alcanzar soluciones con resultados muy cercanos a los mejores conocidos. Este hecho muestra que el algoritmo de optimización basado en arrecifes de coral tiene un rendimiento excelente cuando se aplica al problema de distribución en planta de áreas desiguales.

Por otro lado, en esta tesis doctoral, se ha propuesto un nuevo enfoque que combina un modelo de islas y el algoritmo de optimización de arrecifes de coral. Este novedoso sistema de modelo de islas basado en la optimización de arrecifes de coral se ha aplicado al problema de distribución en planta de áreas desiguales considerando la estructura de bahías flexibles como estructura de representación de la planta industrial. Se han propuesto y probado dos versiones diferentes del algoritmo de modelo de islas basado en la optimización de arrecifes de coral: las versiones del modelo de islas basado en la optimización de arrecifes de coral básica y la del modelo de islas basado en la optimización de arrecifes de coral extendida. Para ilustrar la efectividad de ambas versiones, se han probado 22 instancias conocidas del problema de distribución en planta de áreas desiguales extraídas de la literatura, y se ha llevado a cabo una comparación completa con los trabajos de investigación previos. A partir del estudio empírico y el análisis realizado, es evidente que el modelo de islas basado en la optimización de arrecifes de coral propuesto es un algoritmo robusto, capaz de alcanzar las

mejores soluciones de diseño en la mayoría de los problemas probados. Específicamente, el enfoque del modelo de islas basado en la optimización de arrecifes de coral alcanza la mejor solución en 19 casos de los 22 problemas de distribución en planta de áreas desiguales probados. Además, en 10 problemas de distribución en planta de áreas desiguales, el enfoque del modelo de islas basado en la optimización de arrecifes de coral supera los resultados alcanzados por propuestas anteriores. Al analizar el rendimiento de las versiones básica y extendida del enfoque del modelo de islas basado en la optimización de arrecifes de coral al resolver el problema de distribución en planta de áreas desiguales, parece que la versión extendida alcanza mejores resultados que la básica, ya que es capaz de superar los resultados logrados por la versión básica del algoritmo del modelo de islas basado en la optimización de arrecifes de coral en 4 instancias de los problemas de distribución en planta de áreas desiguales. Además, en los 18 problemas de distribución en planta de áreas desiguales restantes, la versión del modelo de islas basado en la optimización de arrecifes de coral extendida iguala a los mejores resultados logrados por la versión del modelo de islas basado en la optimización de arrecifes de coral básica. El sistema del modelo de islas basado en la optimización de arrecifes de coral también ha demostrado ser eficaz para abordar los problemas de distribución en planta de áreas desiguales de diferentes categorías según su tamaño, es decir, los problemas de distribución en planta de áreas desiguales pequeños, medianos y grandes. Esta efectividad se debe a la independencia existente entre cada arrecife (o isla), que alienta a explorar diferentes zonas del espacio de búsqueda en la misma cantidad de tiempo, lo que lleva a un uso eficiente de los recursos informáticos, y además mejora en un factor significativo el tiempo necesario para encontrar una solución óptima. Por otro lado, la migración también ayuda a aumentar la diversidad, debido a la independencia de las islas mencionadas anteriormente.

Dada la importancia de una distribución correcta de los departamentos en cualquier instalación, especialmente en las industriales, el impacto de la nueva propuesta es alto. El diseño de la planta afecta sustancialmente en la eficiencia

de la producción y es un tema relevante en cada construcción de planta nueva o en la remodelación de las existentes. Aquí, la minimización del costo de flujo de materiales se considera crucial para el éxito del diseño en términos de la economía de producción. Sin embargo, también es importante tener en cuenta la forma de cada departamento individual para mejorar el rendimiento de la planta. Estos dos requisitos son opuestos, en la forma en que cuanto peor es la relación de aspecto, mejor es el costo de manejo del material. Por lo tanto, encontrar el diseño que ofrezca el mejor costo posible de manejo de materiales y al mismo tiempo respetar las restricciones de relación de aspecto es un problema crítico que aún no se ha resuelto satisfactoriamente, especialmente en problemas de gran tamaño y relaciones complejas de flujo de materiales entre departamentos. En el enfoque actual, se ha obtenido una clara mejora en la mayoría de los problemas probados, por lo que es una línea prometedora que debe explorarse más a fondo.

8. Trabajo futuro

Como ha quedado patente, la investigación desarrollada en esta tesis doctoral ha resultado muy exitosa ya que se han logrado unos resultados excelentes que han superado los resultados previos en la mayoría de los casos. Por otro parte, con el desarrollo de la investigación realizada en esta tesis doctoral, también se ha abierto una nueva línea de investigación cuya exploración parece tener una proyección muy prometedora de trabajo futuro. En este sentido, vale la pena mencionar varias tendencias y posibles líneas de trabajo futuras, entre ellas podemos destacar las siguientes:

- En primer lugar, tras el análisis y estudio realizado en el estado del arte se ha podido ver que los trabajos recientes incluyen características cada vez más complejas y/o realistas de los sistemas de fabricación estudiados. Sin embargo, aún se necesita investigación. Así, por ejemplo, el uso de una tercera dimensión al diseñar una planta industrial es una consideración muy importante a tener en cuenta, por ejemplo, para seleccionar y optimizar recursos relacionados con el transporte vertical de piezas entre diferentes plantas. Esta línea se puede complementar con la incorporación de obstáculos, pasillos, ascensores, etc.
- Otro aspecto muy interesante a considerar como posible trabajo futuro es la consideración de industrias dinámicas donde la sistemas producción pueden tener diferentes requerimientos y características de producción que son cambiantes en un horizonte de corto plazo, por ejemplo, en temporadas de Navidad o campañas especiales de verano. Por lo que, la aplicación de las técnicas desarrolladas en esta tesis en el problema de distribución en planta en el entorno dinámico de producción parece ser muy interesante y prometedor.
- La consideración de una propuesta multiobjetivo podría ser muy

interesante, por ejemplo, con una propuesta interactiva que permita tener en cuenta la experiencia y/o preferencias del diseñador en el diseño de la distribución en planta, permitiendo también considerar factores cuantitativos como el coste de flujo de material. Esto haría que los diseños resulten más creativos y realistas. Esta línea de trabajo futuro podría extenderse a la consideración simultánea de las preferencias de varios diseñadores sobre un diseño de distribución en planta al mismo tiempo.

- Otra línea interesante de trabajo futuro podría ser la combinación de los enfoques presentados en esta tesis doctoral con otros métodos existentes, por ejemplo: evolución diferencial, redes neuronales artificiales, *machine learning*, entre otros, para mejorar el rendimiento de los enfoques propuestos.
- Otra posible línea trabajo futura podría ser el uso de simulación para incluir las preferencias del diseñador experto sin que se requiera mucho tiempo y/o trabajo durante la evaluación experta de la población y sin sobrecargar al diseñador. De esta manera, se podría reemplazar al humano por una simulación que diera una calificación a cada diseño.

9. Referencias

A. Gomez, Q.I. Fernandez, D. De la Fuente Garca, P.J. Garcia: Using genetic algorithms to resolve layout problems in facilities where there are aisles; *Int. J. Production Economics* 84 (2003) 271–282.

A. Kusiak, S. Heragu, The facility layout problem, *Eur. J. Operational Res.* 29 (198). 229–251.

A.M. Brintup, J. Ramsden, A. Tiwari, An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization, *Comput. Ind.* 58 (2007) 279–291.

A.M.E. Sikaroudi, K. Shahanaghi, Facility layout by collision detection and forceexertion heuristics, *J. Manuf. Syst.* 41 (2016) 21–30.

Abdullah Konak, Sadan Kulturel-Konak, Bryan A. Norman, Alice E. Smith: A new mixed integer programming formulation for FLD using flexible bays; *Operations Research Letters* 34 (2006) 660 – 672

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.

A. H. Land; A. G. Doig: An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, Vol. 28, No. 3. (Jul., 1960), pp. 497-520.

Alan R. McKendall Jr., Artak Hakobyan: Heuristics for the dynamic facility layout problem with unequal-area departments; *European Journal of Operational Research* 201 (2010) 171–182.

Alan R. McKendall Jr., J.R. Jaramillo: A tabu search heuristic for the dynamic space allocation problem; *Computers & Operations Research* 33 (2006) 768–789.

Alan R. McKendall Jr.: Improved Tabu search heuristics for the dynamic space allocation problem; *Computers & Operations Research* 35 (2008) 3347–3359.

Amine Drira, Henri Pierreval, Sonia Hajri-Gabouj: Facility layout problems: a survey; *Annual Reviews in Control* 31 (2007) 255–267.

Amy H.I. Lee, He-Yau Kang, Ching-Ter Chang: Fuzzy multiple goal programming applied to TFT-LCD supplier selection by downstream manufacturers; *Expert Systems with Applications* 36 (2009) 6318–6325.

Andrew KUSIAK and Sunderesh S. HERAGU: The facility Layout Problem; *European Journal of Operational Research* 29 (1987) 229–251.

Armour, G. C., & Buffa, E. S. (1963). A heuristic algorithm and simulation approach to relative allocation of facilities. *Management Science*, 9(2), 294–300.

Azadivar, F., & Wang, J. (2000). Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, 38(17), 4369–4383.

B Sumaovakul and P Thajchayapong, “A Knowledge Base to Assist a Heuristic Search Approach to Facility Layout,” *International Journal of Production Research* (p32, n I, 1994), pp141- 160.

B. Montreuil, U. Venkatadri, H.D. Ratliff, Generating a layout from a

design skeleton, IIE Trans., January, 1993.

B. Montreuil. A modeling framework for integrating layout design and flow network design. Proceedings of the material handling research colloquium, pages 43–58, 1990.

B.H. Ulutas, S. Kulturel-Konak, An artificial immune system based algorithm to solve unequal area facility layout problem, Expert Syst. Appl. 39 (2012) 5384–5395.

Badiru, A.B. And Arif, A., FLEXPART: facility layout expert system using fuzzy linguistic relationship codes. IIE Transactions. Vol 28, 1996, pp. 295–308.

Balakrishnan, J., & Cheng, C. H. (2000). Genetic search and the dynamic layout problem. Computers & Operations Research, 27(6), 587–593.

Balakrishnan, J.; Cheng, C.H.; Conway, D.J. and Lau, C.M., A hybrid Genetic Algorithm for the Dynamic Plant Layout Problem, International Journal of Production Economics, Vol. 86, 2003, pp. 107-120.

Banerjee, P., & Zhou, Y. (1995). Facilities layout design optimization with single loop material flow path configuration. International Journal of Production Research, 33(1), 183–204.

Baykasoğlu, A., Dereli, T., & Sabuncu, I. (2006). An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. Omega, 34(4), 385–396.

Baykasoğlu, A.; Gindy, N., A Simulated Annealing Algorithm for Dynamic Layout Problem. Computers & Operations Research. Vol. 28,

2001, pp. 1403-1426.

Bermejo, E., Chica, M., Damas, S., Salcedo-Sanz, S., and Cordón, O. (2018). Coral reef optimization with substrate layers for medical image registration. *Swarm and Evolutionary Computation*, 42:138 – 159.

Berna Haktanirlar Ulutas, Sadan Kulturel-Konak: An artificial immune system based algorithm to solve unequal area facility layout problem; *Expert Systems with Applications* 39 (2012) 5384–5395.

Berna Haktanirlar Ulutas, A. Attila Islier: A clonal selection algorithm for dynamic facility layout problems; *Journal of Manufacturing Systems* 28 (2009) pp. 123_131.

Bland, J.A., Layout of facilities using an ant system approach, *Engineering Optimization*, Vol. 32, N. 1, 1999, pp. 101-115.

Bozer, Y.A.; Meller, R.D. and Erlebacher, S.J., An improvement-type layout algorithm for single and multiple floor facilities, *Management Science*, Vol. 40, N. 7, 1994, pp. 918-932.

Brintup, A.M.; Ramsden, J. and Tiwari, A., An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization, *Computers in Industry*, Vol. 58, 2007, pp. 279-291.

Brintup, A.M.; Takagi, H.; Tiwari, A. and Ramsden, J., Evaluation of sequential, multi-objective, and parallel interactive genetic algorithms for multi-objective optimization problems, *Journal of Biological Physics and Chemistry*, Vol. 6, 2006, pp. 137-146.

Bryan A. Norman, Alice E. Smith: A continuous approach to considering uncertainty in facility design; *Computers & Operations Research* 33 (2006) 1760–1775

Buffa, E.S., Armour, G.C. and Vollman, T.E., Allocating facilities with CRAFT. *Harvard Business Review*, Vol. 42, 1964, pp. 136- 158.

Burkard R.E. and Rend F., A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operation Research*. Vol. 17, 1984, pp. 169–174.

C. Hicks: A genetic algorithm tool for designing manufacturing facilities in the capital goods industry; *Int. J. Production Economics* 90 (2004) 199–211.

Camacho-Gómez, C., Wang, X., Pereira, E., Díaz, I., and Salcedo-Sanz, S. (2018). Active vibration control design using the coral reefs optimization with substrate layer algorithm. *Engineering Structures*, 157:14 – 26.

Çela E., *The quadratic assignment problem: theory and algorithms*. Springer, Dordrecht, Netherlands, 1998.

Chan, C. and Tansri, H., A Study of Genetic Crossover Operations on the Facilities Layout Problem, *Computers & Industrial Engineering*, Vol. 26, 1994, pp. 537-550.

Chiang, W. C., and Kouvelis, P., An improved tabu search heuristic for solving facility layout design problems. *International Journal of Production Research*, Vol. 34, N. 9, 1996, pp. 2565–2585.

Chiang, W., & Chiang, C. (1998). Intelligent local search strategies for solving facility layout problems with the QAP formulation. *European Journal of operations research*, 106, 457-488.

Conway, D. and Ventakaramanan, M., Genetic Search and the Dynamic Facility Layout Problem, *Computers & Operations Research*, Vol. 21,

1994, pp. 995-960.

Corry, P. and Kozan, E. Ant colony optimisation for machine layout problems. *Computational Optimization and Applications*, Vol. 28, N. 3, 2004, pp. 287-310.

D. Tate, A. Smith, A genetic approach to the quadratic assignment problem, *Computers Ops. Res.* 22 (1) (1995) 73–83.

D. Tate, A. Smith, Unequal-area facility layout by genetic search, *IIE Trans.* 27 (4) (1995) 465–473.

D. van Camp, M. Carter, A. Vannelli, A nonlinear optimization approach for solving facility layout problems, *Eur. J. Operational Res.* 57 (1991) 174–189.

D.J.van Camp, M.W. Carter, and A. Vannelli, A nonlinear optimization approach for solving facility layout problems, *European Journal of Operational Research*, Vol. 57, 1992, pp.174-189.

Daniel Scholz, Anita Petrick, Wolfgang Domschke: StaTS= A Slicing Tree and Tabu Search based heuristic for the unequal area facility layout problem; *European Journal of Operational Research* 197 (2009) 166–178.

David M. Tate, Senior Member IIE and Alice E. Smith, Senior Member IIE1, Unequal Area Facility Layout Using Genetic Search by Department of Industrial Engineering, University of Pittsburgh-- Accepted to IIE Transactions, April 1994.

Deisenroth, M.P. and Apple, J.M., A computerized plant layout analysis and evaluation technique. Technical paper, Annual AIIE Conference, Norcross, GA, 1972.

Dorigo, M., *Optimization, Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano, Italie, 1992.

Drezner, Z., A heuristic procedure for the layout of a large number of facilities. *International Journal of Management Science*, Vol. 33, N. 7, 1980, pp. 907– 915.

Dunker, T., Radonsb, G. and Westkampera, E., A coevolutionary algorithm for a facility layout problem. *International Journal of Production Research*, Vol. 41, N. 15, 2003, pp. 3479–3500.

Dweiri, F., Fuzzy development of crisp activity relation- ship charts for facilities layout. *Computer & Industrial Engineering*, Vol. 36, N. 1, 1999, pp. 1-16.

E. Shayan, A. Chittilappilly, Genetic algorithm for facilities layout problems basedon slicing tree structure, *Int. J. Prod. Res.* 42 (2004) 4055–4067.

Eduardo Vila Gonçalves Filho, Alexandre Joséé Tiberti: A group genetic algorithm for the machine cell formation problem; *Int. J. Production Economics* 102 (2006) 1–21.

Eiben, A.E. and Smith, J.E., *Introduction to Evolutionary Computing*. Springer. 2nd edition, 2007.

Eida Nadirah Roslin, Siti Zawiah Md Dawal, Shamsuddin Ahmed: A review of Facility Layout selection models in manufacturing organizations (2010).

Enea, M., Galante, G. & Panascia, E. The facility layout problem approached using a fuzzy model and a genetic search. *J Intell Manuf* 16, 303–316 (2005)

Enea, M.; Galante, G. and Panascia, E., The Facility Layout Problem Approached using a Fuzzy Model and a Genetic Search, *Journal of Intelligence and Manufacturing*, Vol. 16, 2005, pp. 303-315.

F. Hillier, M. Connors, Quadratic assignment problem algorithms and location of indivisible facilities, *Manage. Sci.* 13(1) (1966) 42–57.

F.G. Paes, A.A. Pessoa, T. Vidal, A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem, *Eur. J. Oper. Res.* 256 (2017) 742–756.

Ficco, M., Esposito, C., Palmieri, F., and Castiglione, A. (2018). A coralreefs and game theory-based approach for optimizing elastic cloud resource allocation. *Future Generation Computer Systems*, 78:343 – 352.

Ficko, M.; Brezocnik, M. and Balic, J., Designing the Layout of Single- and Multiple-Rows Flexible Manufacturing System by Genetic Algorithms, *Journal of Materials Processing Technology*, Vol. 157-158, 2004, pp. 150-158.

Foulds, L.R. and Robinson, D.F., Graph Theoretic Heuristic for the Plant Layout Problem. *International Journal of Production Research*, Vol. 16, N. 1, 1978, pp. 27-37.

Francis R.L. and White, J. A., *Facility layout and location: an analytical approach*, Prentice Hall, Englewood Cliffs, NJ. 1974.

G. Abdou, S. Dutta, An integrated approach to facilities layout using expert systems, *Int. J. Prod. Res.* 28 (4) (1990) 685–708.

G. Aiello, M. Enea, G. Galante: A multi-objective approach to facility layout problem by genetic search algorithm and Electre method;

Robotics and Computer-Integrated Manufacturing 22 (2006) 447–455.

Giuseppe Aiello, Giada La Scalia, Mario Enea: A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding; *Expert Systems with Applications* 39 (2012) 10352–10358.

Goetschalckx, M., An interactive Layout Heuristic Based on Hexagonal Adjacency Graphs. *European Journal of Operational Research*. Vol. 63, N. 2, 1992, pp. 304-321.

Goldberg, D. E. and Lingle, R., Alleles, Loci and the Traveling Salesman Problem, *First International Conference on Genetic Algorithms and Their Applications*, 1985, pp. 154-159.

Gómez, A.; Fernández, Q.I.; De la Fuente García, D. and García, P.J., Using Genetic Algorithms to Resolve Layout Problems in Facilities where there are Aisles, *International Journal of Production Economics*, Vol. 84, 2003, pp. 271-282.

Grobelny, J., The ‘linguistic pattern’ method for a workstation layout analysis. *International Journal of Production Research*, Vol. 26, 1988, pp. 1779–1798.

H. Edwards, B. Gillett, M. Hale, Modular allocation technique _MAT_, *Management Sci.* 17 (3) (1970) 161–169.

H. Pierreval, C. Caux, J.L. Paris, F. Viguiet: Evolutionary approaches to the design and organization of manufacturing systems; *Computers & Industrial Engineering* 44 (2003) 339–364

Hasan, M. and Osman, Local Search Algorithms for the Maximal Planar Layout Problem. *International Transactions in Operational Research*,

Vol. 2, 1995, pp. 89-106.

Heragu, S.S. and Alfa, A.S., Experimental Analysis of Simulated Annealing Based Algorithms for the Layout Problem. *European Journal of Operational Research*, Vol. 57, 1992, pp. 190-202.

Heragu, S.S. and Kusiak, A., Efficient Models for the Facility Layout Problem. *European Journal of Operational Research*, Vol. 53, 1991, pp. 1-13.

Heragu, S.S. And Kusiak, A., Machine layout: an optimization and knowledge based approach. *International Journal of Production Research*. Vol. 28, N. 4, 1990, pp. 615–635.

Heragu, S.S. *Facilities design*. CRC Press. Taylor & Francis Group. 3rd edition, University of Louisville, Kentucky, USA, 2008.

Heragu, S.S., and Kusiak, A., Machine layout problem in flexible manufacturing systems, Working paper. Department of Mechanical and Industrial Engineering, University of Manitoba, Winnipeg, Manitoba, Canada. 1986.

Hicks, P.E. and Cowen, T.E., CRAFT-M for layout Rearrangement, *Industrial Engineering*, Vol. 8, 1976, pp. 30-35.

Hillier, F.S., Quantitative tools for plant layout analysis. *Journal of Industrial Engineering*, Vol. 14, 1963, pp. 33-40.

Holland, J.H. *Adaptation in Natural and Artificial Systems*, The MIT Press, Michigan, 1975.

Hu, M.H. and Wang, M.J., Using Genetic Algorithms on Facility Layout Problems. *International Journal of Advanced Manufacturing*

Technology, Vol. 23, 2004, pp. 301-310.

J. Koopmans, M. Beckmann, Assignment problems and location of economic activities, *Econometrica* 25 (1967) 53–76.

J. Liu, J. Liu, Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems, *Appl. Soft Comput.* 74 (2019) pp.167–189

J. Tompkins, J. White, Y. Bozer, J. Tanchoco, *Facilities Planning*, Wiley, New York, 4rd ed. edition, 2010.

J.F. Gonçalves, M.G. Resende, A biased random-key genetic algorithm for the unequal area facility layout problem, *Eur. J. Oper. Res.* 246 (2015) 86–107.

J.G. Kim and Y.D Kim. A space partitioning method for facility layout problems with shape constraints. *IIE Trans.*, 30(10):947–957, 1998.

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

Jaydeep Balakrishnan, Chun-Hung Cheng, Kam-Fai Wong: FACOPT= a user friendly FACility layout OPTimization system; *Computers & Operations Research* 30 (2003) 1625–1641.

K. L. Mak, Y. S. Wong and E T. S. Ghan: A genetic algorithm for facility layout problems; *Computer Integrated Manufacturing Systems* Vol. 11, No. 1-2, p. 113-127, 1998.

K. Tam, Genetic algorithms, function optimization and facility layout design, *Eur. J. Operational Res.* 63 (1992) 322–346.

K. Tam, Simulated annealing algorithm for allocating space to manufacturing cells, *Int. J. Prod. Res.* 30 (1991) 63–87.

K. Ueda, N. Fujii, I. Hatono, M. Kobayashi: Facility layout planning using self-organization method (2002).

K.D. Maniya, M.G. Bhatt: An alternative multiple attribute decision making methodology for solving optimal facility layout design selection problems; *Computers & Industrial Engineering* 61 (2011) 542–549.

Khalil, T.M., Facilities relative allocation technique (FRAT). *International Journal of Productions Research*, Vol. 11, N. 2, 1973, pp. 183–194.

Komarudin and Wong, K.Y., Applying Ant System for solving Unequal Area Facility Layout Problems. *European Journal of Operational Research*, Vol. 202, N. 3, 2010a, pp. 730-746.

Goldberg, D. E. and Komarudin. An improved ant system algorithm for unequal area facility layout problem. *University Technology Malaysia*, 2009.

Kouvelis, P., & Kim, M.W. (1992). Unidirectional loop network layout problem in automated manufacturing systems. *Operations Research*, 40, 533–550.

Kuan Yew Wong, Komarudin: Solving facility layout problems using flexible bay structure representation and ant system algorithm; *Expert Systems with Applications* 37 (2010b) 5523–5527.

Kulturel-Konak, S., Approaches to Uncertainties in Facilities Layout Problems: Perspectives at the Beginning of the 21st Century, *Journal of*

Intelligence and Manufacturing, Vol. 18, 2007, pp. 273-284.

L. Garcia-Hernandez, A. Arauzo-Azofra, L. Salas-Morera: An interactive genetic algorithm for the UAFLP.

L. Garcia-Hernandez, A. Arauzo-Azofra, L. Salas-Morera, H. Pierreval, E. Corchado, Facility layout design using a multi-objective interactive genetic algorithm to support the DM, *Expert Syst.* 32 (2015b) 94–107

L. Garcia-Hernandez, H. Pierreval, L. Salas-Morera, A. Arauzo-Azofra, An Interactive Genetic Algorithm with c-Means clustering for the Unequal Area Facility Layout Problem. In 10th International Conference on Intelligent Systems Design and Applications (ISDA), 2010, pp. 61-66.

L. Garcia-Hernandez, A. Arauzo-Azofra, L. Salas-Morera, H. Pierreval, E. Corchado, 2013a. Recycling plants layout design by means of an interactive genetic algorithm. *Intell. Autom. Soft Comput.* 19 (3), 457–468.

L. Garcia-Hernandez, H. Pierreval, L. Salas-Morera, A. Arauzo-Azofra, Handling qualitative aspects in Unequal Area Facility Layout Problem: An Interactive Genetic Algorithm, *Applied Soft Computing*, Volume 13, Issue 4, 2013b, pp. 1718-1727.

L. Garcia-Hernandez, J.M. Palomo-Romero, L. Salas-Morera, A. Arauzo-Azofra, H. Pierreval, A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem, *Expert Syst. Appl.* 42(2015a) 4697–4708

L. Garcia-Hernandez, L. Salas-Morera, C. Carmona-Muñoz, A. Abraham, S. Salcedo-Sanz, A novel multi-objective Interactive Coral

Reefs Optimization algorithm for the Unequal Area Facility Layout Problem, *Swarm and Evolutionary Computation*, Volume 55, 2020a, 100688.

L. Garcia-Hernandez; A. Araúzo-Azofra; H. Pierreval, and L. Salas-Morera, Encoding Structures and Operators Used in Facility Layout Problems with Genetic Algorithms, *ISDA '09: Proceedings of the 2009 Ninth International Conference on Intelligent Systems Design and Applications*, Washington, DC, USA: IEEE Computer Society, 2009, pp. 43-48.

L. Garcia-Hernandez, M. Pérez-Ortiz, A. Araúzo-Azofra, L. Salas-Morera, C. Hervás-Martínez, An evolutionary neural system for incorporating expert knowledge into the UA-FLP, *Neurocomputing*, Volume 135, 2014, Pages 69-78, ISSN 0925-2312.

L. Garcia-Hernandez, J.A. Garcia-Hernandez, L. Salas-Morera, C. Carmona-Munoz, Norah S. Alghamdi, J. Valente de Oliveira, S. Salcedo-Sanz., Addressing Unequal Area Facility Layout Problems using a Coral Reef Optimization Algorithm with Substrate Layers, *Engineering Applications of Artificial Intelligence*, *Expert Syst.* (2020b).

Leonardo Chwif, Marcos R. Pereira Barretto, Lucas Antonio Moscato: A solution to the facility layout problem using simulated annealing; *Computers in Industry* 36 _1998. 125–132.

Liu, Q. and Meller, R.D., A sequence-pair representation and MIP-model-based heuristic for the facility layout problem with rectangular departments. *IEE Transactions*. Vol. 39, N. 4, 2007, pp. 377-394.

M J Rosenblatt and B Golany, ‘A Distance Assignment Approach to the

Facility Layout Problem,” *European Journal of Operational Research* (p57, 1992), pp253-270.

M. Adel El-Baz: A genetic algorithm for facility problems of different manufacturing environments; *Computers & Industrial Engineering* 47 (2004) 233–246

M. Babbar-Sebens, B.S. Minsker. Interactive genetic algorithm with mixed initiative interaction for multi-criteria ground water monitoring design *Applied Soft Computing*, 12 (2012), pp. 182-195.

M. Solimanpur, Prem Vrat, Ravi Shankar: An ant algorithm for the single row layout problem in flexible manufacturing systems; *Computers & Operations Research* 32 (2005) 583–598.

M.G. Del Río-Cidoncha, J.E. Iglesias, J. Martínez-Palacios: A comparison of floor plan design strategies in architecture and engineering; *Automation in Construction* 16 (2007) 559–568.

Malakooti, B. and Tsurushima, A., An expert system using priorities for solving multiple-criteria facility layout problems. *International Journal of Production Research*. Vol. 27, N. 5, 1989, pp. 793–808.

Matsuzaki, K.; Irohara, T. and Yoshimoto, K., Heuristic Algorithm to Solve the Multi-Floor Layout Problem with Consideration of Elevator Utilization. *Computers & Industrial Engineering*, Vol. 36, 1999, pp. 487-502.

McKendall Jr., A.R. and Shang, J. Hybrid ant systems for the dynamic facility layout problem. *Computers and Operations Research*, Vol. 33, N. 3, 2006, pp. 790-803.

Medeiros, I. G., Xavier, J. C., and Canuto, A. M. P. (2015). Applying

the coral reefs optimization algorithm to clustering problems. In 2015 International Joint Conference on Neural Networks (IJCNN), pages 1–8.

Meir J Rosenblatt, “The Dynamics of Plant Layout,” *Management Science* (p32, n I, 1986) pp76-86.

Meller, R.D. and Bozer, Y.A., A new simulated annealing algorithm for the facility layout problem. *International Journal of Production Research*. Vol. 34, 1996, pp. 1675-1692.

Meller, R.D. and Bozer, Y.A., Alternative approaches to solve multi-floor facility layout problem. *Journal of Manufacturing Systems*, Vol. 16, N. 3, 1997, pp. 192-203.

Meller, R.D.; Narayanan, V. and Vance, P.H., Optimal Facility Layout Design. *Operations Research Letters*, Vol. 23, 1999, pp. 117-127.

Michalewicz, Z.; Dasgupta, D.; Le Richie, R. and Schoenauer. M., Evolutionary Algorithms for Constrained Engineering Problems. *Computers & Industrial Engineering*. Vol. 30, 1996, pp. 851-870.

Dong, Chang Wu, Forest Hou: Shortest path based simulated annealing

Muther, R. *Systematic Layout Planning*. Cahnners Books. 2d ed., Boston, 1973.

Nearchou, A. C., Meta-heuristics from nature for the loop layout design problem. *International Journal of Production Economics*, Vol. 101, N. 2, 2006, pp. 312–328.

P. Banerjee, S. Y. Nof: Knowledge base facility planning.

Pierreval, H.; Caux, C.; Paris, J.L. and Viguier, F., Evolutionary

approaches to the design and organization of manufacturing systems, *Computers & Industrial Engineering*, Vol. 44, 2003, pp. 39-364.

Potts, C.N., and Whitehead, J.D., Workload balancing and loop layout in the design of a flexible manufacturing system. *European Journal of Operational Research*, Vol. 129, N. 2, 2001, pp. 326–336.

R. Liggett, Automated facilities layout= past, present and future; *Automation in Construction* 9 (2000) 197–215.

R. Tavakkoli-Moghaddain, E. Shayan: Facilities layout design by genetic algorithms; *Computers ind. Engng* Vol. 35, Nos 3-4, pp. 527-530, 1998.

R.D. Meller and K.Y. Gau, The facility layout problem: Recent and emerging trends and perspectives, *Journal of Manufacturing Systems*, vol.15, issue 5, 1996, pp. 351-366.

Ramkumar, A.S., Ponnambalam, S.G., Hybrid ant colony system for solving quadratic assignment formulation of machine layout problems. 2006 IEEE Conference on Cybernetics and Intelligent Systems, art. no. 4017845, 2006.

Ramkumar, A.S.; Ponnambalam, S.G. and Jawahar, N., A population-based hybrid ant system for quadratic assignment formulations in facility layout design. *International Journal of Advanced Manufacturing Technology*, Vol. 44, N. 5-6, 2009, pp. 548-558.

Raoot, A.D. and Rakshit, A., A fuzzy heuristic for the quadratic assignment formulation to the facility layout problem *International Journal of Production Research*, Vol. 32, N. 3, 1994, pp. 563–581.

Rezazadeh, H.; Ghazanfari, M.; Saidi-Mehrabad, M. and Jafar Sadjadi,

S., An extended discrete particle swarm optimization algorithm for the dynamic facility layout problem, *Journal of Zhejiang University: Science A*, Vol. 10, N. 4, 2009, pp. 520-529.

Russell D. Meller, Weiping Chen, Hanif D. Sherali: Applying the sequence-pair representation to optimal facility layout designs; *Operations Research Letters* 35 (2007) 651 – 659.

Russell D. Meller, Zeynep Kirkizoglua, Weiping Chen: A new optimization model to support a bottom-up approach to facility design; *Computers & Operations Research* 37 (2010) 42-49.

S. Kang, J. Chae, Harmony search for the layout design of an unequal area facility, *Expert Syst. Appl.* 79 (2017) 269–281.

S. Kulturel-Konak, A. Konak, A new relaxed flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem, *Eng. Optim.* 43 (2011) 1263–1287.

S. Salcedo-Sanz, D. Gallo-Marazuela, A. Pastor-Sánchez, L. Carro-Calvo, A. Portilla-Figueras and L. Prieto, "Offshore wind farm design with the Coral Reefs Optimization algorithm," *Renewable Energy*, vol. 63, pp.109-115, 2014.

Sadan Kulturel-Konak: A linear programming embedded probabilistic tabu search for the unequal-area facility layout problem with flexible bays; *European Journal of Operational Research* 223 (2012) 614–625.

Sahin, R.; Ertogral, K. and Turkbey, O., A simulated annealing heuristic for the dynamic layout problem with budget constraint, *Computers & Industrial Engineering*, Vol. 59, N. 2, 2010, pp. 308-313.

Salas-Morera, L., Cubero-Atienza, A.J. and Ayuso-Munoz, R.,

Computer-Aided Plant Layout, *Informacion Tecnológica*, Vol. 7, 1996, pp. 39-46.

Salas-Morera, L., García-Hernández, L., Antolí-Cabrera, A. et al. Using eye-tracking into decision makers evaluation in evolutionary interactive UA-FLP algorithms. *Neural Comput & Applic* (2020)

Salcedo-Sanz, S. (2017). A review on the coral reefs optimization algorithm: new development lines and current applications. *Progress in Artificial Intelligence*, 6(1):1–15.

Salcedo-Sanz, S., Camacho-Gómez, C., Magdaleno, A., Pereira, E., and Lorenzana, A. (2017). Structures vibration control via tuned mass dampers using a co-evolution coral reefs optimization algorithm. *Journal of Sound and Vibration*, 393:62 – 75.

Salcedo-Sanz, S., Camacho-Gómez, C., Mallol-Poyato, R., Jiménez-Fernández, S., and DelSer, J. (2016). A novel coral reefs optimization algorithm with substrate layers for optimal battery scheduling optimization in micro-grids. *Soft Computing*, 20(11):4287–4300.

Salcedo-Sanz, S., Del Ser, J., Landa-Torres, I., Gil-López, S., and Portilla-Figueras, A. (2013). The coral reefs optimization algorithm: an efficient meta-heuristic for solving hard optimization problems. In *Proceedings of the 15th International Conference on Applied Stochastic Models and Data Analysis (ASMDA2013)*, Matar'ó, pages 751–758.

Salcedo-Sanz, S., Del Ser, J., Landa-Torres, I., Gil-López, S., and Portilla-Figueras, J. A. (2014a). The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems. *Scientific World Journal*, 2014.

Salcedo-Sanz, S., García-Díaz, P., Portilla-Figueras, J., Ser, J. D., and Gil-López, S. (2014b). A coral reefs optimization algorithm for optimal mobile network deployment with electromagnetic pollution control criterion. *Applied Soft Computing*, 24:239 – 248.

Salcedo-Sanz, S., Pastor-Sánchez, A., Prieto, L., Blanco-Aguilera, A., and García-Herrera, R. (2014c). Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization – extreme learning machine approach. *Energy Conversion and Management*, 87:10 – 18.

Salcedo-Sanz, S., Sanchez-Garcia, J., J.A., P.-F., Jimenez-Fernandez, S., and Ahmadzadeh, A. (2014d). A coral-reef optimization algorithm for the optimal service distribution problem in mobile radio access networks. *Transactions on Emerging Telecommunications Technologies*, 25(11):1057–1069.

Sanjay Sharma, Narayan Agrawal: Selection of a pull production control policy under different situations for a manufacturing system by AHP- algorithm; *Computers & Operations Research* 36 (2009) 1622-1632.

Seehof, J. M., and Evans, W. O., Automated layout design program. *The Journal of Industrial Engineering*, Vol. 18, 1967, pp. 690–695.

Shayan, E. and Chittilappilly, A., Genetic Algorithm for facility layout problems based on slicing tree structure, *International Journal of Production Research*, Vol. 42, N. 19, 2004, pp. 4055-4067.

Sherali, H.D., Fraticelli, B.M.P., and Meller, R.D., Enhanced Model Formulations for Optimal Facility Layout. *Operations Research*. Vol. 51, N. 4, 2003, pp. 629- 644.

Shore, R.H. and Tompkins, J.A., Flexible facilities design, AIIE Transactions, Vol 12, N. 2, 1980, pp. 200-205.

Singh, J.; Foster, B.T. and Heragu, S.S., A Genetic Algorithm for the Unequal Area Facility Layout Problem, Computers & Operations Research, Vol. 25, 1998, pp. 583–594.

Singh, S.P. and Sharma, R.R.K., A Review of Different Approaches to the Facility Layout Problems, International Journal of Advanced Manufacturing Technology, Vol. 30, 2006, pp. 425–433.

Singh, S.P., Ant system embedded with local search for solving facility layout problem. Communications in Computer and Information Science, Vol. 70, 2010, pp. 621-628.

Sirinaovakul, B. and Thajchayapong, P., A knowledge base to assist a heuristic search approach to facility layout. International Journal of Production Research. Vol. 32, N. 1, 1994, pp. 141–160.

Solimanpur, M., Vrat, P., & Shankar, R. (2005). An ant algorithm for the single row layout problem in flexible manufacturing systems. Computers & Operations Research, 32(3), 583–598.

Soroush Safarzadeh, Hamidreza Koosha, Solving an extended multi-row facility layout problem with fuzzy clearances using GA, Applied Soft Computing, Volume 61, 2017, Pages 819-831

Syswerda, G., Uniform crossover in genetic algorithms. Proceedings of the Third International Conference on Genetic Algorithms, Los Altos, CA, 1989, pp. 2-9.

T. Sawa, A. Nagao, T. KANBE, and I. Shirakawa, A Method for Rectangle Packing Problem, Technical Report of IEICE., CAS97-38,

VLD97-38, DSP97-53, 1997-June.

T. Vollmann, C. Nugent, R. Zartler, A computerized model for office layout, *J. Ind. Eng.* 19 (7) (1968) 321–329.

Talbi E.G.; Roux O.; Fonlupt C., and Robillard D., Parallel ant colonies for quadratic assignment problem. *Future Generation Computer Systems*, Vol. 17, 2001, pp. 441–449.

Tam, K.Y., “A simulated annealing algorithm for allocating space to manufacturing cells.” *International Journal of Production Research*, 1992 30, 63-87.

Tam, K.Y., A Simulated Annealing for Allocating Space to Manufacturing Cells. *European Journal of Operational Research*, Vol. 63, 1992, pp. 322-346.

Tam, K.Y., and Chan, S.K., Solving facility layout problems with geometric constraints using parallel genetic algorithms: Experimentation and findings. *International Journal of Production Research*, Vol. 36, N. 12, 1998, pp. 3253– 3272.

Thomas Dunker, Günter Radons, Engelbert Westkämper: Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem; *European Journal of Operational Research* 165 (2005) 55–69.

Thürmer, G. and Wüthrich, C.A. (1997), Normal Computation for Discrete Surfaces in 3D Space. *Computer Graphics Forum*, 16: C15-C26. doi:10.1111/1467-8659.00138.

Tijen Ertay, Da Ruan, Umut Rifat Tuzkaya: Integrated data envelopment analysis and analytic hierarchy for the facility layout

design in manufacturing systems; *Information Sciences* 176 (2006) 237–262.

Tompkins, J.A., & Reed, J. R. (1976). An applied model for the facilities design problem. *International Journal of Production Research*, 14, 583–595.

Tompkins, J.A.; White, J.A.; Bozer, Y.A. and Tanchoco, J.M.A., *Facilities Planning*, Wiley, 3rd ed., New York, 2003.

Tong, X., *SECOT: A Sequential Construction Technique For Facility Design*. University of Pittsburg: Doctoral Dissertation, 1991.

Tsuchiya, K.; Bharitkar, S. and Takefuji, Y., A neural network approach to facility layout problems. *European Journal of Operational Research*. Vol. 89, 1996, pp. 556– 563.

Tuzkaya, U., Ertay, T., An application of fuzzy ahp/dea methodology for the facility layout design in the presence of both quantitative and qualitative data. In: *Applied Computational Intelligence – Proceedings of the 6th International FLINS Conference*, 2004, pp. 507–512.

Wong, K.Y. and See, P.C., A hybrid ant colony optimization algorithm for solving facility layout problems formulated as quadratic assignment problems. *Engineering Computations*, Vol. 27, N. 1, 2010, pp. 117-128.

Wu, Y. and Appelton, E. The Optimisation of Block Layout and Aisle Structure by a Genetic Algorithm. *Computers & Industrial Engineering*, Vol. 41, 2002, pp. 371-387.

Yan, C., Ma, J., Luo, H., and Patel, A. (2019). Hybrid binary coral reefs optimization algorithm with simulated annealing for feature selection in high-dimensional biomedical datasets. *Chemometrics and Intelligent*

Laboratory Systems, 184:102 – 111.

Yang, Z., Zhang, T., and Zhang, D. (2016). A novel algorithm with differential evolution and coral reef optimization for extreme learning machine training. *Cognitive Neurodynamics*, 10(1):73–83.

Yavuz A. Bozer, Russell D. Meller, Steven J. Erlebacher: An Improvement-type layout algorithm for multiple floor facilities; technical Report 91-11.

Yavuz A. Bozer, Russell D. Meller: A construction-type layout algorithm for multi-floor facilities with capacitated lifts; technical

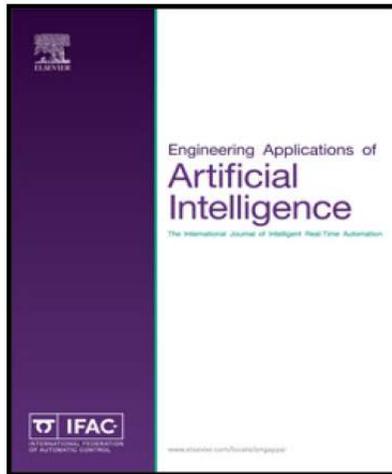
Z. Michalewicz, *Genetic Algorithms+Data Structures Evolution Programs*, Springer-Verlag, Berlin, 1992.

Zoller, K. and Adendorff, K., Layout planning by computer simulation. *AIIE Transactions*, Vol. 4, N. 2, 1972, pp. 116–125.

A. Publicaciones

TÍTULO: Addressing Unequal Area Facility Layout Problems using a Coral Reef Optimization Algorithm with Substrate Layers, Engineering Applications of Artificial Intelligence.

AUTORES: L. Garcia-Hernandez, **J.A. Garcia-Hernandez**, L. Salas-Morera, C. Carmona-Munoz, Norah S. Alghamdi, J. Valente de Oliveira, S. Salcedo-Sanz.



ISSN: 0952-1976

DOI: *por determinar (aceptado)*.

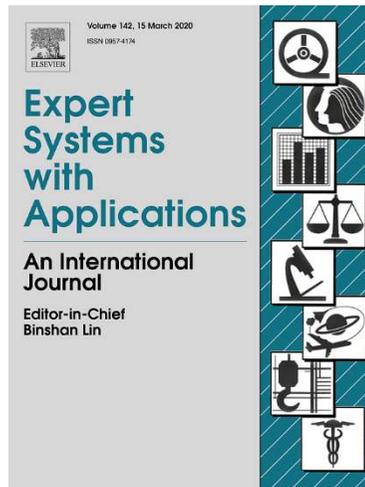
RANKING JCR 2018

FACTOR DE IMPACTO: 3.526

Categoría de JCR	Ranking	Cuartil
Computer Science, Artificial Intelligence	19 de 62	Q2
Engineering, Electrical & Electronic	36 de 134	Q2
Operations Research & Management Science	69 de 266	Q2
Engineering Multidisciplinary	15 de 88	Q1

TÍTULO: Applying the Coral Reefs Optimization Algorithm for solving Unequal Area Facility Layout Problems.

AUTORES: L. Garcia-Hernandez, L. Salas-Morera, **J.A. Garcia-Hernandez**, S. Salcedo-Sanz, J. Valente de Oliveira.



ISSN: 0957-4174

DOI: <https://doi.org/10.1016/j.eswa.2019.07.036>

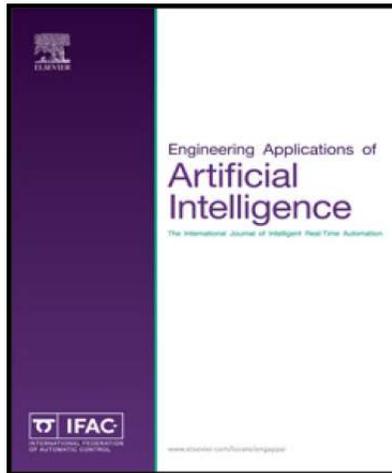
RANKING JCR 2018

FACTOR DE IMPACTO: 4.292

Categoría de JCR	Ranking	Cuartil
Computer Science, Artificial Intelligence	24 de 134	Q1
Engineering, Electrical & Electronic	49 de 266	Q1
Operations Research & Management Science	7 de 84	Q1

TÍTULO: A novel Island Model based on Coral Reefs Optimization algorithm for solving the unequal area facility layout problem.

AUTORES: L. Garcia-Hernandez, L. Salas-Morera, C. Carmona-Muñoz, **J.A. Garcia-Hernandez**, S. Salcedo-Sanz.



ISSN: 0952-1976

DOI: <https://doi.org/10.1016/j.engappai.2019.103445>

RANKING JCR 2018

FACTOR DE IMPACTO: 3.526

Categoría de JCR	Ranking	Cuartil
Computer Science, Artificial Intelligence	19 de 62	Q2
Engineering, Electrical & Electronic	36 de 134	Q2
Operations Research & Management Science	69 de 266	Q2
Engineering Multidisciplinary	15 de 88	Q1

