

**SOLUCIÓN AL PROBLEMA DE DISTRIBUCIÓN DE PLANTA (FLP) DE
MÚLTIPLES NIVELES, UN ELEVADOR Y DEPARTAMENTOS DESIGUALES, A
TRAVÉS DE MÉTODOS METAHEURÍSTICOS**

**JULIÁN ALBERTO GÓMEZ
DIEGO ALEXANDER RUEDA PRADA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
BUCARAMANGA**

2016

**SOLUCIÓN AL PROBLEMA DE DISTRIBUCIÓN DE PLANTA (FLP) DE
MÚLTIPLES NIVELES, UN ELEVADOR Y DEPARTAMENTOS DESIGUALES, A
TRAVÉS DE MÉTODOS METAHEURÍSTICOS**

**JULIÁN ALBERTO GÓMEZ
DIEGO ALEXANDER RUEDA PRADA**

Trabajo de grado para optar al título de Ingeniería Industrial

**Director:
CARLOS EDUARDO DÍAZ BOHÓRQUEZ
Msc. Ingeniería Industrial**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
BUCARAMANGA**

2016

CONTENIDO

	Pág.
INTRODUCCIÓN	12
1. GENERALIDADES DEL PROYECTO.....	16
1.1 PLANTEAMIENTO DEL PROBLEMA	16
1.2 JUSTIFICACIÓN DEL PROYECTO	17
1.7 OBJETIVOS.....	18
1.7.1 Objetivo general.....	18
1.7.2 Objetivos específicos	18
1.8 METODOLOGÍA	19
1.8.1 Revisión bibliográfica	19
1.8.2 Construcción del modelo.....	20
1.8.3 Programación y validación del algoritmo.....	20
2. MARCO TEÓRICO	21
2.1 FACILITY LAYOUT PROBLEM	21
2.2 OPTIMIZACIÓN COMBINATORIA.....	23
2.2.1 Tipos de complejidad computacional	25
2.3. VARIACIONES DEL PROBLEMA DE DISTRIBUCIÓN DE PLANTA	26
2.3.1 FLP considerando la variedad y volumen de los productos	27
2.3.2 FLP considerando los puntos de <i>Pick Up</i> y <i>Drop Off</i> de los puestos de trabajo.....	28
2.3.3 FLP con retrocesos y saltos.....	28
2.3.4 FLP considerando las dimensiones y formas de los departamentos	29
2.3.5 FLP Multi-floor (Con más de un nivel).....	34
2.4. MÉTODOS DE SOLUCIÓN PARA EL PROBLEMA FLP MULTI-FLOOR.....	38

2.4.1 Métodos exactos.....	39
2.4.2 Heurísticas.....	45
2.4.3 Metaheurísticas.....	47
2.4.4 Recocido simulado en la optimización combinatoria y problemas del tipo FLP	54
3. REVISIÓN DE LITERATURA.....	57
4. MARCO DE ANTECEDENTES.....	65
5. ALGORITMO DE RECOCIDO SIMULADO APLICADO AL PROBLEMA DE DISTRIBUCIÓN DE PLANTA (FLP) DE Múltiples NIVELES, UN ELEVADOR Y DEPARTAMENTOS de áreas desIGUALES.....	70
5.1 DESCRIPCIÓN DEL ALGORITMO.....	71
5.1.1 Etapa de parametrización y generación del punto de partida	71
5.1.2. Cálculo del costo vertical	72
5.1.3. Cálculo del costo horizontal	75
5.1.4 Pseudocódigo para el algoritmo de solución para el problema FLP Multi- Floor.....	77
6. VALIDACIÓN DEL ALGORITMO	79
7. RESULTADOS.....	82
8. CONCLUSIONES	87
9. RECOMENDACIONES	89
BIBLIOGRAFÍA.....	90
ANEXOS	102

LISTA DE FIGURAS

	Pág.
Figura 1. Dominios de las clases de complejidad computacional	26
Figura 2. Distribuciones de planta según algoritmos CRAFT, MULTIPLE y GTH..	30
Figura 3. Ejemplo de la regularización de una distribución con departamentos irregulares	31
Figura 4. Distribución de planta según algoritmo híbrido de Lee & Lee	32
Figura 5. Patrones de curvas de llenado	34
Figura 6. Distribución de planta de más de un nivel	35
Figura 7. Ejemplo construcción de un grafo para problemas de Distribución de Planta.....	42
Figura 8. Primera etapa del algoritmo.....	72
Figura 9. Ejemplo de Gráfica de Función Objetivo – Primera Etapa.....	74
Figura 12. Construcción de la distribución de planta a partir de la curva de llenado	77
Figura 13. Gráfico de relación entre el número de elevadores y el costo total.....	80

LISTA DE TABLAS

	Pág.
Tabla 1. Valores de costo generados por el algoritmo considerando $n=20$	82
Tabla 2. Costos totales obtenidos de la muestra $n=20$	83

LISTA DE ANEXOS

	Pág.
ANEXO A. Valores de áreas para cada departamento	102
ANEXO B.	105
ANEXO C. Información obtenida cada vez que se corre el algoritmo (3 pisos, 40 departamentos).....	172
ANEXO D. Iteraciones del algoritmo con $n=78$	177

RESUMEN

TÍTULO: Solución al problema de distribución de planta (FLP) de múltiples niveles, un elevador y departamentos desiguales, a través de métodos metaheurísticos.*

AUTORES: Julián Alberto Gómez, Diego Alexander Rueda.**

PALABRAS CLAVE: Distribución de planta, Metaheurísticas, Recocido Simulado, Optimización, MATLAB.

En este trabajo se aborda el problema de distribución de planta FLP, que tiene como objetivo minimizar los costos asociados al transporte de material a partir de la maximización de adyacencias entre departamentos. Para ello, se toma como punto de partida una documentación y revisión bibliográfica con el fin de definir una alternativa de solución al problema acorde con las restricciones planteadas.

En base a la revisión anterior, se define como método de solución la metaheurística de Recocido Simulado ya que ésta permite obtener una buena solución en problemas de FLP de grandes instancias y numerosas restricciones. No se tienen en cuenta los métodos exactos ya que estos son ineficientes al enfrentarse a problemas de alta complejidad, es decir, sólo brindan una solución óptima cuando el problema tiene pocas instalaciones, departamentos y restricciones.

La complejidad existente en este problema de distribución de planta reside en las siguientes consideraciones: se suponen departamentos de formas irregulares, áreas desiguales conocidas, múltiples niveles y un elevador de ubicación fija, que permite la interacción entre estos.

Teniendo en cuenta la revisión bibliográfica realizada, se diseña y construye una herramienta informática en lenguaje MATLAB que aplica el método de recocido simulado al presente problema. A continuación, se desarrolla un análisis del desempeño de la herramienta basado en criterios de tiempo computacional y los resultados obtenidos, en contraste con una instancia de referencia encontrada en la literatura.

*Proyecto de Grado

**Facultad de Ingenierías Físico-Mecánicas. Escuela de Estudios Industriales y Empresariales. Director: Msc. Carlos Eduardo Díaz Bohórquez.

ABSTRACT

TITTLE: Metaheuristics methods applied to solving the Facility Layout Problem (FLP) with multiple floors, one elevator and unequal departments.*

AUTHORS: Julián Alberto Gómez, Diego Alexander Rueda.**

KEYWORDS: Facility Layout Problem, Metaheuristics, Simulated Annealing, Optimization, MATLAB.

This project studies the facility layout problem (FLP) with the goal of minimizing the total material handling cost by maximizing the closeness between the existing departments within the facility. To do so, a literature review of the FLP is made as a starting point for deciding the solution method that best suits the constraints formulated for the project.

Based on this review the simulated annealing solution method is chosen since it is capable of obtaining good quality solutions for complex FLP problems with big instances and numerous restrictions. Exact methods are not taken into account as they are inefficient when faced with high complexity problems, precluding the detection of an optimal solution since exact formulations are only applicable to problems with a few amount of departments and constraints.

The complexity of the proposed facility layout problem resides within the following considerations: the departments' shapes are presuming to be irregular, their areas are known to be unequal, the facility has multiple levels and a single elevator with a fixed and known location allowing the material flux between the departments.

Starting from the compiled information, a computational tool in MATLAB software is designed and developed applying the Simulated Annealing solution method to the presented problem. Then, a performance test for the tool is made, based on the criteria of computing time and the quality of the obtained results when compared with a research paper found in the literature.

*Thesis degree project

**Facultad de Ingenierías Físico-Mecánicas. Escuela de Estudios Industriales y Empresariales. Director: Msc. Carlos Eduardo Díaz Bohórquez.

INTRODUCCIÓN

El problema de distribución de planta es uno de los problemas más estudiados por los sistemas de manufactura¹ debido al impacto que éste posee en los costos, los indicadores de productividad y el tiempo de ciclo de los productos; ya que una distribución ineficiente de los centros de trabajo perjudica considerablemente la industria al incurrir en despilfarros de tiempo y dinero.

Cuando las diferentes empresas deciden establecer sus instalaciones en nuevas áreas se ven enfrentadas a la tarea de encontrar un espacio apropiado para la ubicación de la fábrica contemplando una oferta de terrenos de espacio insuficiente y de alto costo.

Considerando dichas condiciones las empresas a menudo optan por localizar sus instalaciones productivas en estructuras de más de un piso, lo cual obliga a las industrias a buscar la distribución más eficiente de sus centros de trabajo de manera que ésta abarque los diferentes niveles de la infraestructura contando con la asistencia de un elevador que comunique los pisos.

En éste libro, se abordará el problema de *Facility Layout Problem* (FLP) enfocándose en una de sus variaciones en la cual el número de ascensores y el área de los departamentos están determinadas para los múltiples (2 o más) niveles que conforman la estructura. Para resolver el problema de *FLP Multi-floor* se hace necesario recurrir a métodos de solución no exactos, ya que es un problema de alta complejidad computacional.

¹ Surya P. Singh & Renduchintala Sharma. A review of different approaches to the facility layout problems. *The International Journey of Advanced Manufacturing Technology*. 5-6. p 425-433. 2005

La dificultad del problema *FLP Multi-floor* hace necesario el uso de metodologías de solución que no solo consideren la calidad de la distribución generada sino el tiempo de cómputo que se requiere para obtenerla, dicho esto se plantea el uso de la metaheurística de Recocido Simulado en este proyecto, la cual permite una aproximación bastante cercana a una solución óptima global sin incurrir en tiempos de procesamiento demasiado altos.

Consecuentemente se construyó un pseudocódigo en lenguaje Matlab que permite abordar el problema FLP Multifloor mediante la técnica anteriormente descrita, validada a través de la solución comparativa de problemas encontrados en la literatura. Posteriormente se presentan las conclusiones obtenidas a partir de la siguiente investigación y recomendaciones a considerar para posteriores estudios en el tema abordado.

1. GENERALIDADES DEL PROYECTO

1.1 PLANTEAMIENTO DEL PROBLEMA

El origen del problema de distribución de planta FLP Multifloor se evidencia en el momento en el que las empresas buscan instalar o, en determinados casos, trasladar sus sistemas productivos a nuevas instalaciones buscando aprovechar al máximo el espacio disponible de los terrenos considerando el alto costo de los mismos, por lo cual es muy común que se elijan infraestructuras de más de un nivel para éste propósito.

Al llevar a cabo la ubicación del sistema productivo de la empresa en una instalación de más de un nivel no se debe perder de vista que la distribución de centros de trabajo, efectuada dentro de dicha estructura, debe minimizar los costos de manejo de materiales y contribuir a la eficiencia del proceso productivo; teniendo en cuenta los requerimientos operativos y logísticos de la empresa y el espacio disponible.

Para dar solución a dicha problemática, en este proyecto se establecen tres consideraciones primarias: La instalación a modelar consta de dos o más niveles o pisos; el número de niveles de la estructura es irrestricto y determinado por el programador. Los departamentos a ubicar se suponen de áreas desiguales y formas irregulares, lo cual significa que los departamentos a ubicar no son necesariamente rectangulares; sino que también pueden adoptar forma de polígonos de diferentes áreas y formas. Finalmente para el transporte vertical del material entre niveles se empleará un ascensor con una ubicación fija de capacidad irrestricta y que abastece a cada nivel de la estructura. Dicha ubicación es determinada por el programador. En este proyecto se utilizará la metaheurística de Recocido Simulado, identificada por las siglas SA (*Simulated Annealing*, en inglés) para dar solución a la

problemática planteada ya que dicha técnica permite llevar a cabo una búsqueda intensiva dentro del espacio de soluciones factibles de la función objetivo para llegar a una buena solución evitando que el proceso se detenga en óptimos locales.

1.2 JUSTIFICACIÓN DEL PROYECTO

Cuándo las empresas deciden trasladar sus sistemas productivos a instalaciones de más de un nivel lo hacen comúnmente motivadas por el alto costo que representa la adquisición y adecuación de un predio de área suficiente para llevar a cabo sus operaciones gracias a la alta valorización de los terrenos.

Sin embargo la eficiencia y los costos de manejo de dichos sistemas productivos deben optimizarse al ser trasladados a infraestructuras de más de un nivel, lo cual otorga validez al análisis del problema FLP Multifloor siendo éste el medio por el cual se busca optimizar los elementos anteriormente mencionados además de una aplicabilidad en el mundo real.

Con este proyecto se planteará un nuevo método de solución para el tipo de problema descrito anteriormente, de manera que se logre llegar a una buena distribución o *“layout”* de los departamentos dentro de la estructura. Este proyecto se enmarca dentro del campo de estudio de la distribución de centros productivos, el cual pertenece al diseño de sistemas productivos y a la investigación de operaciones.

El objetivo de éste proyecto es encontrar una “buena” solución que logre minimizar el costo de manejo de material mediante una distribución de departamentos que permita además minimizar el recorrido de las unidades productivas a través de los niveles de la estructura. Se programará un algoritmo que tomará en cuenta las

restricciones expuestas anteriormente con el fin de encontrar una solución que sea lo más cercana posible a la solución óptima.

1.7 OBJETIVOS

1.7.1 Objetivo general. Plantear una alternativa de solución al problema de distribución de planta (FLP) de múltiples niveles, un elevador y departamentos desiguales, a través de métodos metaheurísticos.

1.7.2 Objetivos específicos

- Realizar una revisión bibliográfica del problema de distribución de planta (FLP) de múltiples niveles, un elevador y departamentos de áreas desiguales, incluyendo planteamientos y enfoques aplicados al problema.
- Desarrollar un algoritmo para el problema de distribución de planta (FLP) de múltiples niveles, un elevador y departamentos de áreas desiguales.
- Programar el algoritmo en un software computacional y realizar los ajustes correspondientes al diseño del programa.
- Evaluar la efectividad del algoritmo, mediante la solución de problemas encontrados en la literatura.

1.8 METODOLOGÍA

1.8.1 Revisión bibliográfica. En primera instancia para desarrollar éste proyecto se empezó por realizar una extensa investigación en el campo del problema de distribución de planta y sus diferentes variaciones con el objetivo de que dicha revisión ofreciera un entendimiento mucho más profundo del problema a tratar y las diferentes consideraciones que éste conlleva. Se inició por investigar lo concerniente al origen del problema de distribución de planta FLP, sus primeros planteamientos y las herramientas utilizadas por los autores para darle solución a dichas formulaciones.

Seguidamente se efectuó una revisión literaria específica del problema de distribución de planta de más de un nivel, lo cual brinda un mayor entendimiento respecto a las complicaciones que dicha variación del problema FLP trae, como lo son las consideraciones respecto al transporte vertical de los productos y los métodos metaheurísticos que existen para solucionar dichos problemas. Como resultado de dicha investigación se obtiene claridad respecto al origen del problema, su justificación y su planteamiento.

A continuación se procede a investigar a fondo los diferentes tópicos esenciales para la comprensión del problema FLP Multi-floor como tal, dentro de dichos temas se encuentran la optimización combinatoria, los diferentes tipos de complejidad computacional y los diferentes métodos de solución tanto exactos como heurísticos y metaheurísticos.

1.8.2 Construcción del modelo. Luego de realizar la revisión literaria correspondiente al problema FLP se procede a construir el modelo matemático a partir de las características definidas a través de la investigación de literatura. Dichas características son las variables y parámetros utilizados para definir tanto el número como las áreas de los departamentos a ubicar, el espacio disponible para la organización de los mismos, la fijación de las coordenadas del elevador, y las demás restricciones que inciden en la función objetivo.

1.8.3 Programación y validación del algoritmo. Para realizar la programación del algoritmo se llevaron a cabo dos etapas importantes, en la primera se seleccionó el método de solución a emplear para el modelo, el cual fue el recocido simulado o SA por sus siglas en inglés y se prosiguió a construir el código para dicho método; con el objetivo de cimentar las bases estructurales del mismo para posteriormente adaptarlas a las necesidades y limitaciones del modelo matemático del problema FLP. En ésta etapa se puso a prueba el método al evaluar su rendimiento sobre funciones de prueba encontradas en la literatura tales como la función de Bohachevsky².

Luego de comprobar que el método es efectivo mediante las pruebas realizadas se adaptó el mismo al modelo matemático definido.

² Derek Bingham. Virtual Library of Simulation Experiments: Test Functions and Databases. Bohachevsky Functions. Simon Fraser University. 2013. Recuperado de <https://www.sfu.ca>.

2. MARCO TEÓRICO

2.1 FACILITY LAYOUT PROBLEM

Determinar la organización física de los elementos que componen un sistema de producción se define como el problema de distribución de planta (Facility Layout Problem). Dónde localizar las instalaciones y el eficiente diseño de las mismas son decisiones estratégicas sumamente importantes a las que cualquier organización se debe enfrentar (Singh & Sharma, 2005)³.

Este tipo de problema emerge en diferentes tipos de organizaciones, desde esquematizar la distribución de un hospital, organizar los salones de una institución educativa, localizar las distintas salas de espera dentro de un aeropuerto, entre otras. Su enfoque principal es encontrar la configuración más eficiente de un número n de departamentos indivisibles caracterizados por tener áreas irregulares lo cual dificulta considerablemente el planteamiento del problema.

La distribución resultante del problema FLP minimiza el costo total del manejo de material dentro de la instalación sujeto a dos restricciones principales: 1. El número de departamentos a distribuir y el área dispuesta para dicha distribución y 2. Las limitaciones locativas de los departamentos (no se permite la superposición de departamentos, todos deben ser ubicados dentro de la instalación, algunos departamentos debe ser asignados a cierto espacio específico o pueden no poder ser localizados en regiones específicas del plano de la instalación).

Al permitir la minimización de dichos costos, se evidencia la importancia estratégica que tiene lograr distribución eficiente de los departamentos que constituyen la ruta

³ Surya P. Singh & Renduchintala Sharma. A review of different approaches to the facility layout problems. The International Journal of Advanced Manufacturing Technology. 5-6. p 425-433. 2005

productiva de la organización ya que “una buena distribución de los departamentos puede contribuir en la eficiencia general de las operaciones, reducir el costo de manufactura, disminuir los “lead times” e incrementar la productividad de la empresa” (Misola & Navarro, 2013)⁴.

Generalmente el problema FLP se ha formulado como un problema de asignación cuadrática (QAP por sus siglas en inglés) clasificado dentro de la categoría NP-Completo, tal y como fue presentado por Koopmans y Beckman en 1957⁵. Debido a las limitaciones de los equipos computacionales de la época, resolver un problema de tal escala se dificultaba. El objetivo de dicho modelo podía variar dependiendo del enfoque deseado ya que las variables consideradas en su formulación abarcan los criterios de costo, distancia recorrida de un departamento a otro y el flujo de material que pasa por estos. Por lo cual su modelo se muestra de la siguiente manera:

$$\text{Min } f = \frac{1}{2} (\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n F_{ik} * D_{jl} * X_{ij} * X_{kl}) \quad (1)$$

Sujeto a:

$$\sum_{j=1}^n X_{ij} = 1 \quad \text{Para todo } i = 1, 2, 3, \dots, n \quad (2)$$

$$\sum_{j=1}^n X_{ij} = 1 \quad \text{Para todo } j = 1, 2, 3, \dots, n \quad (3)$$

Donde:

$X_{ij} = 1$ Si el departamento “ i ” es asignada a la ubicación “ j ”.

⁴ Maricar G. Misola & Bryan B. Navarro. Optimal Facility Layout Problem Solution Using Genetic Algorithm. International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, 7, 616-621. 2015, mayo, De World Academy of Science, Engineering and Technology Base de datos. 2013

⁵ Tjalling C. Koopmans & Martin Beckman. Assignment problems and the location of economic activities. Econometrica 25:53–76. 1957

$X_{ij} = 0$ Si el departamento "i" no es asignada a la región "j".

F_{ik} Es el flujo material entre dos departamentos i y k .

D_{jl} Es la distancia entre dos regiones j y l .

La restricción (1) posibilita que solo un departamento sea asignado a una región, la restricción (2) asegura que a cada región se le asigne solo un departamento. El objetivo es minimizar la distancia total recorrida por el material entre los departamentos $i=1$ hasta n y $k=1$ hasta n .

Dentro de las técnicas de solución del problema FLP formulado como un problema de asignación cuadrática (QAP) se destacan dos metodologías principalmente: Modelo de Teoría de Grafos y Modelo de Programación Entera Mixta (MIP).

Estas formulaciones se hicieron obsoletas a medida que la complejidad de los problemas aumentó y se incluyeron características que asemejaron dichos modelos a condiciones encontradas en las instalaciones del mundo real.

Dentro de estas características están la consideración de pasillos en la distribución final de la planta para el movimiento de los operarios, la presencia de estructuras inamovibles tales como columnas y paredes internas, que limitaban la libre distribución de los centros de trabajo, etc.

Con la adición de características a los modelos, las herramientas de computo empleadas para su solución debieron adaptarse a la complejidad de los problemas. Dicha adaptación dio como fruto la creación de los tipos de complejidad computacional, en los cuales se clasifica un problema de optimización combinatoria según el tiempo que se requiere para darle solución.

2.2 OPTIMIZACIÓN COMBINATORIA

La optimización combinatoria se concibe como la rama de la matemática aplicada y la ciencia de la computación que se enfoca en el modelado y solución de problemas en los cuales se tiene un objetivo de optimización sobre una función multivariable a través de métodos algorítmicos. Las variables definidas dentro de la función a optimizar deben estar definidas sobre un conjunto discreto de la siguiente manera:

Un problema de optimización combinatoria $P(S, f)$ se caracteriza por:

- Un conjunto discreto de variables $X = \{x_1, \dots, x_n\}$;
- Con dominios D_1, \dots, D_n ;
- Restricciones entre las variables
- Función objetivo f a minimizar, donde $f: D_1x_1, \dots, D_nx_n \rightarrow R^+$;

El conjunto de todas las posibles asignaciones factibles entre las variables se denota por:

$$S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} | v_i \in D_i, s \text{ satisface todas las restricciones}\}$$

Donde S se denomina espacio de búsqueda, en el cual cada elemento individual del conjunto puede ser considerado apto para generar una solución. Para resolver un problema de optimización combinatoria primero se debe encontrar una solución óptima global que arroje el mínimo valor de la función objetivo, esta se denota $s^* \in S$ y se caracteriza por $f(s^*) \leq f(s) \forall s \in S$.

Siendo s^* una solución global del sistema (S, f) se puede extrapolar un conjunto de soluciones óptimas globales $S^* \subseteq S$.

2.2.1 Tipos de complejidad computacional. Una de las maneras de clasificar los problemas de optimización combinatoria es tomando como criterio base su grado de dificultad a la hora de recibir un tratamiento computacional para generar una solución. Este juicio es también llamado complejidad computacional.

La teoría de la complejidad toma como objeto de estudio los recursos tiempo y espacio para estudiar su incidencia en la resolución algorítmica de un problema en función de la cantidad de operaciones efectuadas en el cálculo y el tamaño de los datos de entrada requeridos. La máquina de Turing es comúnmente utilizada como modelo matemático de un algoritmo además de la herramienta para clasificarlos al determinar la complejidad computacional midiendo el tiempo requerido por la máquina para procesar el algoritmo (Hartmanis & Stearns, 1965)⁶. La taxonomía de la complejidad computacional está dada por:

- **Clase P:** Se considera un subconjunto de los problemas de clase NP cuya característica principal es que pueden ser resueltos a través de un algoritmo en una máquina de Turing determinística en un tiempo polinomial.
- **Clase NP:** Se le denomina de ésta manera a los problemas que pueden ser resueltos por un algoritmo en un tiempo polinomial mediante una máquina de Turing no determinística. Para éste tipo de problemas existe siempre un algoritmo capaz de verificar la validez de la solución generada. El tiempo de cómputo requerido para dar solución a éste tipo de problemas se ve incrementado de manera exponencial en función del tamaño del problema. Su tamaño no permite que su tiempo de cómputo sea polinomial.
- **Clase NP-Completos:** Se define un problema P^* como un problema NP. Entonces P^* es NP-Completo si su tiempo de cómputo es reducible a un tiempo polinomial.

⁶ Juris Hartmanis & Richard Stearns. On the computational complexity of algorithms. Transactions of the American Mathematical Society. Vol 117. Pp 285-306. American Mathematical Society. 1965

- **Clase NP-Hard:** Esta clasificación es un subconjunto de los problemas clase NP para los cuales es imposible obtener una solución dentro de un tiempo polinomial para todas sus instancias.

Figura 1. Dominios de las clases de complejidad computacional



Fuente: Duarte, Abraham (2008)

2.3. VARIACIONES DEL PROBLEMA DE DISTRIBUCIÓN DE PLANTA

A través del tiempo el estudio del problema de distribución de planta se ha visto enriquecido enormemente gracias al surgimiento de nuevos planteamientos y consideraciones que han modificado las condiciones y reglas de juego que regían las formulaciones que inicialmente se habían postulado por parte de la comunidad académica.

La manipulación de las variables analizadas en el planteamiento de la función objetivo y el objetivo asociado a la misma abren una plétora de posibles formulaciones que responden a la distribución óptima aplicable dentro de una instalación bajo el criterio o enfoque dispuesto desde la función objetivo. Las variaciones más destacadas del problema de distribución de planta FLP son las mencionadas a continuación.

2.3.1 FLP considerando la variedad y volumen de los productos. La distribución de planta que debe procurar una entidad es influida por el tipo de producto que realiza y por el volumen del mismo ya que estos parámetros restringen la ruta productiva o la dirección que un producto debe seguir dentro de la instalación al trasladarse de departamento en departamento. Dilworth (1996)⁷ distingue cuatro tipos de distribuciones a considerar por el efecto de las variables mencionadas: *Distribución con producto fijo*, *Distribución de proceso*, *Distribución de producto* y *Distribución de celda*.

La principal característica de la *Distribución con producto fijo* se basa en que los diferentes recursos productivos de la organización (maquinas, trabajadores, herramientas) gravitan en torno a un producto inmóvil, es decir, los elementos deben transportarse hacia el producto para trabajar sobre el mismo. Éste tipo de distribución es común en la producción de artículos de gran tamaño pero con un volumen de producción lento, por ejemplo la construcción de barcos y aviones. La *Distribución de proceso* se caracteriza por la agrupación de departamentos con funciones similares o complementarias (por ejemplo los departamentos de corte y de soldadura de una empresa metalmecánica). Este tipo de distribución es utilizada en una amplia gama de productos. Para productos con un alto volumen de fabricación y una poca variabilidad en su naturaleza se emplea la *Distribución de Producto*, la cual permite secuenciar los departamentos de manera lógica según su proceso de manufactura. La *Distribución de celda* se emplea cuando se desea reunir varias máquinas para procesar familias de productos o partes similares de los mismos, para lo cual se desea distribuir dichas máquinas dentro de la celda de trabajo.

⁷ James B. Dilworth. Operation management. McGraw Hill. 1996.

2.3.2 FLP considerando los puntos de *Pick Up* y *Drop Off* de los puestos de trabajo. En determinados casos se hace necesario ubicar los puntos en los cuales el material entra y sale de los departamentos, determinados puntos de *Pick Up* (*P*) o recogida y *Drop Off* (*D*) o punto de entrada de material. La identificación de dichos puntos hace que se eviten interrupciones en el flujo del material ya que la distribución generada al resolver el problema FLP hará que solo existan empalmes P/D haciendo más ágil el movimiento del material y por tanto disminuyendo costos (Kim & Kim, 2000)⁸.

2.3.3 FLP con retrocesos y saltos. Dentro de un proceso productivo, es común encontrar movimientos particulares adjudicados a los productos en fabricación como lo son los movimientos de retroceso, donde el producto se desplaza de un departamento a otro previamente recorrido. La suma de éste tipo de movimientos debe minimizarse. Zhou (1998)⁹ denominó a éste Problema de la formación de la línea de producción (PLFP por sus siglas en inglés) donde se busca encontrar el orden en el cual se deben disponer los departamentos para disminuir el valor de los movimientos que se contraponen en dirección a la del proceso o flujo global.

Los movimientos denominados “saltos” se identifican por la omisión de departamentos dentro del flujo del producto, los cuales afectan su tiempo de manufactura y por tanto deben ser minimizados. Hassan (1994)¹⁰ fue pionero en la formulación de éste tipo de problema de distribución de planta.

⁸ Jae-Gon Kim & Yeong-Dae Kim. Layout planning for facilities with fixed shapes and input output points. International Journal of Production Research. 38(18). pp 4635-4653. 2000

⁹ Jiande Zhou. Algorithmes et outils pour l'analyse des flux de production à l'aide du concept d'ordre. Ph.D. Disertación doctoral (en Francés). Universidad de Estrasburgo. 1998

¹⁰ Mohsen M d Hassan. Machine layout problem in modern manufacturing facilities. International Journal of Production Research, 32(11), 2559– 2584. 1994

2.3.4 FLP considerando las dimensiones y formas de los departamentos. La literatura reconoce dos tipos principales de formas empleadas en los problemas de distribución de planta: Regulares e Irregulares ¹¹. La representación de un departamento varía dependiendo de la definición de su longitud (L) y su ancho (A). En caso de que ambos valores sean iguales se considera que el departamento posee unas dimensiones fijas y se le denomina bloque rígido (Chwif et al., 1998)¹². Para los departamentos de formas irregulares se establece el valor del radio aparente ($a = L/A$), el cual utiliza un límite superior (a_s) y un límite inferior (a_i) de modo que $a_i > a > a_s$.

La irregularidad de las formas de los departamentos complica el proceso de organización de los mismos, ya que la formulación de algoritmos competentes en esta tarea supone un trabajo bastante arduo.

En la literatura los problemas de distribución de planta con departamentos de formas irregulares se dividen en dos clases según el tipo de plano en donde se hace la representación del *layout*¹³: Plano de red o cuadrícula y plano continuo.

El plano de red se caracteriza por estar dividido en una cuadrícula donde cada elemento individual representa una unidad de área, esta red permite una sencilla ubicación de los departamentos según sus dimensiones. El plano continuo no tiene guías que ayuden a la localización de los centros de trabajo, por lo cual hay una libre ubicación de los mismos. La mayoría de los algoritmos desarrollados para dar solución al problema FLP emplean el plano de red¹⁴.

¹¹ Amine Drira, Henri Pierreval, Sonia Hajri-Gabouj. Facility layout problems: A survey. Annual reviews in control. pp 255-267. 2007.

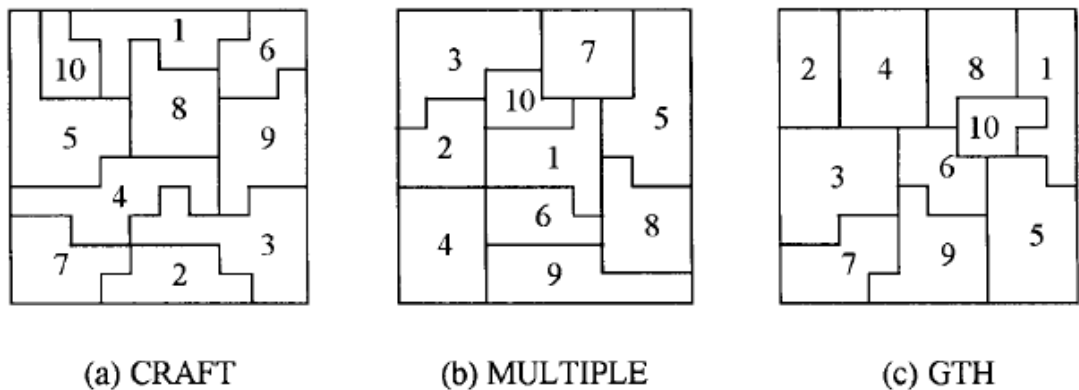
¹² Leonardo Chwif, Marcos R.Pereira Barretto, Lucas Antonio Moscato. A solution to the facility layout problem using simulated annealing. Computers in Industry, 36(1-2), 125-132. 1998

¹³ Geung-Chol Lee & Yeong-Dae Kim. Algorithms for adjusting shapes of departments in block layouts on the grid-based plane. Omega. 28. pp 111-122. 2000.

¹⁴ Geung-Chol Lee & Yeong-Dae Kim. Algorithms for adjusting shapes of departments in block layouts on the grid-based plane. Omega. 28. pp 111-122. 2000.

Algoritmos como CRAFT¹⁵, CLASS¹⁶, MULTIPLE¹⁷ y GTH¹⁸ emplean efectivamente la distribución de departamentos en planos de red más sin embargo existen ocasiones en las que, al llevar a cabo dicha tarea, la forma de los centros de trabajo se altera demasiado.

Figura 2. Distribuciones de planta según algoritmos CRAFT, MULTIPLE y GTH



Fuente: Autores: Geung-Chol Lee & Yeong-Dae Kim (2000)

Aún cuando el *layout* obtenido empleando dichos algoritmos logra una buena distribución de los departamentos, su aplicabilidad en la realidad se reduce gracias a la forma que toman los centros de trabajo. Para evitar que esto suceda se han empleado diferentes algoritmos que permiten la regularización de las formas de los departamentos sin deteriorar la calidad de la solución obtenida.

Existen tres enfoques destacados a la hora de regularizar las formas obtenidas en una distribución de planta. Estos son: Regularización basada en la distancia entre

¹⁵ Gordon C. Armour & Elwood S. Buffa. A heuristic algorithm and simulation approach to relative location of facilities. *Management Sciences*. 9(2). pp 294-309

¹⁶ Satish Jajodia, Ioannis Minis, George Harhalakis & Jean-Marie Proth. CLASS: Computerized layout solutions using simulated annealing. *International journal of production research*. Vol30(1). pp 95-108. 1992.

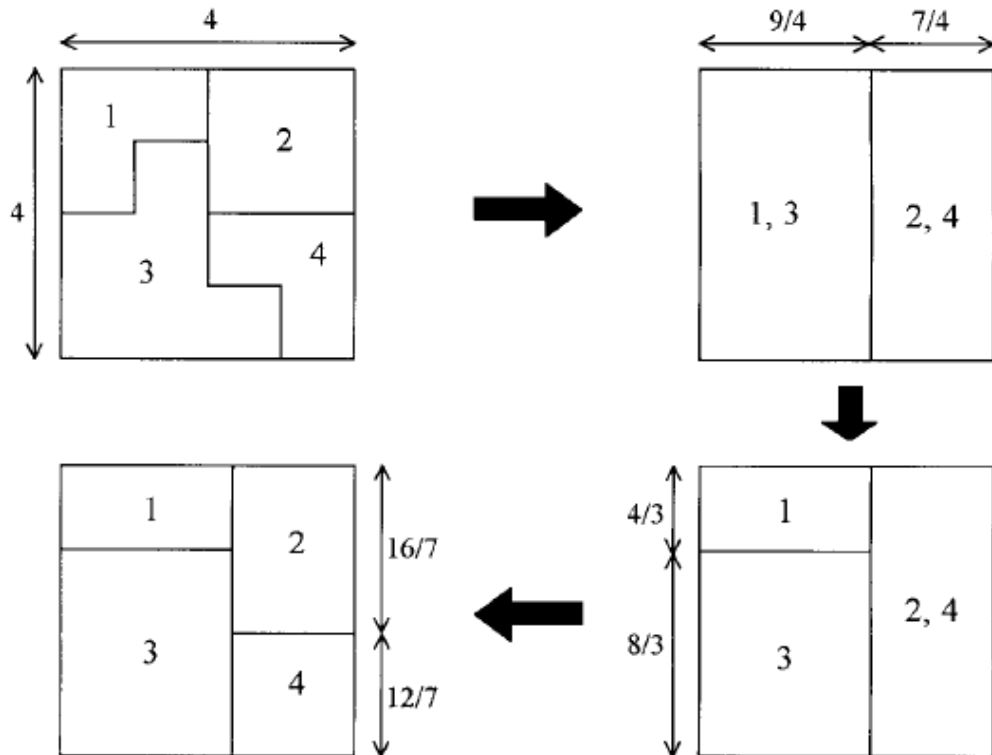
¹⁷ Yavuz A. Bozer, Russell D. Meller & Steven J. Erlebacher. An improvement-type layout algorithm for single and multiple floor facilities. *Management Sciences*. Vol 40(7). pp 918-932. 1994.

¹⁸ J-Y Kim, Y-D Kim, Graph theoretic heuristics for unequal-sized facility layout problems. *Omega*. Vol 218(2), p382-391. 1995.

centroides, regularización utilizando “*Beam Search*” y regularización usando líneas transversales.

Los enfoques anteriormente mencionados utilizan diferentes criterios para seleccionar los departamentos que se van a “cortar” o modificar para de esa manera obtener una distribución de planta con centros de trabajo de formas regulares.

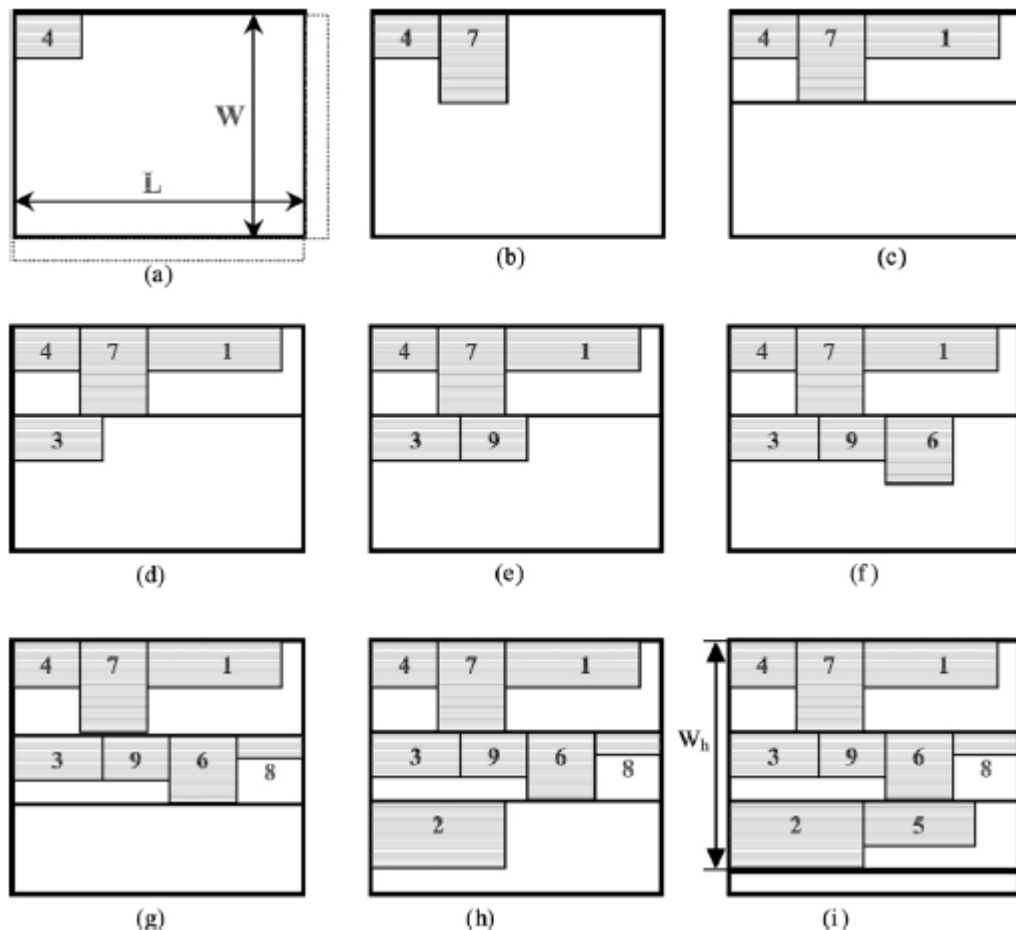
Figura 3. Ejemplo de la regularización de una distribución con departamentos irregulares



Fuente: Geun-Cheol Lee & Yeong-Dae Kim. 2000.

Lee & Lee (2002)¹⁹ proponen un algoritmo genético híbrido que permite dividir el espacio disponible dentro del piso para la organización de los departamentos. Estas divisiones se hacen en base a las medidas de los centros de trabajo. Seguidamente, basado en el orden determinado por el algoritmo genético, se ubican los departamentos de forma seguida en dirección de izquierda a derecha, de manera que el tamaño de las divisiones es dictado por el lado del centro de trabajo más amplio.

Figura 4. Distribución de planta según algoritmo híbrido de Lee & Lee



Fuente: Young Hae Lee & Moon Hwan Lee (2002)

¹⁹ Young Hae Lee & Moon Hwan Lee. A shape-based block layout approach to the facility layout problems using a hybrid genetic algorithm. *Computers & Industrial Engineering*, 42, pp 237-248, 2002.

En este proyecto no se emplearan algoritmos que regularicen las formas obtenidas de los departamentos, ya que dichas representaciones son obtenidas mediante la técnica de curva de llenado.

2.3.4.1 Curvas de Llenado: Empleadas por primera vez por los autores Bartholdi y Platzman²⁰ al resolver el problema del vendedor viajero a través de métodos heurísticos, para lo cual requiere encontrar la menor distancia entre una serie de nodos. Dentro del campo de la distribución de planta las curvas de llenado han sido utilizadas por Bozer et al²¹.

Las curvas de llenado se forman conectando una serie de curvas de Hilbert, las cuales son curvas fractales continuas que recubren el espacio de un plano. Fueron planteadas en primera instancia por el matemático alemán David Hilbert en 1891 y se emplean en numerosas aplicaciones en el campo de la informática debido a que si (x,y) son las coordenadas de un punto P dentro de un cuadrado unitario y d la distancia a lo largo de la curva cuando se alcanza el punto P, entonces todos los puntos que tienen distancias cercanas a d también poseen valores cercanos a los de (x,y) . Esta característica hace que las curvas de Hilbert se empleen mucho en la ubicación de direcciones IP.

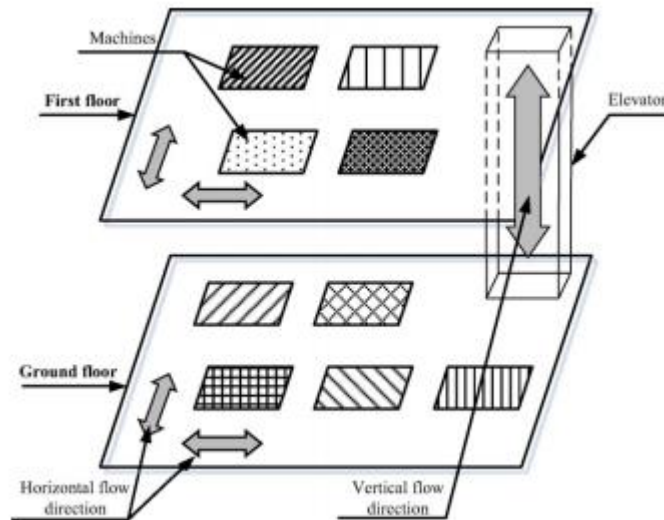
Cuando el plano en el que se va a aplicar la curva de llenado tiene irregularidades como departamentos fijos o áreas bloqueadas, la curva de llenado puede ser programada libremente aunque no sea propiamente una curva de Hilbert.

A la hora de aplicar una curva de llenado a la superficie de un plano de red se debe asegurar que esta atraviese todos los recuadros del plano sin que la misma curva pase por un recuadro más de una sola vez, evitando redundancias.

²⁰ John J. Bartholdi & Lauren K. Platzman. An $O(N \log N)$ planar traveling salesman heuristic based on spacefilling curves. Operations research letters. Vol 1(4). Pp121-125. 1982.

²¹ Yavuz A. Bozer, Russell D. Meller & Steven J. Erlebacher, An Improvement-Type Layout Algorithm for Single and Multiple Floor Facilities, Management Science, 40(7), 918-932. 1994.

Figura 6. Distribución de planta de más de un nivel



Fuente: Drira, A., Pierreval, H., Hajri-Gabouj, S..

En la figura se muestra la posibilidad de movimiento de los materiales o productos por la instalación tanto en su dimensión horizontal al pasar por los diferentes departamentos y centros de trabajo del nivel en el que se encuentra, como en su dimensión vertical haciendo uso de un elevador como medio de traslado de los productos hacia los niveles adyacentes.

En estos casos se debe determinar tanto la posición del centro de trabajo dentro del área plana de la estructura como el nivel en el cual dicho centro de trabajo debe ser ubicado para lograr la distribución que satisfaga las restricciones de espacio y flujo de material además de la función objetivo.

Los problemas FLP Multi-floor generalmente requieren de la especificación de la ubicación del elevador que posibilitará el transporte de productos de un nivel a otro además del costo de trasladar una unidad de carga o producto tanto su dirección horizontal como vertical. Para ello en la literatura se plantean las variables

asociadas al costo referente a los movimientos de los productos de la siguiente manera:

- C_{ij}^V Es el costo del transporte de material desde el departamento i al j en dirección vertical.
- C_{ij}^H Es el costo del transporte de material desde el departamento i al j en dirección horizontal.

La función objetivo asociada a éste tipo de problemas se presenta a continuación:

$$\min Ct = \sum_{i=1}^N \sum_{j=1}^N (C_{ij}^H * d_{ij}^H + C_{ij}^V * d_{ij}^V) * f_{ij}$$

Donde d_{ij}^H y d_{ij}^V representan las distancias entre los departamentos i y j en sus direcciones horizontales y verticales correspondientemente. De igual manera se define f_{ij} como el flujo existente entre el departamento i y el departamento j . Las distancias horizontales entre departamentos generalmente son calculadas a partir de los centroides de éstos de manera que se minimice la magnitud horizontal del recorrido del producto hacia el elevador y desde el mismo hacia los centros de trabajo (Meller, R. D. & Gau, K., 1996)²³.

Existe también dentro de las formulaciones del problema de distribución de planta FLP la posibilidad de plantear funciones multi-objetivos, las cuales generalmente se enfocan en abarcar un espectro amplio de variables que se ven afectadas por la distribución de la instalación además del costo del manejo de materiales que se contemplaba en el modelo clásico.

²³ Russell D. Meller, Kai-Yin Gau. The Facility Layout Problem: Recent and Emerging Trends and Perspectives. Journal of Manufacturing Systems, 15, 351-366. 2015, Junio. 1996

La representación más común de una función multi-objetivo para un problema FLP integra en adición a las consideraciones de costo de transporte de material dentro de la instalación, la minimización del espacio en desuso entre los departamentos, denominado la tasa de cercanía (Ramtin, F. et al, 2010)²⁴.

La función objetivo de éste planteamiento se muestra a continuación:

$$\min \alpha \sum_{j=1}^N \sum_{i=1}^N (f_{ij} c_{ij}) * d_{ij} - (1 - \alpha) \sum_{j=1}^N \sum_{i=1}^N r_{ij} * x_{ij}$$

Donde:

- ❖ α Es el coeficiente de ponderación de las funciones objetivo.
- ❖ f_{ij} Representa el flujo de material entre los departamentos i y j .
- ❖ c_{ij} Es el costo de transportar una unidad de material entre los departamentos i y j .
- ❖ r_{ij} Es la tasa de cercanía entre los departamentos i y j .
- ❖ $x_{ij} = \begin{cases} 1, & \text{Si los departamentos } i \text{ y } j \text{ tienen un límite en común} \\ 0, & \text{En caso contrario} \end{cases}$

La importancia de fijar correctamente el valor del parámetro α para el modelo ha sido estudiado por Meller y Gau (1997)²⁵.

²⁴ Faraz Ramtin, M. Abolhasanpour, Hossein Hojabri, Ahmad Hemmati. & Ayyoub A. Jaafari. Optimal Multi Floor Facility Layout. International MultiConference of Engineers and Computer Scientists. Pages 1797-1800. Marzo 2010

²⁵ Russel D.Meller, Kai Yin Gau, The Facility Layout Problem: Recent and Emerging Trends and Perspectives, Journal of Manufacturing Systems, Vol. 15, 1997

Dentro del estudio de las variaciones del problema FLP Multi-floor también se han considerado el efecto de las diferentes restricciones que el movimiento vertical de los productos puede tener según la capacidad instalada de los elevadores, la cantidad de estos y su localización.

Este proyecto se enfoca en una variación del problema de distribución de planta multinivel (FLP Multi-floor) en la cual los departamentos o centros de trabajo poseen una forma regular y serán distribuidos dentro de una instalación de dos niveles la cual posee un ascensor que abastece ambos niveles, cuya localización es fija. El objetivo de la formulación es la minimización del costo de transporte de material entre departamentos²⁶.

2.4. MÉTODOS DE SOLUCIÓN PARA EL PROBLEMA FLP MULTI-FLOOR

Debido a la complejidad asociada con los problemas FLP de más de un nivel, se formulan métodos que generan buenas soluciones dentro de las limitaciones que la información suministrada a dichos métodos permita. El objetivo de la aplicación de estos métodos es la búsqueda de soluciones de alta calidad que satisfagan las restricciones a las que el problema está sujeto. Se han propuesto varios tipos de enfoques de optimización en la literatura. Algunos de ellos son: métodos exactos, como *branch & bound* que generan soluciones óptimas para problemas pequeños, y los enfoques de aproximación, como heurísticas y metaheurísticas²⁷. La diferencia en su aplicación radica en la complejidad del problema, ya que los algoritmos

²⁶ Matsuzaki, Kenichiro; Irohara, Takashi; Yoshimoto, Kazuho. Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization. *Computers & Industrial Engineering*, 36(2), 487-502. 1999.

²⁷ Drira, Amine; Pierrevala, Henri; Hajri-Gaboujb, Sonia. Facility layout problems: A survey. *Annual Reviews in Control*, 31, 255-267. Julio 2015.

exactos no se pueden aplicar para resolver problemas de gran tamaño de optimización combinatoria en un tiempo razonable.²⁸

2.4.1 Métodos exactos. Para abordar los problemas de FLP multi-floor se plantean tres estrategias principales: la metodología *Branch & Bound*, la cual genera una solución óptima al problema planteado, y los modelos de asignación cuadrática (QAP) como programación entera mixta (MIP) y la teoría de grafos.

2.4.1.1 Branch & Bound: Los métodos exactos aplicados como técnicas de solución para los problemas de FLP están basados en los enfoques brindados por el algoritmo de *Branch & Bound* o de Ramificación y Poda. Este algoritmo se utiliza para hallar una solución óptima en problemas de asignación cuadrática (QAP) ya que éste sólo involucra variables binarias.²⁹

Este método está basado en un árbol de búsquedas, donde en cada etapa todas las posibles soluciones del problema son particionadas en dos subconjuntos, cada una representada por nodos en un árbol de decisiones.

Esta forma de ramificar y acotar la solución permite descartar un buen número de subconjuntos de solución, esto evita realizar una búsqueda infructuosa. Dentro del ámbito del problema de distribución de planta, Simmons (1969)³⁰ fue de los primeros en proponer un algoritmo branch and bound para obtener una distribución óptima. Su planteamiento se basaba en la creación de numerosas distribuciones “posibles” para luego buscar la distribución óptima y finalizar el algoritmo.

28 Maghsud Solimanpura, Amir Jafarib. Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers & Industrial Engineering*, 55, 606-619. 2008

29 Meller, Russell D.; Gau, Kai-Yin. The Facility Layout Problem: Recent and Emerging Trends and Perspectives. *Journal of Manufacturing Systems*, 15, 351-366. 2015, Junio. 1996

30 Donald Simmons. One-dimensional space allocation: An ordering algorithm. *Operations Research*. Vol 17. 812-826. 1969

Love y Wong (1976)³¹ formularon el problema FLP mediante programación entera mixta lineal donde M es un número alto seleccionado arbitrariamente y U es la suma de las longitudes de todos los departamentos. Se denota u_i como el punto del departamento i más alejado del origen dentro del intervalo $[0, U]$. Defina $\alpha_{ij} = 1$ cuando el departamento i está localizado a la izquierda del departamento j y $\alpha_{ij} = 0$ en el caso contrario. R_{ij} es la distancia entre los centroides de los departamentos i y j sólo si el departamento i se encuentra a la derecha del departamento j , y $R_{ij} = 0$ en caso contrario. L_{ij} Es la distancia entre los centroides de los departamentos i y j sólo si el departamento i está a la izquierda del departamento j , en caso de no ser así $L_{ij} = 0$. El modelo propuesto por los autores es,

$$\min \left\{ \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} (R_{ij} + L_{ij}) \right\}$$

Sujeto a:

$$R_{ij} - L_{ij} = u_i - u_j + \frac{1}{2}(h_j - h_i)$$

$$u_i - u_j + M(\alpha_{ij}) \geq h_i$$

$$u_j - u_i + M(1 - \alpha_{ij}) \geq h_j$$

$$h_i \leq x_i \leq N$$

La primera restricción convierte la distancia entre los puntos u_i y u_j en la distancia entre los dos centroides de los departamentos i y j (sean estos R_{ij} o L_{ij}). La segunda y tercera restricción aseguran que se respete la distancia entre departamentos. Finalmente la cuarta restricción asegura que cualquier departamento sea ubicado en el intervalo $[0, N]$.

³¹ Robert Love & Jsun Y. Wong. On solving a one-dimensional space allocation problem with integer programming. INFOR. Vol 4(2). p139-143. 1976

2.4.1.2 Modelo de teoría de grafos: La teoría de grafos, como rama de la topología, es el estudio de estructuras matemáticas utilizadas para modelar las relaciones entre objetos dentro de una colección mediante conexiones representadas por vértices y uniones. Dicha disciplina matemática es de profundo interés dentro de las ciencias de la computación. Su precursor fue Leonard Euler en 1736³².

Los grafos son frecuentemente utilizados para modelar relaciones binarias entre objetos de un mismo dominio, por ejemplo, computadores dentro de una red. (Godsil & Royle, 2001)³³

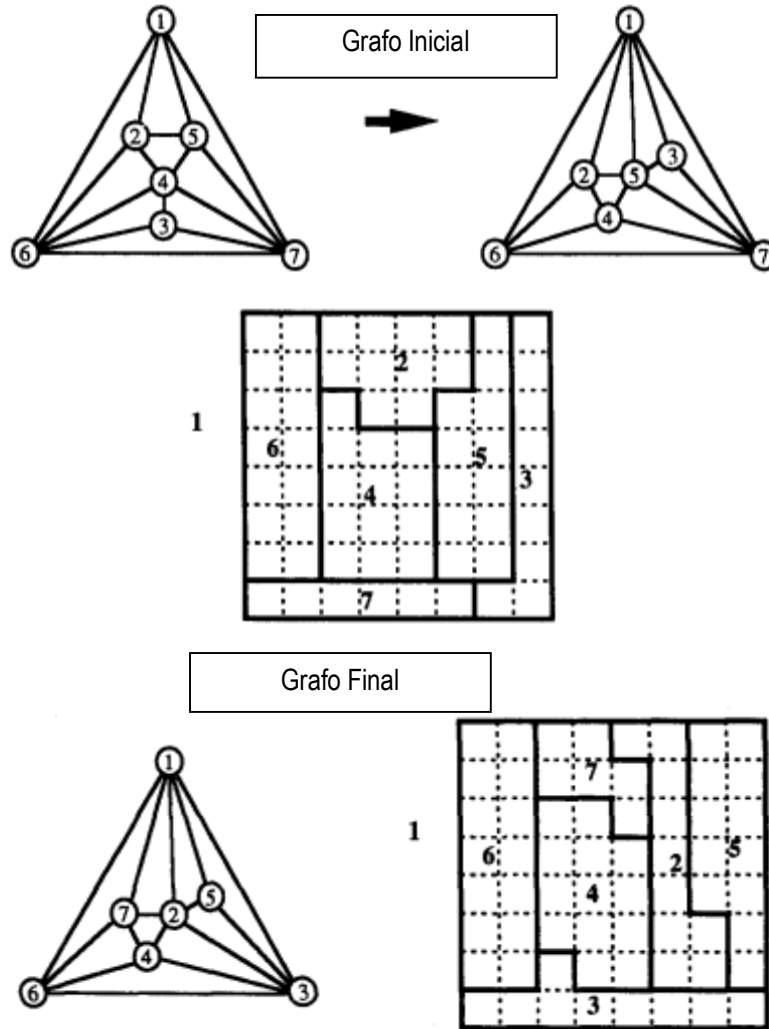
Para éste modelo de solución al problema de FLP cada departamento o máquina, sin tener en cuenta su demanda de espacio o naturaleza geométrica, es representada como un nodo dentro de una red gráfica de relaciones entre elementos denominada *grafo*³⁴.

³² Biggs, Norman; Lloyd, Keith; Wilson, Robin. Graph Theory 1736-1936. Oxford University Press. 1986

³³ Chris Godsil, C. & Gordon Royle. Algebraic Graph Theory. New York: Springer-Verlag. 2001

³⁴ Surya P.N. Singh & Renduchintala R. K. Sharma. A review of different approaches to the facility layout problems. The International Journey of Advanced Manufacturing Technology. 5-6. p 425-433. 2005

Figura 7. Ejemplo construcción de un grafo para problemas de Distribución de Planta.



Fuente: Kim. J, Kim. Y³⁵. Modificado por Autores

³⁵ J-Y Kim, Y-D Kim, Graph theoretic heuristics for unequal-sized facility layout problems. Omega. Vol 218(2), p382-391. 1995.

La ubicación de dichos nodos depende de la deseada cercanía que un departamento debe tener con otro. Dicho de otra manera, los modelos de teoría de grafos consideran la cercanía esperada entre un par de departamentos o centros de trabajo para luego ubicarlos dentro del espacio dispuesto en la instalación.³⁶

Es común que los planteamientos del problema FLP que emplean la teoría de grafos utilicen una metodología de construcción, es decir, la solución de un problema es robustecida mediante un proceso iterativo el cual parte de una distribución base hasta que se obtiene una solución óptima.

Dichos procedimientos tienen varias limitaciones, las cuales impactan directamente en su utilidad o capacidad de resolver los problemas planteados; en particular su funcionamiento es altamente dependiente de la distribución inicial otorgada. Desafortunadamente debido a la naturaleza combinatoria de los grafos, la distribución ideal de partida para un problema puede no ser apropiada para otro.³⁷

2.4.1.3 Modelo QAP: Koopmans y Beckman en 1957³⁸ fueron los primeros en formular un problema FLP mediante el modelo QAP. A pesar de considerar las variables de costo, distancia entre departamentos y flujo de material entre los mismos la limitación más grande a la que se enfrentaron estos autores fue la precariedad de los equipos computacionales de su época, los cuales sólo permitían resolver problemas pequeños (pocos departamentos a organizar) de manera óptima.

El modelo QAP es particularmente famoso para dar solución a los problemas de distribución de planta discretos (de instancias pequeñas). Estos se caracterizan por

³⁶ Mohsen Hassan, Gary L. Hogg, A review of graph theory applications to the facilities layout problem. *Omega* 15:291–300. 1987

³⁷ Latif A. Al-Hakim. Two graph-theoretic procedures for an improved solution for the facilities layout problem. *International journal of production research*. Vol 29, N°8. Pp 1701-1718. 1991.

³⁸ Tjalling C. Koopmans & Martin Beckman. Assignment problems and the location of economic activities. *Econometrica* 25:53–76. 1957

dividir en partes iguales el espacio disponible para la distribución de departamentos.³⁹

2.4.1.4 Modelo de programación entera mixta (MIP): Es una representación del problema de FLP y por esto ha recibido mucha atención como una alternativa de solución al mismo al considerar la distancia rectilínea entre dos departamentos para minimizar el tiempo de flujo de material entre los mismos y así obtener una distribución de planta óptima bajo ese criterio. Montreuil (1990)⁴⁰ fue pionero en formular el problema de FLP bajo un marco de programación entera mixta al minimizar la distancia recorrida por el producto dentro de una instalación continua. Este modelo se derivó de la discretización del QAP.

Las metodologías de solución exacta del FLP basadas en MIP se caracterizan por su limitación en cuanto al tamaño del problema que son capaces de afrontar, es decir, los métodos derivados de la programación entera mixta no pueden resolver problemas a gran escala o con gran número de departamentos (típicamente más de 9 departamentos) a menos que se asuman supuestos como departamentos con formas y orientaciones definidas que son difíciles de justificar en la realidad.⁴¹

³⁹ Ming-Jaan Wang, Michael H. Hu, Meei-Yuh Ku. A solution to the unequal area facilities layout problem by genetic algorithm. Computers in industry. Vol 56. Pp 207-220. 2005

⁴⁰ Montreuil, Benoit. modeling framework for integrating layout design and flow network design. In: Proceedings of the material handling research colloquium, Hebron, KY, pp 43–58 1990

⁴¹ Abdullah Konak, Sadan Kulturel-Konak, Bryan Norman, Alice E. Smith. A new mixed integer programming formulation for facility layout design using flexible bays. Operations Research Letters. Vol 34(6). Pp 660-672. 2005

2.4.2 Heurísticas. Cuando se emplea un enfoque metodológico práctico a la resolución de un problema aun cuando éste procedimiento no garantiza el retorno de una solución óptima o perfecta al mismo, pero se considera suficiente para el cumplimiento de los objetivos que se planteen, se está utilizando una técnica heurística. Los métodos heurísticos son empleados en numerosos campos de estudio como la ciencia computacional y la inteligencia artificial para resolver ágilmente problemas de gran tamaño y complejidad.

Los algoritmos heurísticos pueden clasificarse en dos grandes grupos: Algoritmos de *construcción*, donde se genera una solución única y algoritmos de *mejoramiento*, donde se parte de una solución sub-óptima y a través de iteraciones ésta se va refinando hasta convertirse en una solución de alta calidad al ser la más cercana a la solución óptima.

Los algoritmos de construcción son considerados como los primeros métodos de solución concebidos para el problema de asignación cuadrática (QAP) desde un punto de vista conceptual, pero la calidad de las soluciones generadas por éste método generalmente no son satisfactorias. (Singh & Sharma, 2005)⁴²

Los métodos de mejoramiento dependen en la formulación de una solución posible al problema planteado y a través de iteraciones donde se modifican los valores de las variables individuales de manera sistemática hasta obtener una solución mejor que la inicialmente generada. La combinación de métodos de construcción con los métodos de mejoramiento es común en la solución de problemas complejos ya que hace más eficiente el proceso de iteración y la calidad de la solución resultante.

Tanto para el enfoque de construcción como de mejoramiento se aplican dos criterios base para el desarrollo del respectivo algoritmo heurístico de solución: el

⁴² Surya P.N. Singh & Renduchintala R. K. Sharma. A review of different approaches to the facility layout problems. The International Journey of Advanced Manufacturing Technology. 5-6. p 425-433. 2005

criterio de distancia y el criterio de adyacencia. La diferencia primordial entre ambos criterios reside en la función objetivo del problema.

Para el caso de un problema FLP Multi-Floor, dada la consideración de los movimientos verticales y horizontales que debe realizar el producto en su proceso, las consideraciones de costo y distancia deben obedecer a dichas posibilidades. Por tal motivo, la función objetivo a utilizar desde un enfoque heurístico sería la siguiente:

$$\min \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N (C_{ikH} * D_{jIH} + C_{ikV} * D_{jIV}) * X_{ij} * X_{kl}$$

Donde C_{ikH} y C_{ikV} representan los costos de manejo de material en las direcciones horizontal y vertical respectivamente, y D_{jIH} junto con D_{jIV} son las distancias horizontal y vertical entre departamentos respectivamente.

La consideración del transporte vertical dentro de las distribuciones multi-floor es de las más importantes a realizar. El costo de dicho transporte, asistido por un elevador, es significativamente mayor al costo del movimiento horizontal del material, lo cual otorga un peso considerable a la minimización del costo de movimiento vertical del material para lograr una formulación eficiente del problema asegurando una buena distribución como resultado del mismo.

Al pensar en el transporte vertical de material se pueden derivar una amplia gama de variantes adicionales para el problema, tales como el número de elevadores a

utilizar, la localización de dichos elevadores y la capacidad de cada uno de éstos; consideraciones que sin duda elevan la dificultad del problema.⁴³

2.4.3 Metaheurísticas. El principal problema que presentan los algoritmos heurísticos es su incapacidad para escapar de los óptimos locales y que, en muchos casos, la solución que proveen no es muy satisfactoria. De allí nacen los algoritmos metaheurísticos, los cuales son otros algoritmos de búsqueda más efectivos y eficientes que evitan en lo posible quedar atrapados en los óptimos locales. De hecho, el prefijo griego “meta” se usa para indicar que estos algoritmos son de un nivel más alto que los heurísticos.

Estos algoritmos están diseñados para aplicarse a un amplio rango de problemas de optimización para los cuales no existe un algoritmo de solución específico y satisfactorio. Las metaheurísticas son ampliamente usadas para solucionar problemas complejos en los sectores de industrias y servicios, y en áreas que van desde las finanzas hasta el manejo de la producción.

Algunas de las metaheurísticas más utilizadas para solucionar los problemas de FLP son: recocido simulado (SA), algoritmo genético (GA), optimización por enjambre de partículas (PSO), optimización por colonia de hormigas (ACO) y búsqueda tabú (TS), entre otras.⁴⁴

2.4.3.1 Búsqueda Tabú: La técnica de Búsqueda Tabú o TS (Por sus siglas en inglés, *Tabu Search*), es un algoritmo metaheurístico que se utiliza para resolver problemas de optimización combinatoria. Fue definida y formalizada por Glover en

⁴³ Kenichiro Matsuzaki, Takashi Irohara, Kazuho Yoshimoto. Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization. *Computers & Industrial Engineering*. Vol 36 (2). Pp 487-502. 1999.

⁴⁴ Mani S. Saravanan & Shailendra G. Kumar. Different approaches for the loop layout problems: a review. *Manufacturing Technology Journal*, 69, 2513–2529. 2013

1986. Es una técnica diseñada para gestionar un algoritmo de búsqueda local integrado. Utiliza de forma explícita la historia de la búsqueda para escapar de los mínimos locales e implementar una estrategia de exploración. Su principal característica se basa de hecho en el uso de mecanismos inspirados en la memoria humana.⁴⁵

El proceso desarrollado en una búsqueda tabú comienza con una solución inicial y luego inicia una búsqueda en la vecindad de la solución actual. Por ello, lo primero que se debe definir es una solución viable al problema y ciertos parámetros como el tamaño de la vecindad, de la lista tabú, los criterios de búsqueda y de parada.

Cuando se inicia la búsqueda se genera un conjunto de soluciones de la vecindad a través de un cambio predefinido a la solución actual. A continuación, la mejor solución se selecciona entre el conjunto actual de soluciones vecinas y esto se convierte en la nueva solución actual. Una vez más, se genera un nuevo conjunto de soluciones vecinas de la nueva solución actual y el proceso se repite hasta que se cumplan los criterios de parada. La lista tabú registra las últimas soluciones encontradas (o algunos atributos de ellas) y prohíbe que estas soluciones (o soluciones que contienen uno de estos atributos) sean visitadas otra vez, siempre y cuando se encuentren en la lista. De ésta manera se logra restringir algunos movimientos para prevenir que la búsqueda caiga en un ciclo y lograr que el proceso abandone los óptimos locales en busca del óptimo global. El tamaño o longitud de la lista tabú controla la memoria del proceso de búsqueda por medio de la *tabu tenure*, el cual es el tiempo o número de iteraciones que un elemento (movimiento o atributo) permanece en la lista tabú antes de poder ser considerado nuevamente. Este tiempo puede ser fijo o variable.

⁴⁵ Lou Y. Liang, & Wen C. Chao. The strategies of tabu search technique for facility layout optimization. *Automation in Construction*, 17, pp.657-669. 2008

Tal y como las costumbres sociales humanas pueden cambiar dependiendo del tiempo, las soluciones tabú pueden dejar de serlo; esto se logra mediante la base de una memoria cambiante donde se define una forma de “olvido estratégico” llamado criterio de aspiración, el cual permite que una solución o atributo pueda salir de la lista tabú antes de que se cumpla su plazo⁴⁶.

Si la longitud de la lista es baja, la búsqueda se concentra en pequeñas áreas del espacio de búsqueda. Por el contrario, una longitud alta fuerza a que el proceso de búsqueda explore regiones más grandes, ya que prohíbe volver a visitar un mayor número de soluciones.⁴⁷

2.4.3.2 Recocido Simulado: El recocido simulado o SA (Por sus siglas en inglés, *simulated annealing*) es un método de utilizado para resolver problemas de optimización combinatoria que fue propuesto inicialmente por Kirkpatrick. Su nombre proviene del proceso del recocido de los sólidos. Este proceso consiste en calentar un sólido hasta lograr su fundición, seguido de esto se disminuye lentamente su temperatura hasta que el sólido alcanza el estado de energía más bajo o estado fundamental. Si la temperatura inicial no es lo suficientemente alta o si la temperatura se disminuyó rápidamente, el sólido en el estado fundamental tendrá muchos defectos o imperfecciones.⁴⁸

El algoritmo de recocido simulado posee la capacidad de saltar o escapar de los óptimos locales formados dentro del espacio factible de soluciones del problema en búsqueda del óptimo global de la función objetivo. Esta habilidad se le atribuye a la definición de la cual establece la probabilidad con la cual el algoritmo aceptará

⁴⁶ Riojas Cañari, Alicia. Conceptos, algoritmo y aplicación al problema de las N-reinas. Trabajo de grado Licenciada en investigación operativa. Lima, Perú: Universidad Nacional Mayor de San Marcos. Facultad de ciencias matemáticas. Pregrado en licenciatura en investigación de operaciones y sistemas. 2005. p32-50.

⁴⁷ Bozorgi, Najmeh; Abedzadeh, Mostafa & Zeinali. Mohsen. Tabu search heuristic for efficiency of dynamic facility layout problem. *Manufacture Technology*, 77, pp.689-703. 2015

⁴⁸ McKendall, Alan R.; Shang, Jin & Kuppusamy, Saravanan. Simulated annealing heuristics for the dynamic facility layout problem. *Computers & Operations Research*, 33, pp.2431-2444. 2005

variaciones en el valor de la función objetivo que resulten en soluciones de menor calidad. Esto logra diversificar el proceso iterativo por cuanto hace un barrido completo del espacio factible de soluciones (soluciones de alta y baja calidad) para luego intensificar la búsqueda en las vecindades de las buenas soluciones encontradas; y así hallar la solución que más se aproxime al óptimo global. Para ello se establece la Temperatura como parámetro de control.

Los componentes más importantes de un algoritmo de recocido simulado son la probabilidad de aceptación y el programa de enfriamiento. La probabilidad de aceptación define qué tan probable es que el algoritmo admita una solución de baja calidad como la solución actual. Uno de los parámetros más importantes que se deben definir en el algoritmo SA, es el programa de enfriamiento o *cooling Schedule*. Este parámetro determina qué tan alta será la temperatura inicial y las reglas que la determinarán, así como también cuándo deberá bajar la temperatura, en qué cantidad, y cuándo deberá terminar el proceso de recocido.⁴⁹

2.4.3.3 Algoritmos genéticos: El Algoritmo Genético o GA (por sus siglas en inglés, *Genetic Algorithm*) son un mecanismo de búsqueda para solucionar problemas complejos. Fue introducido inicialmente por Holland en la década de 1960. El objetivo inicial de éste método era proporcionar soluciones a los problemas que surgieron en muchos sectores industriales y de servicios, los cuales resultaron extremadamente difíciles de resolver con los métodos que se conocían hasta ese momento. Es un algoritmo ampliamente utilizado en problemas de optimización y recientemente se ha usado como herramienta de la ingeniería industrial para resolver diferentes problemas de optimización y búsqueda.⁵⁰

⁴⁹ Chwif, Leonardo; Pereira Barretto, Marcos R.; Moscato, Lucas Antonio. A solution to the facility layout problem using simulated annealing. *Computers in Industry*, 36, pp.125-132. 1998

⁵⁰ Gómez, Alberto; Fernández, Isabel; De la Fuente, David & Puente García, Javier. Using genetic algorithms to resolve layout problems in facilities where there are aisles. *Int. J. Production Economics*, 84, pp.271-282. 2003

La metodología de éste algoritmo se caracteriza por utilizar un lenguaje de genética natural modelando, de forma artificial, mecanismos de la evolución natural aplicados a la optimización de problemas. En los algoritmos genéticos se asocia el concepto de *individuo* a una solución factible del problema y el de *población* a un conjunto de individuos (soluciones factibles). Los individuos están formados por *genes* (elementos ordenados en una sucesión lineal), que se evalúan a través de una función de aptitud denominada *fitness*, que corresponde a una medida de la calidad del individuo como solución del problema. La población inicial de soluciones se determina de manera aleatoria y el proceso de selección durante el proceso evolutivo se realiza de acuerdo con una distribución de probabilidades, que determina que un individuo tiene una probabilidad de ser seleccionado proporcional a su *fitness*.⁵¹ Es decir, en el proceso de selección los individuos de mejor *fitness* tienen mayor probabilidad de ser seleccionados.

La facilidad de formular un algoritmo híbrido de GA con otros algoritmos ha hecho de este método una herramienta muy usada para la optimización multi-objetivo en problemas de FLP. En general un Algoritmo Genético híbrido con métodos como recocido simulado o búsqueda tabú es elegido como el algoritmo para optimizar múltiples objetivos como el costo de manejo de materiales y la utilización del espacio, la ubicación de instalaciones en torno a una estructura de pasillo y la optimización de los costos de manejo de materiales, minimización de costos de manipulación y re-diseño de material.⁵²

2.4.3.4 Colonia de Hormigas: El algoritmo de Colonia de Hormigas o ACO (por sus siglas en inglés, *Ant Colony Optimization*) fue introducido primero por Dorigo y luego fue formalizado como una metaheurística por Dorigo & Di Caro. Es un enfoque que

⁵¹ Salazar-Horning, Eduardo & Medina, Juan Carlos. Minimización del makespan en máquinas paralelas idénticas con tiempos de preparación dependientes de la secuencia utilizando un algoritmo genético. *Ingeniería Investigación y Tecnología*, 14, pp.43-51. 2012

⁵² Kundu, Anirban & Kumar Dan, Pranab. The scope of genetic algorithms in dealing with facility layout problems. *South African Journal of Industrial Engineering*, 21, pp.39-49. 1998

se ha hecho muy popular en la literatura para resolver problemas de optimización combinatoria.⁵³

La idea fundamental de la heurística de hormigas se basa en el comportamiento natural de las hormigas, las cuales tienen éxito en la búsqueda de los caminos más cortos desde su nido a las fuentes de alimento mediante la comunicación a través de una memoria colectiva que se da por medio de senderos de feromonas. Es decir, Cada hormiga toma en consideración los senderos (elección probabilística) de feromonas que dejan todos los otros miembros de la colonia de hormigas que precedieron a su curso, esos rastros de feromonas son, en otras palabras, un olor dejado por cada hormiga en su camino. Esta feromona se evapora con el tiempo, y por lo tanto la elección probabilística para cada hormiga cambia con el tiempo. Después de varios trayectos de hormigas, la ruta de acceso a la comida tendrá un rastro de feromonas mucho más alto y por lo tanto todas las hormigas seguirán el mismo camino.

Para adaptar éste fenómeno colectivo a los problemas de optimización combinatoria analógicamente se debe tener en cuenta que: el espacio de búsqueda de la hormiga se convierte en el espacio de las soluciones del problema, la cantidad de comida que se halle en una fuente de alimento se convierte en la evaluación de la función objetivo para la solución correspondiente, y los senderos de feromonas serían la memoria adaptativa del problema.⁵⁴

El algoritmo de Colonia de Hormigas ha sido planteado para solucionar algunas variaciones del FLP como por ejemplo, el FLP dinámico (DLP). Baykasoglu, Dereli & Sabuncu (2004) formularon un algoritmo ACO para un DLP con restricciones de presupuesto basadas en una demanda inestable que se desarrolla bajo unas

⁵³ Uğur, Aybars; Aydin, Doğan. An interactive simulation and analysis software for solving TSP using Ant Colony Optimization algorithms. *Advances in Engineering Software*, 40, pp.341-349. 2009

⁵⁴ Baykasoglu, Adil; Dereli Turkey; Sabuncu, Ibrahim. An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. *Omega* 34, pp.385-396. 2006

condiciones industriales muy volátiles. En su investigación muestran que un ACO puede ser aplicado a éste tipo de problemas y para ello presentan diferentes ejemplos computacionales que muestran la efectividad de su algoritmo.⁵⁵

2.4.3.5 Optimización por enjambre de partículas: La optimización por enjambre de partículas o PSO (por sus siglas en inglés, *Particle Swarm Optimization*) es un algoritmo metaheurístico diseñado por Kennedy y Eberhart en el año 1995. Este método se basa en la simulación de comportamientos sociales tales como la reunión en manada de ciertos animales, la congregación de las aves y los peces.

Este algoritmo fue diseñado para resolver problemas de optimización y al igual que otras técnicas de computación evolutiva, PSO es un algoritmo de búsqueda basado en la población ya que inicia con una población de soluciones aleatorias, llamadas partículas. Difiere de otras técnicas evolutivas en que este método asocia cada partícula con una velocidad que se ajusta dinámicamente en función al comportamiento histórico de cada partícula para luego evaluar la función objetivo según la posición de dicha partícula y así a través de la iteración encontrar una buena solución al problema.

Recientemente, Ali Derakshan & Kwan Wong (Febrero, 2015) desarrollaron un algoritmo de enjambre de partículas PSO modificado para solucionar un problema de FLP dinámico o estático con áreas desiguales. Para ello, se sugirió que los departamentos tienen áreas y formas fijas que se mantienen constantes en el tiempo. El algoritmo planteado fue evaluado por medio de problemas disponibles en la literatura.

⁵⁵ Hani, Yasmina; Amodeo, Lionel; Yalanoui, Farouk & Chen, Haoxun. Ant colony optimization for solving an industrial layout problem. *European Journal of Operational Research*, 183, pp.633-642. 2007

De forma un poco más general, se puede mencionar que los pasos que sigue comúnmente un PSO normal son los siguientes:

1. Crear las partículas y evaluarlas
2. Determinar la mejor solución en cada una de las partículas y la mejor solución global
3. Actualizar la velocidad y la posición de cada partícula teniendo en cuenta los criterios mencionados anteriormente
4. Si se cumplen las condiciones de detención del algoritmo, pasar al siguiente paso. Si no es así, volver al paso 2
5. Fin del algoritmo⁵⁶

2.4.4 Recocido simulado en la optimización combinatoria y problemas del tipo FLP. Dentro del campo de la optimización combinatoria Kirkpatrick et al (1983)⁵⁷ fueron pioneros en la formulación de dicho método. El método de recocido simulado parte desde un estado inicial desde el cual mediante una perturbación aleatoria en su vecindad se desplaza a un nuevo estado lo cual causa un cambio en el valor de la función objetivo representado por ΔE . Dentro de un proceso de minimización como el que se pretende para este proyecto, sólo un valor negativo de ΔE será aceptado para validar la transformación del valor de la función objetivo con una probabilidad de $p(\Delta E) = e^{\frac{-\Delta E}{Kb+T}}$ donde T es un parámetro de control que simula a la temperatura en analogía con el proceso industrial y Kb es la constante de Boltzman.

De esta forma a través de la manejo del parámetro T se puede controlar la probabilidad de aceptación de una transformación en el valor de la función objetivo de la forma $p(\Delta E)$, de manera que el proceso iterativo trascienda desde las

⁵⁶ Asl, Ali Derakhsha & Kuan Yew Wong. Solving unequal-area static and dynamic facility layout problems using modified particle swarm optimization. *Manufacturing and Industrial Engineering Journal*. 2015

⁵⁷ Kirkpatrick, Scott; Gelatt, Daniel; Vecchi, Mario P. Optimization by simulated annealing, *Science*, 220, pp 671-680. 1983

soluciones óptimas locales generadas al inicio del recocido hacia soluciones globales de más alta calidad.

Según el autor, las principales consideraciones según dentro del método de recocido simulado son:

- La temperatura T , la cual es el parámetro que controlará la probabilidad con la cual se aceptará peores soluciones al momento de saltar de óptimos locales en búsqueda de la mejor aproximación al óptimo global. Durante el proceso de búsqueda el parámetro T será disminuido gradualmente de manera que la probabilidad de aceptación de una peor solución se disminuya y de esa manera mejorar la calidad de las soluciones generadas.
- El punto de equilibrio el cual establece una condición dentro de la cual el proceso iterativo se detendrá, generando una buena solución al problema.
- El programa de recocido, el cual determina en qué cantidad y cada cuánto se reducirá la temperatura.

Dentro del campo del problema FLP, el recocido simulado se ha empleado para dar solución a múltiples variaciones del problema. Heragu y Alfa (1992)⁵⁸ implementan el algoritmo para dar solución a una distribución de una sola línea de producción (single-row FLP) y a una distribución con múltiples líneas (multi-row FLP) para los cuales se estableció un criterio de parada fijo basado en el número de iteraciones determinadas por el número n de departamentos a distribuir de la forma $100n$.

Jajodia, S et al (1992)⁵⁹ aplicaron recocido simulado a una distribución de celdas o células de trabajo (*cellular FLP*) para lo cual se debió determinar la posición relativa

⁵⁸ Heragu, Sunderesh S. & Alfa, Attahiru Sule. Experimental analysis of simulated annealing based algorithms for the layout problem, European journal of operational research, 57, 1992, pp 190-202.

⁵⁹ Jajodia, Satish Minis, Ioannis; Harhalakis, George & Proth, Jean-Marie. Class: computerized layout solutions using simulates annealing, International journal of production research, 30, 1, 1993, pp 95-108.

de n entidades las cuales podían pertenecer a una sola célula (intra-cell) o representar las celdas de trabajo existentes dentro de una planta (inter-cell). Para ambos casos el objetivo fue minimizar el costo total de transporte de material entre las entidades. A la metodología propuesta por los autores se le denominó CLASS por sus siglas en inglés que traducen: Soluciones computarizadas de distribución a través de recocido simulado.

Puntualizando en el ámbito de Multi-Floor FLP en el cual se centra este proyecto Kievani, A et al (2010)⁶⁰ aplicaron el algoritmo para dar solución a dicha variación del problema en la cual en dos niveles se modelaron distribuciones con 3, 5, 7, 10 y 15 departamentos y se compararon los costos resultantes de dichas réplicas con los generados mediante un método exacto validando el modelo de ésta manera.

⁶⁰ Ali Kievani, Seyed Nima Rafienejad, Mohamadreza Kaviani & Hossein Afshari. A simulated annealing for multi floor facility layout problem En: The world congress of engineering and computer science (2010, Octubre 20-22, San Francisco, USA), p 1-2.

3. REVISIÓN DE LITERATURA

El diseño o planeación de instalaciones ha sido estudiado desde hace tiempo por varios autores dado su impacto en el desempeño del sistema productivo en las empresas. El término “distribución de instalaciones” se constituye de tres actividades principales que lo identifican como una etapa del estudio de los sistemas de manufactura: El diseño del sistema productivo, diseño del sistema de manejo de materiales y finalmente el diseño de la distribución de la planta, en el cual se enfocará el proyecto⁶¹

Determinar la configuración más eficiente de centros de trabajo dentro de una instalación caracterizada se define como el Problema de Distribución de Planta (FLP por sus siglas en inglés Facility Layout Problem). Estos problemas se identifican por su alta complejidad y rigurosidad algorítmica, dado que métodos como branch & bound sólo permiten solucionar problemas de menor escala. Para su resolución se necesitan herramientas metaheurísticas ya que se considera un problema de categoría NP-Hard.

El problema de distribución de planta de un solo nivel o piso ha sido objeto de estudio extensivo iniciando con el planteamiento del modelo lineal entero mixto presentado por Love y Wong (1976)⁶² al cual se le dio solución a través de métodos exactos como *Branch and Bound*, el cual fue planteado por Kaku y Thompson (1986)⁶³, y más adelante mediante transformaciones lineales del problema de

⁶¹ Faraz Ramtin, M. Abolhasanpour, Hossein Hojabri, Ahmad Hemmati. & Ayyoub A. Jaafari. Optimal Multi Floor Facility Layout. International MultiConference of Engineers and Computer Scientists. Pages 1797-1800. Marzo 2010

⁶² Robert Love & Jsun Y. Wong. Solving quadratic assignment problems with rectangular distances and integer programming. *Naval Research Logistics Quarterly*, 23, 623–627. 1976

⁶³ Bharat K Kaku & Gerald L. Thompson An exact algorithm for the general quadratic assignment problem. *European Journal of Operational Research*, 23, 382–390. 1986

asignación cuadrática (QAP) realizadas por Bazzara y Sherali (1980)⁶⁴ y Burkard y Bonniger (1983)⁶⁵. Dichos planteamientos exactos se realizaron para problemas de pequeña escala (escaso número de departamentos a distribuir).

Meller y Gau (1996)⁶⁶ desarrollan el primer planteamiento discreto del problema de distribución de planta FLP de una sola dimensión (piso) al considerar tanto el espacio de la instalación como el área de los departamentos o centros de trabajo como una colección de departamentos de áreas iguales. Anterior al modelo discreto, Montreuil (1987) había planteado un modelo continuo en el cual se configuraron los departamentos como polígonos, acercándose a una formulación más realista, ya que los lados ortogonales y vértices de dichos polígonos permitían que los centros de trabajo se ubicaran en cualquier lugar dentro del espacio del piso, pero el trabajo de hallar la solución óptima a estos enfoques se desarrolló por Meller et al (1999)⁶⁷, Al-Khayyal et al (2001)⁶⁸ y Sherali et al (2003)⁶⁹.

La configuración de planta sugerida por los algoritmos anteriores no era aplicable en la mayoría de los casos, debido a restricciones de espacio y dimensión de los centros de trabajo ubicados. Para hacer más prácticas las distribuciones generadas por dichos planteamientos a la hora de su implementación se introdujo dentro de los modelos la configuración del sistema de manejo de materiales. Trabajos en este

⁶⁴ Mokhtar S. Bazzara & Hanif D. Sherali. Benders' partitioning scheme applied to a new formulation of quadratic assignment problem. *Naval Research Logistics Quarterly*, 27(1), 29–41. 1980

⁶⁵ Rainer E. Burkard & Tilman Bonniger. A heuristic for quadratic Boolean program with application to quadratic assignment problems. *European Journal of operational Research*, 13, 347–386. 1983

⁶⁶ Russell D. Meller & Kai-Yin Gau. "The Facility Layout Problem: Recent and Emerging Trends and Perspectives". *Journal Of Manufacturing Systems*, 15(5), 351-366. 1997

⁶⁷ Russell D Meller, Venkat Narayanan, Pamela H Vance. Optimal Facility Layout Design. *Operations Research Letters*, 23, 117-127. 1999

⁶⁸ Faiz Al-Khayyal, Marc Goetschalckx & Timothy Van Voorhis, Alternative formulations and new solutions for the facility layout problem. *Reporte Técnico*, Escuela de Ingeniería Industrial y Sistemas de Ingeniería, Georgia Institute of Technology, Atlanta, Georgia. 2001

⁶⁹ Hanif D. Sherali, Barbara M. P. Fraticelli & Russell D. Meller, Enhanced Model Formulations for Optimal Facility Layout, *Operations Research*, 51(4), 629-644. 2003

campo se originaron inicialmente por Montreuil (1987)⁷⁰. Goetschalckx & Kim (2006)⁷¹ plantean un algoritmo que permite regularizar los límites de los diferentes departamentos y centros de trabajo de manera que el flujo de materiales se distribuye a través de pasillos delimitados por esquinas.

En contraste a la vasta investigación realizada sobre el problema de distribución de planta de un solo nivel, el estudio dirigido al problema FLP multinivel es limitado. Johnson (1982)⁷² fue uno de los primeros en dirigir su atención hacia dicho problema al definir la localización relativa de diferentes departamentos dentro de un edificio de varios niveles. Tiempo después, diferentes autores se enfocan en considerar movimientos verticales del flujo de materiales de un piso a otro a través de elevadores, lo cual origina como tal el problema FLP multinivel.

Bozer et al (1994) emplean curvas de llenado espacial que permitían limitar los espacios libres entre departamentos e intercambios orientados por parejas de centros de trabajo para resolver de manera rápida los planteamientos discretos del problema y así generar distribuciones de planta. Un modelo de descomposición jerárquico fue propuesto por Meller y Bozer (1997)⁷³, donde se asignaban primero los departamentos a los niveles y seguidamente se organizaban dichos departamentos en cada nivel para así obtener una distribución funcional donde no se sobrepusieron centros de trabajo.

⁷⁰ Benoit Montreuil, Integrating design of cell layout, input/output I/O point configuration, and flow network of manufacturing systems. En: Liu, C.R., Requicha, A. y Chandrasekar, S. Intelligent and Integrated Manufacturing Analysis and Synthesis. Ped-vol 25, ASME: New York, 315-326. 1987

⁷¹ Marc Goetschalckx & Jin-Kuk Kim, An integrated algorithm for the design of block layouts with regularized material handling flow paths, Progress in Material Handling Research: 171-185. 2006

⁷² Roger Vivian Johnson. SPACECRAFT for multi-floor layout planning. Management. Sciences, 28(4), 407–417. 1982

⁷³ Russell D. Meller & Yavuz A. Bozer, Alternative approaches to solve multi-floor facility layout problem, Journal of Manufacturing Systems, 16(3), 192-203. 1997

Continuamente varios investigadores se enfocan en las consideraciones respectivas a los movimientos verticales de material entre un nivel y otro (Bozer, Meller & Erlebacher, 1994)⁷⁴. Para dicho propósito el instrumento comúnmente utilizado en los estudios en el tema es el elevador (Lee, Roh & Jeong, 2005). El número y la localización de los mismos es conocida de antemano (Lee et al. 2005)⁷⁵ o es determinada a través de optimización (Matsuzaki et al., 1999)⁷⁶.

El uso de metaheurísticas se evidencia a lo largo de la investigación realizada en el campo del diseño de la distribución de planta ya que se convierte en la mejor opción de tratamiento para dichos problemas por su naturaleza NP-Hard. Abdinnour, Helm y Hadley (2000)⁷⁷ desarrollaron un proceso minucioso de búsqueda aleatoria adaptativa (GRASP por sus siglas en inglés) y un algoritmo no lineal de optimización para crear una distribución de planta inicial que luego era mejorada a través de un método de recombinación de centros de trabajo similar a la búsqueda tabú (*Tabu Search*) que junto al recocido simulado (*Simulated Annealing*) y al desarrollo de algoritmos genéticos (*Genetic Algorithm*) son las metodologías de solución más comunes para los problemas FLP en general.

En la última década se ha evidenciado un aumento en el interés científico por el estudio del problema de distribución de planta con más de un nivel y se han aplicado diversos enfoques a la solución del mismo; de forma que el campo de estudio se ha enriquecido.

⁷⁴ Yavuz A. Bozer, Russell D. Meller & Steven J. Erlebacher, An Improvement-Type Layout Algorithm for Single and Multiple Floor Facilities, *Management Science*, 40(7), 918-932. 1994

⁷⁵ Kyu-Yeul Leea, Myung-Il Rohb, Hyuk-Su Jeongb. An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. *Computers & Operations Research*, 32(4), 879-899. 2005

⁷⁶ Kenichiro Matsuzaki, Takashi Irohara, Kazuho Yoshimoto, K. Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization. *Computers & Industrial Engineering*, 36(2), 487-502. 1999.

⁷⁷ Sue Abdinnour-Helm & Scott W. Hadley, Tabu search based heuristics for multi-floor layout, *International Journal of Production Research*, 38(2), 365-383. 2000

Chen, Xiao & Tang (2011)⁷⁸ lograron plantear un modelo robusto para el FLP multi-floor con capacidad de transporte vertical limitada, es decir, un elevador con una capacidad restringida. Su intención al incluir dicha restricción fue lograr que el modelo propuesto reflejara una situación real en la cual el transporte vertical no siempre está disponible inmediatamente y se producen tiempos de espera para llevar el material de una planta a otra. Para simular dicha situación los autores se apoyaron en el software AUTOMOD con el fin de obtener un tiempo de espera promedio para la disponibilidad del elevador; además de las consideraciones referentes al elevador y a la minimización del costo de manejo de material también se incluyó como objetivo maximizar la relevancia que el modelo otorgaba a los productos con una alta demanda, ya que las demoras en el elevadora harían que el proceso de *picking* se tardara, logrando así un esquema jerarquizado de tiempos para la salida de productos. Dicho modelo fue resuelto empleando algoritmo genético.

Típicamente los problemas de distribución de planta de más de un nivel consideran fija la localización de los elevadores eliminando las restricciones que surgen al habilitar su libre localización, Rodrigues, Gaspar & Gomes (2013)⁷⁹ atacan éste problema al caracterizar los elevadores y escaleras como objetos parametrizables, es decir, cuya forma, tamaño y demás características podrían variar a lo largo del proceso de solución, logrando así una mayor diversidad de distribuciones posibles. Los autores emplearon esta investigación en el campo de la arquitectura, para alcanzar un mejor entendimiento de los requerimientos locativos de las estructuras según sus usos (hospitales, bancos, empresas de servicios y de manufactura, etc). Los autores plantearon un algoritmo evolutivo derivado del algoritmo genético que permitió dar solución al problema.

⁷⁸ Yanru Chen, Qinxin Xiao & Xiaoping Tang. Product layout optimization and simulation model in a multi-level distribution center. Systems engineering procedia 2. Vol 2. Pp 300-307. 2011

⁷⁹ Eugénio Rodrigues, Adélio Rodrigues Gaspar & Álvaro Gomes. An approach to the multi-level space allocation problem in architecture using a hybrid evolutionary technique. Automaton in construction. Vol 35. Pp 482-498. 2014.

Hosseini, Mirzapour & Wong (2013)⁸⁰ emplearon un modelo de planeación sistemática de la distribución de planta (SLP) a una compañía de envío de paquetes, dicho modelo fue propuesto por Richard Muther en 1961. El enfoque SLP permite hallar múltiples alternativas de distribución sensible a los flujos de material que se mueven entre centros de trabajo además de considerar la cercanía entre éstos. Para el planteamiento de dichas alternativas se deben considerar las relaciones entre los departamentos, la disponibilidad de espacio dentro del recinto y el área que cada uno de dichos departamentos demanda.

Una vez fueron obtenidas las alternativas se procedió a evaluar cada una de ellas mediante el software ARENA realizando una comparación entre su desempeño y costo para luego tomar una decisión.

Algunos autores se han empeñado en cerrar la brecha entre el conocimiento teórico y las situaciones que se presentan en el mundo real que impiden la aplicación de los modelos convencionales. Izadinia, Eshghi & Salmani (2014)⁸¹ basaron su enfoque al problema de distribución de planta de múltiples niveles en la imposibilidad de fijar valores para los parámetros de dicho problema, desvirtuando de esta forma los modelos deterministas que se encuentran en la literatura. Para robustecer su modelo y hacerlo aplicable a situaciones reales los autores optan por definir parámetros cuyos valores numéricos se representaran por intervalos y así minimizar el costo del transporte de material considerando la incertidumbre inherente a los diferentes parámetros del problema FLP multifloor.

⁸⁰ Seyedeh Hosseini, Seyed Ali Mirzapour & Kuan Yew Wong. Improving multi-floor facility layout problems using systematic layout planning and simulation. *Communications in computer and information science. Advances in information technology*. Vol 409. Pp 58-69. 2013

⁸¹ Niloufar Izadinia, Kouros Eshghi & Mohammad Hassan Salmani. A robust model for multi floor optimization. *Computers & Industrial Engineering*. Vol 78. Pp 127-134. 2014

Kia et al (2014)⁸² propusieron un modelo de programación entera mixta para el diseño de una distribución de planta empleando un sistema de manufactura organizado en células de trabajo. Dicho sistema implica ciertas consideraciones como la formación de la célula de trabajo (la agrupación de actividades con requerimientos de funcionamiento similares en familias), la distribución intra-celular (el arreglo de maquinaria ideal para cada célula), la distribución inter-celular (la ubicación de cada célula dentro de cada piso y sus relaciones intrínsecas), la programación de cada célula (el ritmo de trabajo de cada célula en relación con aquellas partes que necesita para producir y las partes que las demás células demandan de la misma) y la disposición de recursos para sostener toda la operación.

Para adaptar dicho modelo a la vida real los autores emplearon un enfoque dinámico al planteamiento, de manera que factores como la vida útil de los productos, la introducción de nuevos productos y demás factores inesperados y volátiles como la demanda esperada y costos extraordinarios no interrumpan la producción ni obliguen a una reconfiguración de la distribución de la planta.

Aún con los diferentes enfoques que se han aplicado al estudio del problema FLP Multi-floor el uso de metaheurísticas como metodologías de solución se mantiene presente. Izadinaa & Eshghia (2016)⁸³ plantearon un modelo discreto especial que en adición a las condiciones típicas del problema de distribución de planta de más de un piso, consideraba desconocida la demanda.

⁸² Reza Kia, Fahimeh Khaksar-Haghani, Nikbakhsh Javadian & Reza Tavakkoli Moghaddam. Solving a multi-floor layout design model of a dynamic cellular manufacturing system by an efficient genetic algorithm. *Journal of manufacturing systems*. Vol 33(1). Pp 218-232. 2014.

⁸³ Niloufar Izadinaa & Kourosh Eshghia. A robust mathematical model and ACO solution for multi-floor discrete layout problem with uncertain locations and demands. *Computers and Industrial Engineering*. 2016

Dentro de la instalación se aclara que el sótano de la estructura se emplea para almacenaje de inventarios y los diferentes departamentos pueden ser localizados en los demás niveles en posiciones predeterminadas y los flujos de material no son constantes debido a la demanda. Para dar solución al modelo se empleó la metodología de colonia de hormigas (ACO).

4. MARCO DE ANTECEDENTES

En Colombia, Adriana García Londoño (2013)⁸⁴ de la Universidad Javeriana implementó un algoritmo genético para dar solución a una de las variantes del problema de distribución de planta clásico en la cual se introdujeron consideraciones de demanda dinámicas a un umbral de tiempo estocástico a partir de la formulación realizada por Moslemipour & Lee (2011) basada en el modelo de asignación cuadrática (QAP) integrando la variabilidad de dicha demanda y los estadísticos pertinentes a la misma.

El concepto dinámico contemplado por la autora hace referencia a que en adición a las consideraciones de flujo de materiales y distancia entre los departamentos, el modelo planteado por la autora considera los escenarios probabilísticos que surgen del comportamiento de la demanda de productos, lo cual afecta la flexibilidad de la fábrica para satisfacer dicha demanda y su capacidad de efectuar diseños robustos que respondan positivamente hacia la incertidumbre del entorno.

A partir del modelo QAP se generó un algoritmo evolutivo (genético) que a partir de la hibridación de soluciones factibles específicas y la mutación o evolución de un conjunto aleatorio de dichas soluciones se encuentra una respuesta o distribución estocástica óptima para el problema planteado. Este algoritmo fue traducido a un pseudocódigo que fue programado en Visual Basic de Excel para facilitar su uso a cualquier usuario de la plataforma Microsoft Office.

⁸⁴ García Londoño, Adriana: Solución del problema dinámico estocástico de la distribución de planta mediante el uso de un algoritmo genético. Bogotá. Trabajo de Grado (Ingeniera Industrial). Pontificia Universidad Javeriana. Facultad de Ingenierías, Escuela de Ingeniería Industrial. 2013

Finalmente se realizaron pruebas de desempeño del algoritmo para validar el funcionamiento del pseudocódigo en dos problemas formulados por Moslemipour & Lee en su artículo, obteniéndose resultados satisfactorios al no sobrepasar una dispersión del 5% en ambos problemas, validando el algoritmo genético planteado.

En la universidad del Valle los autores Moreno. Alfredo D, Álvarez. Aldair A, Noble. Víctor M y López, Jorge M. (2013)⁸⁵ abordaron el problema de distribución de instalaciones denominado UA-FLP, el cual se caracteriza por considerar las áreas dispuestas para los diferentes departamentos a ubicar de naturaleza desigual priorizando como criterio de distribución la minimización de la distancia entre un departamento y otro. Los autores en su artículo modifican el problema al incluir además la minimización del costo de manejo de materiales entre las estaciones de trabajo y formulan una metodología de solución para el problema multiobjetivo que se genera a partir de dichas consideraciones.

Inicialmente se estructura el cálculo de las distancias entre instalaciones. Este se hace de manera aleatoria al no encontrarse en la literatura un criterio que relacione la correspondencia lógica que entre dos instalaciones se genera a nivel de cercanía con los costos originados por el transporte de material de una instalación a otra.

Seguidamente se establece el parámetro α , el cual representa la importancia que cada función objetivo contemplada dentro de la metodología tiene en la toma de decisiones. Dichas funciones se construyen a partir de los criterios de decisión a los que se pretende responder mediante la formulación del modelo matemático.

⁸⁵ Moreno, Alfredo D. Álvarez, Aldair A. Noble, Víctor M. López, Jorge M.. Optimización multiobjetivo del problema de distribución de planta: Un nuevo modelo matemático. En Ingeniería y Competitividad, 16, N°2, pp 257-267.

Se condujeron experimentos computacionales con 17 diferentes problemas encontrados en la literatura para los cuales se estableció que el parámetro α tomará valores entre 0 y 1 para determinar el efecto de la ponderación en la distribución generada por el modelo y el costo que supone dicha instalación. Dicha experimentación debió someterse a evaluación dado que no existen en la literatura formulaciones del problema UA-FLP que aborden los criterios de distancia y costo simultáneamente. Para esto se tomó como indicador el valor de α utilizado en cada experimento y se analizó la calidad de la respuesta obtenida validando así la metodología planteada.

En la Universidad Industrial de Santander - UIS, Melissa Palacios Mendoza y Christian Camilo Jaimes Tami (2011)⁸⁶ abordaron el problema de FLP de áreas iguales y desiguales mediante un algoritmo híbrido genético (HGA), el cual aprovecha la efectividad de los métodos Búsqueda Tabú (TS), Recocido Simulado (SA) y Algoritmos Genéticos (GA), para encontrar la solución global del problema a partir de la formación de cadenas de algoritmos genéticos. Diseñaron un código en lenguaje MATLAB capaz de solucionar el problema documentado y también se analizaron los resultados obtenidos mediante un contraste con problemas resueltos en la literatura.

Erika Melissa Rueda Jaimes y Giovanny Andrés Villamil Rodríguez (2014)⁸⁷, estudiantes de la Universidad Industrial de Santander, desarrollaron una alternativa de solución al problema de distribución de planta (FLP) de un solo nivel, con estructuras internas: muros y pasajes, considerando Algoritmos Genéticos.

⁸⁶ Palacios Mendoza, Melissa & Jaimes Tami, Christian: Alternativa de solución al problema de distribución de planta (flp) para instalaciones de áreas iguales y desiguales mediante un algoritmo híbrido genético. Bucaramanga. Trabajo de Grado (Ingeniera Industrial). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería Industrial. 2011.

⁸⁷ Rueda Jaimes, Erika & Villamil Rodríguez, Giovanny: Alternativa de solución al problema de distribución de planta (flp) de un solo nivel con estructuras internas, muros y pasajes, a través de métodos metaheurísticos. Bucaramanga. Trabajo de Grado (Ingeniera Industrial). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería Industrial. 2014.

Propusieron una alternativa de solución al problema programando en el software MATLAB un nuevo algoritmo partiendo de las características de la planta.

Moncayo, Alvarado & Osorio (2014)⁸⁸ desarrollaron un modelo de solución al problema de distribución de planta en celdas de manufactura a través de la formulación de dos nuevos algoritmos híbridos discretos basados en la quimiotaxis de bacterias y los algoritmos genéticos los cuales abordan simultáneamente los dos principales inconvenientes encontrados al plantear el modelo, los cuales eran la formación de las celdas (las diferentes máquinas que conforman cada celda) y la distribución de planta tanto dentro de las celdas (ubicación de la maquinaria dentro de una misma celda) como entre las mismas (la posición de cada celda dentro del espacio) manteniendo el objetivo de minimizar el costo de manipulación de materiales. Dicho modelo también se mostró efectivo para solucionar problemas del vendedor viajero.

Cesar Julio Collazos Valencia (2013)⁸⁹ implementó el método de planeación sistemática de instalaciones para resolver el problema de distribución de planta de una empresa de alimentos llamada “+ Natural”, la cual posee 5 líneas de producción encargadas cada una de un producto diferente (productos A,B,C,D y E). El “*layout*” o distribución de las líneas se ha visto afectado en su calidad de manera negativa gracias a los diferentes cambios introducidos a través del tiempo a razón de la volatilidad de la demanda de productos y nuevas tecnologías adoptadas por la compañía; lo cual ha generado cuellos de botella, cruce de flujos y pérdida de eficiencia.

⁸⁸ Camilo Mejía Moncayo, Diego Alexander Garzón Avarado & José Manuel Arroyo Osorio. Métodos discretos basados en quimiotaxis de bacterias y algoritmos genéticos para solucionar el problema de distribución de planta en celdas de manufactura. Ciencia e ingeniería neogranadina. 24 (1). Pp 6-28. 2014.

⁸⁹ Cesar Julio Collazos Valencia. Rediseño del sistema productivo utilizando técnicas de distribución de planta: Caso de estudio planta procesadora de alimentos. Manizales. Trabajo de Grado (Maestría en Ingeniería Industrial). Universidad Nacional de Colombia. Facultad de ingeniería y arquitectura. 2013.

A partir del análisis de la demanda anual se determinaron las velocidades con las cuales cada producto debía ser fabricado para así identificar el tipo de distribución que debía tener cada línea de producción. Seguidamente con base en las relaciones espaciales entre los departamentos se construyó y aplicó un algoritmo genético para dar solución al problema.

5. ALGORITMO DE RECOCIDO SIMULADO APLICADO AL PROBLEMA DE DISTRIBUCIÓN DE PLANTA (FLP) DE Múltiples NIVELES, UN ELEVADOR Y DEPARTAMENTOS de áreas desIGUALES.

A continuación se muestran las características del algoritmo programado en lenguaje matlab basado en la metaheurística de Recocido Simulado para dar solución al problema de distribución de planta de más de un nivel.

Para construir dicho algoritmo se establece primero el modelo matemático a utilizar. En este proyecto se ha optado por seguir el modelo propuesto por Meller & Bozer⁹⁰ donde se definen:

- f_{ij} Es el flujo de material desde el departamento i al departamento j
- C_{ij}^H Y C_{ij}^V son los costos de transportar una unidad de flujo de material, una unidad de distancia desde el departamento i al j en dirección horizontal y vertical
- d_{ij}^H Y d_{ij}^V son las distancias horizontales y verticales existentes entre los departamentos i y j

La función objetivo del modelo matemático es:

$$\min \sum_i \sum_j (C_{ij}^H d_{ij}^H + C_{ij}^V d_{ij}^V) f_{ij}$$

⁹⁰ Russell Meller & Yavuz Bozer. Alternative approaches to solve the multi-floor facility layout problem. Journal of Manufacturing Systems. Vol 16 (3). P192-203. 1997.

5.1 DESCRIPCIÓN DEL ALGORITMO

El algoritmo programado se abordó en tres etapas de modo que cada etapa se construyera sobre la etapa inmediatamente anterior y así el código se edificara integralmente para satisfacer las condiciones establecidas en el proyecto y se posibilitara la validación del método a través de la comparación de resultados obtenidos en contraste a problemas encontrados en la literatura. Ver ANEXO N°2.

5.1.1 Etapa de parametrización y generación del punto de partida. En la primera etapa de construcción del algoritmo se trazó el código base sobre el cual se establece la metaheurística de Recocido Simulado empleada para dar solución al problema planteado en el proyecto. Este código base se evalúa a partir de funciones de prueba (función de Bohachevsky) validando la estructura programada para la metaheurística.

Luego de garantizar el funcionamiento de la metaheurística en el código base se procede a enfocar la estructura del código hacia el problema FLP Multifloor mediante la guía de problemas encontrados en la literatura de modo que éstos ayuden a refinar el código en cuanto a los datos requeridos para efectuar un proceso iterativo efectivo y la estructura que el mismo debe poseer para que la solución obtenida al finalizar el proceso sea una representación acertada del problema que se quiere resolver.

Dicha estructura tiene como punto de partida la definición de los siguientes parámetros: el número de departamentos a ubicar denotado por el símbolo “N”, se especifica el área máxima permitida por piso como A_k ya que dicha área debe ser garantizada para que el algoritmo sea aplicable a un nivel real. Se procede a introducir las coordenadas del elevador, las cuales se pueden ajustar dependiendo de lo que el programador requiera.

Se definen las probabilidades de aceptación de soluciones de menor calidad al inicio y final del proceso iterativo, las “temperaturas” inicial y final de acuerdo a las probabilidades definidas, y seguidamente se establece el ritmo de enfriamiento para el proceso. A continuación se procede a generar un punto de partida aleatorio para el proceso.

Figura 8. Primera etapa del algoritmo.

```
% Internal variables

N=40; %numero de departamentos
Amax=140; % Area maxima permitida por piso
lift=[11 1]; % posiciones del ascensor

n_cycles = 10;
n_tries_for_cicle = 10;

target_function = 'bohal';
n_dimention = 2;

% Probabilities of accept the worst solution
initial_probability = 0.7;
final_probability = 0.001;

% Calculate temperature and steps
initial_temperature = -1 / log(initial_probability);
final_temperature = -1 / log(final_probability);
reduction_fraction = (final_temperature / initial_temperature)^(1/
(n_cycles-1));

% Initial position
% Generate random number in the range of each dimention

range_min=1;
range_max=40;
```

5.1.2. Cálculo del costo vertical. Continuando con el proceso de construcción del algoritmo se hace necesario conocer los parámetros de costo de transporte horizontal y vertical por unidad de distancia transportado del problema ya que estos hacen parte de la función objetivo de minimización de costo. Así mismo se deben conocer tanto el flujo que sigue cada producto, el área de cada departamento y la cantidad de material que se transporta dentro de cada flujo.

La ruta o flujo productivo se refiere al camino que sigue el material dentro de su proceso de fabricación, es decir, los departamentos por los que pasa el material durante su manufactura. Dado que usualmente en una fábrica se producen más de un tipo de producto los flujos deben ser especificados por cada ítem fabricado para lograr acoplar todas las rutas dentro de la optimización y así asegurar el funcionamiento del algoritmo. Conocer dichos flujos permite que el algoritmo reconozca la distancia que cada ítem recorre y de esta manera el costo asociado a este transporte.

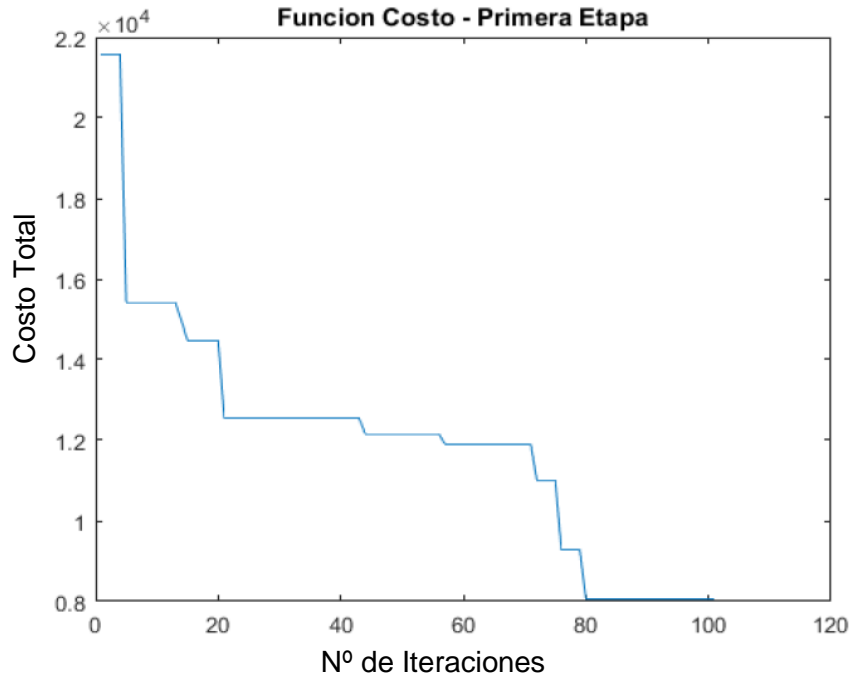
La distancia recorrida por cada producto se cuantifica a partir de la distancia entre los centroides de los departamentos por los que pasa el mismo, por tanto se requieren los datos de área de cada departamento.

Finalmente el diseño del algoritmo hace necesario conocer la cantidad de material que se transporta en cada flujo, ya que esto permite globalizar el costo que se maneja.

Con los datos anteriores se procede a definir la distribución vertical de los departamentos, es decir, se establecen los departamentos que se ubican en cada piso, sin embargo su arreglo o *“layout”* dentro de cada nivel de la estructura se hace aleatoriamente ya que el refinamiento de dicha organización es el siguiente paso que deberá seguir el algoritmo.

De dicha distribución primaria de departamentos se obtiene una función de costo vertical denominada “Función de Costo – Primera Etapa”, la cual demuestra la disminución del costo en la función objetivo tomando en cuenta que en éste punto sólo se han asignado departamentos a los diferentes niveles de la instalación. Esto es posible gracias a que la definición de flujos no se ve afectada gracias a la distribución de los departamentos.

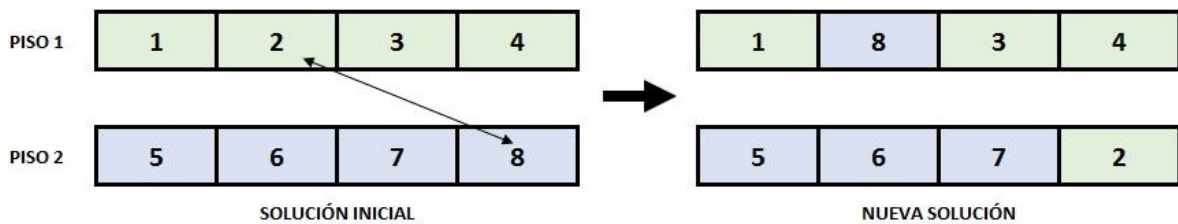
Figura 9. Ejemplo de Gráfica de Función Objetivo – Primera Etapa



La asignación de los departamentos a cada piso se realiza partiendo de una solución aleatoria. En base a esta se realiza un proceso de permutación por cada iteración, es decir, el algoritmo reordena departamentos aleatoriamente, lo cual se conoce como la búsqueda en la vecindad de la solución inicial. Cada vez que se realiza una permutación se evalúa el valor del costo total asociado a la nueva distribución de planta generada con dicha iteración y ese valor entra a ser examinado según el criterio de aceptación definido en el recocido simulado.

Si la nueva solución factible conlleva un valor de costo total menor a la solución obtenida en la iteración inmediatamente anterior, la nueva solución se acepta y se repite el proceso. Si se da el caso contrario, la nueva solución es juzgada según la probabilidad de aceptación definida en el recocido simulado. En dicho caso la nueva solución puede ser aceptada para continuar con el proceso, o descartada.

Figura 10. Ejemplo de iteración para el costo vertical.

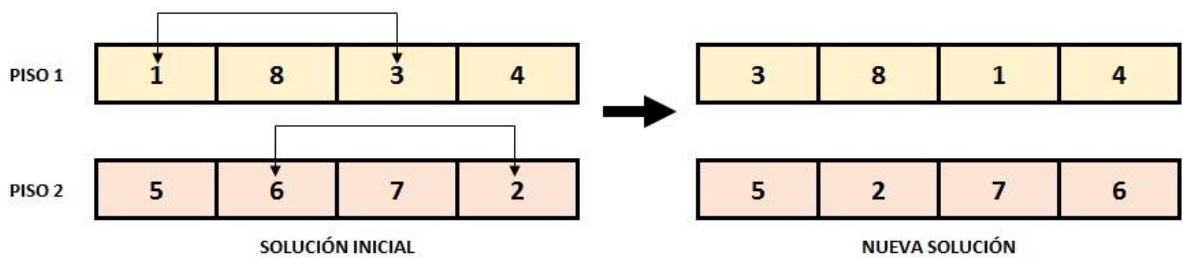


Al terminar el proceso en la segunda etapa, la reorganización obtenida de los departamentos asociados a cada piso acarrea el mínimo costo vertical y se procede a la siguiente parte del algoritmo.

5.1.3. Cálculo del costo horizontal. Realizada la distribución de los departamentos en los pisos dispuestos dentro de la instalación se procede a optimizar el arreglo de dichos centros de trabajo dentro de los niveles. Para esto el algoritmo lleva a cabo una reorganización del “*layout*” de cada piso mediante iteraciones que siguen la metodología del recocido simulado o simulated annealing.

El algoritmo realiza iteraciones de manera aleatoria dentro de cada piso, en las cuales se intercambia el orden de un par de departamentos para generar una nueva solución que será evaluada posteriormente. El resultado de dicha evaluación permitirá saber si el costo total obtenido con la nueva solución es menor o mayor al costo total adscrito a la iteración anterior. A continuación se procede a evaluar si dicha solución es aceptada o rechazada según el criterio de la metaheurística de recocido simulado, como se explicó en la etapa anterior.

Figura 11. Ejemplo de iteración para el costo horizontal.

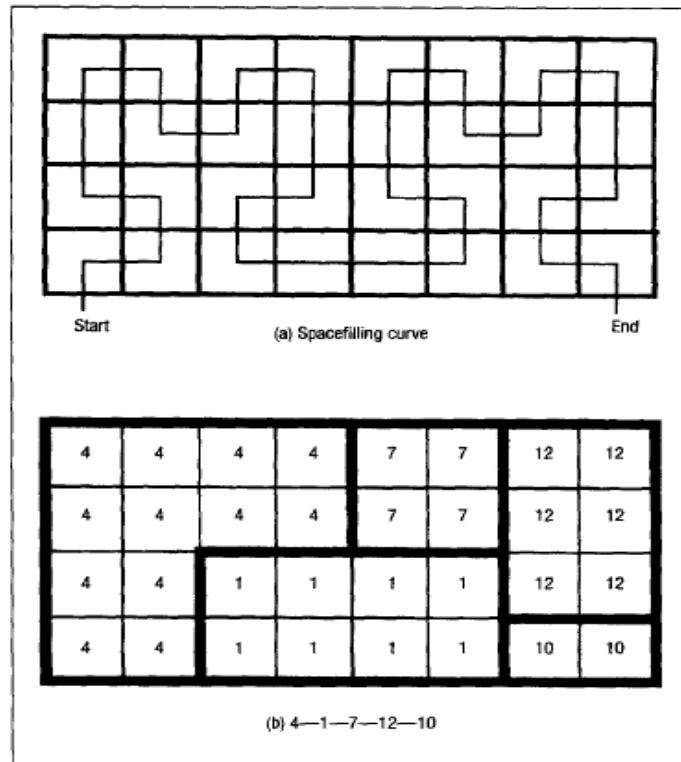


Dicho proceso toma un tiempo de cómputo considerablemente menor al necesario para determinar los departamentos asignados a cada piso ya que el algoritmo se encarga de reordenar los departamentos ya situados en los niveles garantizando que el espacio que ocupan los mismos no exceda el espacio máximo dispuesto por las dimensiones de la instalación.

Luego de haber logrado una buena distribución de departamentos o *“layout”* el algoritmo grafica el arreglo de cada uno de los niveles tanto al iniciar la tercera parte (cuando los departamentos están dispuestos de manera aleatoria en los niveles) como al finalizar la misma (cuando los centros de trabajo se han reorganizado).

El proceso para graficar el *“layout”* se lleva a cabo a través del uso de una secuencia o curva de llenado específica que recorre el espacio dispuesto para la ubicación de los departamentos dentro de cada nivel teniendo en cuenta las unidades de área que éstos requieren para su organización

Figura 12. Construcción de la distribución de planta a partir de la curva de llenado



Fuente: Meller & Bozer 1997

De la construcción del algoritmo se derivó el siguiente pseudocódigo que ejemplifica los pasos que sigue el algoritmo para dar solución al problema planteado:

5.1.4 Pseudocódigo para el algoritmo de solución para el problema FLP Multi-Floor.

1. Generar solución inicial de manera aleatoria para cada uno de los pisos.
2. Evaluar dicha solución en la Función Objetivo para obtener el valor de Costo Vertical.
3. Minimizar función de costo para encontrar una buena solución.

4. Para cada piso generar una solución inicial aleatoria para la organización de los departamentos en cada nivel.
5. Evaluar dicha solución en la Función Objetivo para obtener el valor de Costo Horizontal.
6. Minimizar función de costo para cada piso y encontrar una buena solución a la organización horizontal de departamentos.
7. Finalizar algoritmo.
8. Mostrar solución final.

6. VALIDACIÓN DEL ALGORITMO

Para la validación del algoritmo programado se hizo una evaluación comparativa del desempeño del mismo ante una instancia resuelta en la literatura a través de un método diferente. Esto se realizó con el objetivo de examinar la proximidad entre los resultados obtenidos en dicho problema resuelto en la literatura y el algoritmo desarrollado en el proyecto.

El artículo escogido para validar el algoritmo fue realizado por Russell D. Meller y Yavuz A. Bozer en 1997⁹¹ ya que éstos autores han sido pioneros en el estudio del problema FLP multifloor y gracias a sus publicaciones gozan de respeto a nivel científico en éste campo.

En el problema seleccionado se plantea una instalación de 3 pisos comunicados a través de 6 ascensores ubicados en diferentes puntos dentro del plano de la planta. Dentro de dicha instalación se fabrican 5 tipos de productos para los cuales se cuenta con 40 departamentos. Dentro del desarrollo del problema se especifica el flujo y carga de trabajo (en unidades) que cada producto representaba en los departamentos por los cuales debe pasar (Ver ANEXO N°1).

En el presente proyecto el proyecto se plantea dicha instancia de manera similar con la única diferencia de que sólo se considerará la existencia de un ascensor que abastece todos los niveles, los demás parámetros se mantienen.

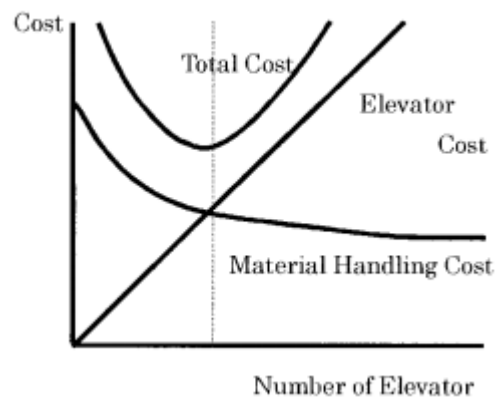
A partir de esto es de gran importancia resaltar la relación de proporcionalidad inversa que existe entre el número de ascensores dentro de la instalación y la cantidad de desplazamientos horizontales existentes, lo cual significa que entre

⁹¹ Russell D. Meller & Yavuz A. Bozer, Alternative approaches to solve multi-floor facility layout problem, Journal of Manufacturing Systems, 16(3), 192-203. 1997

menos elevadores se instalen dentro de la estructura se aumentará la cantidad de desplazamientos horizontales, lo cual deriva directamente en un incremento sustancial del costo total.

Dicha relación se puede apreciar claramente en la figura ¿?

Figura 13. Gráfico de relación entre el número de elevadores y el costo total.



Fuente: Kenichiro Matsuzaki, Takashi Irohara, Kuzuho Yoshimoto (1999)⁹²

Por lo mencionado anteriormente se espera que el problema planteado en el presente proyecto genere un costo mayor al asociado con el artículo de referencia.

Los autores plantean en su artículo un método de solución denominado STAGES el cual se basa en dos etapas de desarrollo. La primera etapa se encarga de distribuir los departamentos en los diferentes pisos de manera que el costo vertical se minimice. Para esto emplean una función objetivo cuadrática basada en la

⁹² Kenichiro Matsuzaki, Takashi Irohara & Kazuho Yoshimoto. Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization. Computers & Industrial Engineering. Vol 36. pp 487-502. 1999

formulación de Kaku, Thompson y Baybars⁹³ que solucionan a través de un algoritmo branch and bound.

En la segunda etapa del método se reordenan los departamentos asignados dentro de cada piso con el objetivo de minimizar el costo del desplazamiento horizontal. Para dar solución a esta etapa los investigadores aplican una versión modificada del algoritmo SABLE, propuesto por Meller y Bozer en 1996⁹⁴.

⁹³ Bharat Kaku, Gerald Tompson & Ilker Baybars. A Heuristic Method for the Multi-Story Layout Problem. *European Journal of Operational Research*. Vol 37(3). pp 384-397. 1988.

⁹⁴ Russell D. Meller & Yavuz A. Bozer. A new simulated annealing algorithm for the facility layout problem. *International journal of production research*. Vol 34(6). pp 1675.1692. 1996.

7. RESULTADOS

Dentro de la información obtenida a partir del algoritmo se encuentran: representación gráfica de los layouts de cada piso tanto en la etapa inicial, como en la etapa final; además se obtienen los gráficos de los costos iniciales y finales. (VER ANEXO B).

Con el fin de obtener datos acertados respecto a los valores de costo obtenidos mediante la aplicación del algoritmo, se debe tener una muestra significativa de datos para conocer los valores estadísticos de media y desviación estándar que permiten comparar los resultados plasmados en el artículo de referencia con los arrojados por el algoritmo. Para ello primero se toma una pre-muestra de $n=20$ con el fin de encontrar la media y la desviación estándar de la misma. Estos resultados se muestran en la Tabla N°1

Tabla 1. Valores de costo generados por el algoritmo considerando $n=20$

Costo Vertical Inicial	Costo Horizontal Inicial	Costo Vertical Final	Costo Horizontal Final
\$ 21,568.0	\$ 8,064.0	\$ 3,103.4	\$ 2,091.6
\$ 35,840.0	\$ 9,152.0	\$ 3,069.8	\$ 2,237.2
\$ 22,336.0	\$ 9,664.0	\$ 3,317.3	\$ 2,354.3
\$ 20,800.0	\$ 7,424.0	\$ 3,027.9	\$ 2,064.7
\$ 24,064.0	\$ 8,576.0	\$ 3,427.7	\$ 2,159.3
\$ 24,384.0	\$ 9,344.0	\$ 3,341.7	\$ 2,359.1
\$ 35,264.0	\$ 9,280.0	\$ 2,966.1	\$ 2,204.0
\$ 24,384.0	\$ 9,088.0	\$ 3,287.9	\$ 2,264.5
\$ 14,144.0	\$ 8,128.0	\$ 3,068.9	\$ 2,105.1
\$ 14,720.0	\$ 10,176.0	\$ 3,075.2	\$ 2,212.4
\$ 22,784.0	\$ 8,896.0	\$ 3,007.3	\$ 2,130.6
\$ 13,184.0	\$ 7,680.0	\$ 3,069.4	\$ 2,022.6
\$ 24,000.0	\$ 8,704.0	\$ 3,312.9	\$ 2,252.2

Costo Vertical Inicial	Costo Horizontal Inicial	Costo Vertical Final	Costo Horizontal Final
\$ 36,288.0	\$ 9,216.0	\$ 3,108.8	\$ 2,119.7
\$ 24,512.0	\$ 9,792.0	\$ 3,301.6	\$ 2,462.2
\$ 23,680.0	\$ 8,640.0	\$ 3,346.7	\$ 2,216.2
\$ 23,616.0	\$ 8,832.0	\$ 3,056.7	\$ 2,217.1
\$ 34,816.0	\$ 9,472.0	\$ 3,231.0	\$ 2,283.5
\$ 36,416.0	\$ 8,128.0	\$ 3,055.5	\$ 2,090.6
\$ 11,456.0	\$ 8,832.0	\$ 3,288.2	\$ 2,240.3

Los costos totales obtenidos en este paso se muestran en la siguiente tabla:

Tabla 2. Costos totales obtenidos de la muestra n=20

Costo Inicial Total	Costo Final Total
\$ 29,632.0	\$ 5,195.0
\$ 44,992.0	\$ 5,307.0
\$ 32,000.0	\$ 5,671.6
\$ 28,224.0	\$ 5,092.6
\$ 32,640.0	\$ 5,587.0
\$ 33,728.0	\$ 5,700.8
\$ 44,544.0	\$ 5,170.1
\$ 33,472.0	\$ 5,552.4
\$ 22,272.0	\$ 5,174.0
\$ 24,896.0	\$ 5,287.6
\$ 31,680.0	\$ 5,137.9
\$ 20,864.0	\$ 5,092.0
\$ 32,704.0	\$ 5,565.1
\$ 45,504.0	\$ 5,228.5
\$ 34,304.0	\$ 5,763.8
\$ 32,320.0	\$ 5,562.9
\$ 32,448.0	\$ 5,273.8
\$ 44,288.0	\$ 5,514.5
\$ 44,544.0	\$ 5,146.1

Costo Inicial Total	Costo Final Total
\$ 20,288.0	\$ 5,528.5

1. Se halla la media y la desviación estándar del Costo Total Final por medio de las siguientes fórmulas:

- Media: $X = \sum_{i=1}^n \frac{X_i}{n}$ $X = \$ 5377.56$

- Desviación Estándar: $s = \sqrt{\frac{\sum_{i=1}^n (X_i - X)^2}{n-1}}$ $S = 225.16$

2. Obtenidos los valores de media y desviación estándar de la muestra se procede a encontrar el número de iteraciones N que se deben realizar para establecer una muestra representativa de los resultados generados por el algoritmo, esto se hace con el fin de obtener resultados con una confianza del 95% y un error muestral o absoluto de \$ 50 (aproximadamente el 1% de la media)

$$N = \frac{Za^2 \times \sigma^2}{E^2} \quad N = 77.902 \approx 78$$

3. Se realizan las 78 iteraciones necesarias para analizar los datos de manera más confiable, obteniendo los siguientes resultados. Ver ANEXO N°4

4. A continuación se debe calcular nuevamente la media y la desviación estándar de las N iteraciones realizadas anteriormente:

- Media: $X = \sum_{i=1}^n \frac{X_i}{n}$ $X = \$ 5389.1$

- Desviación Estándar: $s = \sqrt{\frac{\sum_{i=1}^n (X_i - X)^2}{n-1}}$ $S = 239$

5. Se calcula el porcentaje de error de la media obtenida en el paso anterior respecto al Costo Total Final que obtuvieron los investigadores⁹⁵:

$$\% Error = \frac{V_2 - V_1}{V_1} \times 100$$

$$\% Error = -46.2\%$$

A partir del proceso anterior, se puede deducir que con los diseños de distribución de planta logrados mediante el código desarrollado en el presente proyecto de investigación, se logra reducir el costo en un 46.2% respecto al costo que se generaba con el código desarrollado por los autores del artículo. Esta premisa se puede afirmar con una confiabilidad del 95%, teniendo en cuenta un error de \$50.

En el artículo los autores presentan un costo anual de \$2.502 millones por año, es decir que mediante el código planteado anteriormente, este costo anual podría llegar a disminuir a una cifra aproximada de \$1.347 millones por año.

El resultado obtenido en el ejercicio de la validación del algoritmo es concluyente, ya que se evidencia una mejora incuestionable en el costo total al contrastar los datos alcanzados durante la aplicación del algoritmo desarrollado en este proyecto con los resultados logrados por los investigadores en su artículo, lo cual es pertinente resaltar.

Esta mejora tan significativa comprueba la efectividad del algoritmo propuesto en este proyecto, ya que aún cuando el marco de condiciones en las cuales se encontraba el problema de referencia era desfavorable en comparación a las condiciones establecidas en este trabajo respecto al número de elevadores existentes en la estructura por las razones que anteriormente se han mencionado y

⁹⁵ Russell D. Meller, Yavuz A. Bozer. Alternative Approaches to Solve the Multi-Floor Facility Layout Problem. Journal of Manufacturing Systems. Vol (16). pp192-203. 1997

referenciado, se alcanzó un resultado indiscutiblemente superior al obtenido en el problema encontrado en la literatura.

Además de reconocer la validez del algoritmo desarrollado en este proyecto, el resultado obtenido también pone en perspectiva el efecto que el avance tecnológico tiene sobre la calidad de las soluciones. El paso del tiempo sin duda ha alimentado la creación de equipos de computación con alta capacidad de procesamiento y precisión, los cuales logran que se obtengan mejores resultados a la hora de aplicar métodos de solución meta-heurísticos de manera más eficiente, es decir, en un tiempo menor.

8. CONCLUSIONES

- A la hora de aplicar un problema de distribución de planta a la vida real las restricciones asociadas a este aumentan al igual que su complejidad. Esto se realiza con el fin de que la situación planteada refleje de mejor manera las necesidades reales de las empresas. Este aumento en la complejidad del problema conlleva a que se haga necesario el uso de metaheurísticas como métodos alternativos de solución, con el fin de obtener una buena distribución de centros de trabajo o *layout* , y de esta manera, lograr el objetivo final que es minimizar los costos.
- Cuando el problema de distribución de planta que se busca resolver es de instancias pequeñas, la mejor opción es utilizar un método de solución exacto, ya que estos proporcionan una solución óptima al problema sin incrementar la complejidad del mismo.
- Los métodos de solución metaheurísticos son ideales para resolver problemas de distribución de planta cuyas características se asemejen a la realidad de las empresas. Esto se debe a que su formulación incrementa sustancialmente la complejidad del problema, imposibilitando la búsqueda de una solución óptima en el mismo. Los métodos metaheurísticos logran llegar a una buena solución del problema sin dejar de lado las restricciones asociadas al mismo.
- La formulación del problema es una de las etapas más importantes de este proceso de solución ya que en ella se deben analizar las características de la planta junto con su entorno y la manera como interactúan los procesos que se desarrollan en la misma para luego aplicar el método de solución de manera correcta.

- La distribución de planta tiene como objetivo hallar un orden de las áreas de trabajo que sea la más eficiente en costos y que vaya acorde a las estrategias de la empresa. Por ello, es un problema que debe ser evaluado y analizado detalladamente con el fin de que su solución sea aplicada a largo plazo ya que una distribución de planta no se puede estar modificando de manera continua.
- El tiempo de computo necesario para obtener una solución facilita la aplicación del algoritmo en un escenario de la vida real, ya que considerando los costos y recursos que se deberían destinar para llevar a cabo un proyecto de tal magnitud, la agilidad y eficacia del algoritmo desarrollado en este proyecto podría evitar despilfarros.
- El algoritmo desarrollado en este proyecto genera una mejor solución que el del artículo de referencia teniendo en cuenta que el primero considera un solo ascensor para el diseño, lo cual desencadena no sólo en una mejor respuesta, sino en una menor inversión a la hora de llevar el diseño a un escenario real, ya que el artículo de referencia consideraba seis ascensores.
- La aplicabilidad del algoritmo desarrollado en este proyecto se ve facilitada por la posibilidad de cambiar fácilmente los parámetros del mismo para adaptarlos a las condiciones requeridas por el diseño.

9. RECOMENDACIONES

- Con el fin de obtener una solución mejor y más real para el problema de distribución, se recomienda a los investigadores para un próximo proyecto tener en cuenta otras variaciones del problema que pueden incluir: múltiple número de elevadores, elevadores con capacidad restringida, la inclusión de estructuras internas tales como columnas y pasillos, entre otros.
- Se recomienda a los estudiantes próximos a desarrollar un proyecto relacionado con el problema de distribución de planta, profundizar en los temas vistos en la clase de Diseño de Sistemas Productivos con el fin de adquirir buenas bases para abordar la investigación.
- El apoyo y promoción que se realice a los proyectos de grado enfocados en investigación es vital para la construcción de conocimiento dentro de la comunidad científica y la sociedad estudiantil, enriqueciendo la profesión del ingeniero industrial y la rama investigativa de la universidad.
- Lograr que las pequeñas y medianas empresas de la región apoyen los proyectos de investigación realizados por los estudiantes de ingeniería industrial de la UIS facilitaría un escenario propicio para que ambas partes generen valor al beneficiarse del mutuo apoyo. Los estudiantes y sus proyectos podrían financiarse y aplicarse con mayor facilidad, y de igual manera las empresas se nutrirían de los hallazgos obtenidos en los proyectos.

TABLA DE CUMPLIMIENTO DE OBJETIVOS

OBJETIVO	CUMPLIMIENTO
Realizar una revisión bibliográfica del problema de distribución de planta (FLP) de múltiples niveles, un elevador y departamentos de áreas desiguales, incluyendo planteamientos y enfoques aplicados al problema.	2. 3.
Desarrollar un algoritmo para el problema de distribución de planta (FLP) de múltiples niveles, un elevador y departamentos de áreas desiguales.	5.1.1 5.1.2 5.1.3
Programar el algoritmo en un software computacional y realizar los ajustes correspondientes al diseño del programa.	5.1.4
Evaluar la efectividad del algoritmo, mediante la solución de problemas encontrados en la literatura.	6. 7.

BIBLIOGRAFÍA

KONAK, A., KULTUREL-KONAK, S., NORMAN, B., SMITH, A. E. A new mixed integer programming formulation for facility layout design using flexible bays. *Operations Research Letters*. Vol 34(6). Pp 660-672. 2005

KIEVANI A., RAFIENEJAD, S., KAVIANI, M., & AFSHARI, H. A simulated annealing for multi floor facility layout problem En: *The world congress of engineering and computer science* (2010, Octubre 20-22, San Francisco, USA), p 1-2.

DRIRA, A., PIERREEVAL, H., HAJRI-GABOUJ, S. Facility layout problems: A survey. *Annual reviews in control*. pp 255-267. 2007.

DERAKHSHA A., & WONG, K. Y. Solving unequal-area static and dynamic facility layout problems using modified particle swarm optimization. *Manufacturing and Industrial Engineering Journal*. 2015

ADIL, B., TURKAY, D., IBRAHIM, S. An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. *Omega* 34, pp.385-396. 2006

MONTREUIL, B. Integrating design of cell layout, input/output I/O point configuration, and flow network of manufacturing systems. En: Liu, C.R., Requicha, A. y Chandrasekar, S. *Intelligent and Integrated Manufacturing Analysis and Synthesis*. Ped-vol 25, ASME: New York, 315-326. 1987

KAKU, B. K., & THOMPSON, G. L. An exact algorithm for the general quadratic assignment problem. *European Journal of Operational Research*, 23, 382–390. 1986

NORMAN, B., KEITH, L., ROBIN, W. *Graph Theory 1736-1936*. Oxford University Press. 1986

NAJMEH, B., MOSTAFA, A., & MOHSEN, Z. Tabu search heuristic for efficiency of dynamic facility layout problem. *Manufacture Technology*, 77, pp.689-703. 2015

GODSIL, C., & ROYLE, G. *Algebraic Graph Theory*. New York: Springer-Verlag. 2001

CHWIF, L., PEREIRA, M., P, MOSCATO, L. A. A solution to the facility layout problem using simulated annealing. *Computers in Industry*, 36, pp.125-132. 1998

VALENCIA, C., J. Rediseño del sistema productivo utilizando técnicas de distribución de planta: Caso de estudio planta procesadora de alimentos. Manizales. Trabajo de Grado (Maestría en Ingeniería Industrial). Universidad Nacional de Colombia. Facultad de ingeniería y arquitectura. 2013.

BINGHAM, D. Virtual Library of Simulation Experiments: Test Functions and Databases. Bohachevsky Functions. Simon Fraser University. 2013. Recuperado de <https://www.sfu.ca>.

SIMMONS, D. One-dimensional space allocation: An ordering algorithm. *Operations Research*. Vol 17. 812-826. 1969

DRIRA, A., PIERREEVAL, H., HAJRI-GABOUJ, S. Facility layout problems: A survey. *Annual reviews in control*. pp 255-267. 2007.

RODRIGES, E., RODRIGUES, A., GOMES, G. An approach to the multi-level space allocation problem in architecture using a hybrid evolutionary technique. *Automaton in construction*. Vol 35. Pp 482-498. 2014.

AI-KHAYYAL, F., GOETSCHALCKX, M., & VAN VOORHIS, T. Alternative formulations and new solutions for the facility layout problem. *Reporte Técnico, Escuela de Ingeniería Industrial y Sistemas de Ingeniería, Georgia Institute of Technology, Atlanta, Georgia*. 2001

RAMTIN, F., ABOLHASANPOUR, M., HOJABRI, H., HEMMATI, A., & JAAFARI, A. Optimal Multi Floor Facility Layout. *International MultiConference of Engineers and Computer Scientists*. Pages 1797-1800. Marzo 2010

GARCÍA, A. Solución del problema dinámico estocástico de la distribución de planta mediante el uso de un algoritmo genético. Bogotá. Trabajo de Grado (Ingeniería Industrial). Pontificia Universidad Javeriana. Facultad de Ingenierías, Escuela de Ingeniería Industrial. 2013

GEUNG-CHOL, L., & YEONG-DAE, K. Algorithms for adjusting shapes of departments in block layouts on the grid-based plane. *Omega*. 28. pp 111-122. 2000.

GOMEZ, A., DE LA FUENTE, I., PUENTE, D. Using genetic algorithms to resolve layout problems in facilities where there are aisles. *Int. J. Production Economics*, 84, pp.271-282. 2003

ARMOUR, G., & BUFFA, E. A heuristic algorithm and simulation approach to relative location of facilities. *Management Sciences*. 9(2). pp 294-309

YASMINA, H., LIONEL, A., YALANOUI, F., & CHEN, H. Ant colony optimization for solving an industrial layout problem. *European Journal of Operational Research*, 183, pp.633-642. 2007

SHERALI, H. D., FRATICELLI, B. M., & MELLER, R. D. Enhanced Model Formulations for Optimal Facility Layout, *Operations Research*, 51(4), 629-644. 2003

SUNDERESH, H., & ATTAHIRU, A. Experimental analysis of simulated annealing based algorithms for the layout problem, *European journal of operational research*, 57, 1992, pp 190-202.

JAE-GON, K., & YEONG-DAE, K. Layout planning for facilities with fixed shapes and input output points. *International Journal of Production Research*. 38(18). pp 4635-4653. 2000

JAJODIA, S., MINIS, I., HARHALAKIS, G. & PROTH, J. Class: computerized layout solutions using simulates annealing, *International journal of production research*, 30, 1, 1993, pp 95-108.

B. DILWORTH, J. *Operation management*. McGraw Hill. 1996.

ZHOU, J., *Algorithmes et outils pour l'analyse des flux de production a` l'aide du concept d'ordre*. Ph.D. Disertación doctoral (en Francés). Universidad de Estrasburgo. 1998

BARTHOLDI, J. J., & PLATZMAN, L. K. An $O(N \log N)$ planar traveling salesman heuristic based on spacefilling curves. *Operations research letters*. Vol 1(4). Pp121-125. 1982.

HARTMANIS, J., & STEARNS, R. On the computational complexity of algorithms. Transactions of the American Mathematical Society. Vol 117. Pp 285-306. American Mathematical Society. 1965

JAE-GON, K., & YEONG-DAE, K. Graph theoretic heuristics for unequal-sized facility layout problems. Omega. Vol 218(2), p382-391. 1995.

MATSUZAKI, K., IROHARA, T., & YOSHIMOTO, K. Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization. Computers & Industrial Engineering. Vol 36. pp 487-502. 1999

MATSUZAKI, K., IROHARA, T., YOSHIMOTO, K. Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization. Computers & Industrial Engineering, 36(2), 487–502. 1999.

KIRKPATRICK, S., GELATT, D., VECCHI, M. Optimization by simulated annealing, Science, 220, pp 671-680. 1983

ANIRBAN, K., & PRANAB, K. D. The scope of genetic algorithms in dealing with facility layout problems. *South African Journal of Industrial Engineering*, 21, pp.39-49. 1998

LEE, K., ROH, M., JEONG, H. An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. Computers & Operations Research, 32(4), 879–899. 2005

AL-HAKIM, L. A. Two graph-theoretic procedures for an improved solution for the facilities layout problem. International journal of production research. Vol 29, N°8. Pp 1701-1718. 1991.

CHWIF, L., PEREIRA, M. R., MOSCATO, L. A. A solution to the facility layout problem using simulated annealing. *Computers in Industry*, 36(1–2), 125–132. 1998

LIANG, L. Y. & CHAO, W. C. The strategies of tabu search technique for facility layout optimization. *Automation in Construction*, 17, pp.657-669. 2008

SOLIMANPURA, M., JAFARIB, A. Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers & Industrial Engineering*, 55, 606-619. 2008

SARAVANAN, M. S., & KUMAR, S. Different approaches for the loop layout problems: a review. *Manufacturing Technology Journal*, 69, 2513–2529. 2013

GOETSCHALCKX, M., & JIN-KUN, K. An integrated algorithm for the design of block layouts with regularized material handling flow paths, *Progress in Material Handling Research*: 171-185. 2006

MISOLA, M., & NAVARRO, B. Optimal Facility Layout Problem Solution Using Genetic Algorithm. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 7, 616-621. 2015, mayo, De World Academy of Science, Engineering and Technology Base de datos. 2013

MATSUZAKI, K., IROHARA, T., YOSHIMOTO, K. Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization. *Computers & Industrial Engineering*, 36(2), 487-502. 1999.

MCKENDALL, A., SHANG, J., & KUPPUSAMY, S. Simulated annealing heuristics for the dynamic facility layout problem. *Computers & Operations Research*, 33, pp.2431-2444. 2005

MEJÍA, M., GARZÓN, D. A., & ARROYO, J. Métodos discretos basados en quimiotaxis de bacterias y algoritmos genéticos para solucionar el problema de distribución de planta en celdas de manufactura. *Ciencia e ingeniería neogranadina*. 24 (1). Pp 6-28. 2014.

MELLER, R. D., GAU, K. The Facility Layout Problem: Recent and Emerging Trends and Perspectives. *Journal of Manufacturing Systems*, 15, 351-366. 2015, Junio. 1996

WANG, M., Meei-Yuh, M. H. A solution to the unequal area facilities layout problem by genetic algorithm. *Computers in industry*. Vol 56. Pp 207-220. 2005

HASSAN, M., & HOGG, G. A review of graph theory applications to the facilities layout problem. *Omega* 15:291–300. 1987

HASSAN, M. Machine layout problem in modern manufacturing facilities. *International Journal of Production Research*, 32(11), 2559– 2584. 1994

BAZARAA, M., & SHERALI, H. Benders' partitioning scheme applied to a new formulation of quadratic assignment problem. *Naval Research Logistics Quarterly*, 27(1), 29–41. 1980

MONTREUIL, B. Modeling framework for integrating layout design and flow network design. In: *Proceedings of the material handling research colloquium*, Hebron, KY, pp 43–58 1990

ÁLVAREZ, A., NOBLE, A., LÓPEZ., V Optimización multiobjetivo del problema de distribución de planta: Un nuevo modelo matemático. En *Ingeniería y Competitividad*, 16, N°2, pp 257-267.

IZADINIA, N., ESHGHI, K., & HASSAN, M. A robust model for multi floor optimization. *Computers & Industrial Engineering*. Vol 78. Pp 127-134. 2014

IZADINIA, N., & ESHGHIA, K. A robust mathematical model and ACO solution for multi-floor discrete layout problem with uncertain locations and demands. *Computers and Industrial Engineering*. 2016

PALACIOS, M., & JAIME, C. Alternativa de solución al problema de distribución de planta (flp) para instalaciones de áreas iguales y desiguales mediante un algoritmo híbrido genético. Bucaramanga. Trabajo de Grado (Ingeniera Industrial). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería Industrial. 2011.

BURKARD, R., & BÖNNIGER, T. A heuristic for quadratic Boolean program with application to quadratic assignment problems. *European Journal of operational Research*, 13, 347–386. 1983

KIA, R., KHAKSAR-HAGHANI, F., JAVADIAN, N., & TAVAKKOLI, R. Solving a multi-floor layout design model of a dynamic cellular manufacturing system by an efficient genetic algorithm. *Journal of manufacturing systems*. Vol 33(1). Pp 218-232. 2014.

RIOJAS, A. Conceptos, algoritmo y aplicación al problema de las N-reinas. Trabajo de grado Licenciada en investigación operativa. Lima, Perú: Universidad Nacional Mayor de San Marcos. Facultad de ciencias matemáticas. Pregrado en licenciatura en investigación de operaciones y sistemas. 2005. p32-50.

LOVE, R., & WONG, J. On solving a one-dimensional space allocation problem with integer programming. *INFOR*. Vol 4(2). p139-143. 1976

LOVE, R., & WONG, J. Solving quadratic assignment problems with rectangular distances and integer programming. *Naval Research Logistics Quarterly*, 23, 623–627. 1976

JHONSON, R. SPACECRAFT for multi-floor layout planning. *Management Sciences*, 28(4), 407–417. 1982

RUEDA, E., & VILLAMIL, G. Alternativa de solución al problema de distribución de planta (flp) de un solo nivel con estructuras internas, muros y pasajes, a través de métodos metaheurísticos. Bucaramanga. Trabajo de Grado (Ingeniera Industrial). Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería Industrial. 2014.

MELLER, R., GAU, K. The Facility Layout Problem: Recent and Emerging Trends and Perspectives, *Journal of Manufacturing Systems*, Vol. 15, 1997

MELLER, R., NARAYANANA, V., VANCE, P. Optimal Facility Layout Design. *Operations Research Letters*, 23, 117-127. 1999

MELLER, R., GAU, K. “The Facility Layout Problem: Recent and Emerging Trends and Perspectives”. *Journal Of Manufacturing Systems*, 15(5), 351-366. 1997

MELLER, R., & BOZER, Y. Alternative approaches to solve multi-floor facility layout problem, *Journal of Manufacturing Systems*, 16(3), 192-203. 1997

MELLER, R., & BOZER, Y. A new simulated annealing algorithm for the facility layout problem. *International journal of production research*. Vol 34(6). pp 1675.1692. 1996.

MELLER, R., & BOZER, Y. Alternative Approaches to Solve the Multi-Floor Facility Layout Problem. *Journal of Manufacturing Systems*. Vol (16). pp192-203. 1997

MELLER, R., & BOZER, Y. Alternative approaches to solve the multi-floor facility layout problem. *Journal of Manufacturing Systems*. Vol 16 (3). P192-203. 1997.

SALAZAR-HORNING, E., & MEDINA, J. Minimización del makespan en máquinas paralelas idénticas con tiempos de preparación dependientes de la secuencia utilizando un algoritmo genético. *Ingeniería Investigación y Tecnología*, 14, pp.43-51. 2012

JAJODIA, S., MINIS, I., HARHALAKIS, G., & PROTH, J. CLASS: Computerized layout solutions using simulated annealing. *International journal of production research*. Vol30(1). pp 95-108. 1992.

HOSSEINI, S., MIRZAPOUR, S., & WONG K. Y. Improving multi-floor facility layout problems using systematic layout planning and simulation. *Communications in computer and information science*. Advances in information technology. Vol 409. Pp 58-69. 2013

ABDINNOUR-HELM, S., & HADLEY, S. Tabu search based heuristics for multi-floor layout, *International Journal of Production Research*, 38(2), 365-383. 2000

SINGH, S., & SHARMA, R. A review of different approaches to the facility layout problems. *The International Journey of Advanced Manufacturing Technology*. 5-6. p 425-433. 2005

C. KOOPMANS, T & BECKMAN, M. Assignment problems and the location of economic activities. *Econometrica* 25:53–76. 1957

AYBARS, U., & DOĞAN, A. An interactive simulation and analysis software for solving TSP using Ant Colony Optimization algorithms. *Advances in Engineering Software*, 40, pp.341-349. 2009

CHEN, Y., XIAO, Q., & TANG, X. Product layout optimization and simulation model in a multi-level distribution center. *Systems engineering procedia 2*. Vol 2. Pp 300-307. 2011

MELLER, R., & BOZER, Y. & ERLEBACHER, S. An Improvement-Type Layout Algorithm for Single and Multiple Floor Facilities, *Management Science*, 40(7), 918-932. 1994.

LEE, Y. H., & LEE, M. H. A shape-based block layout approach to the facility layout problems using a hybrid genetic algorithm. *Computers & Industrial Engineering*. 42. pp 237-248. 2002.

ANEXOS

ANEXO A. Valores de áreas para cada departamento

Dpto.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Área	8	16	12	12	16	8	8	16	4	16	16	4	16	16	8	8	4	4	4	4
Dpto.	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Área	16	16	8	8	4	4	8	16	12	4	4	16	16	4	4	4	4	8	4	32

Matriz de flujo para el problema

Producto	Ensamblados	Dpto. 1	Dpto. 2	Dpto. 3	Dpto. 4	Dpto. 5	Dpto. 6	Dpto. 7	Dpto. 8	Dpto. 9	Producción (Unid/día)
1	1	40(250)	1(100)	9(100)	10(250)	11(250)	2(500)	40			18400
1	1	40(250)	1(200)	3(200)	4(100)	10					
1	1	40(100)	1(200)	5(100)	11						
1	2	40(2000)	1(400)	5(200)	6(400)	11					
2	1	40(250)	1(250)	12(500)	13(100)	14(200)	15(250)	2(500)	40		4600
2	1	40(250)	16(100)	17(200)	13						
2	1	40(250)	18(200)	19(200)	13						
2	1	40(500)	1(200)	3(200)	4(200)	14					
2	1	40(500)	1(100)	5(100)	15						
3	1	40(250)	20(100)	21(100)	22(200)	23(200)	8(100)	40			6900
3	1	40(200)	7(100)	24(200)	25(200)	22					
3	1	40(250)	1(200)	3(100)	4(200)	25					
3	1	40(250)	1(100)	5(250)	23						
3	0.5	40(250)	1(50)	5(25)	6(62.5)	23					
4	1	40(250)	26(100)	27(200)	28(200)	29(200)	8(100)	2(200)	40		8050
4	1	40(200)	1(200)	3(100)	4(100)	28					
4	2	40(2000)	1(400)	5(400)	6(200)	29					

Producto	Ensamblés	Dpto. 1	Dpto. 2	Dpto. 3	Dpto. 4	Dpto. 5	Dpto. 6	Dpto. 7	Dpto. 8	Dpto. 9	Producción (Unid/día)
5	1	1(100)	40(500)	30(200)	31(200)	32(100)	33(100)	34(200)	2(500)	40	11500
5	1	1(200)	40(200)	36(250)	37(100)	32					
5	1	1(250)	40(250)	38(100)	39(100)	33					
5	1	1(100)	40(250)	3(200)	4(100)	33					
5	1	1(200)	40(1000)	5(100)	34						

ANEXO B.

```
% v2.0 %
clc
clear all
close all
% Internal variables
N=40; %numero de departamentos
Amax=140; % Area maxima permitida por piso
lift=[11 1]; % posiciones del ascensor
n_cycles = 10;
n_tries_for_cycle = 10;
target_function = 'bohal';
n_dimensions = 2;
% Probabilities of accept the worst solution
initial_probability = 0.7;
final_probability = 0.001;
% Calculate temperature and steps
initial_temperature = -1 / log(initial_probability);
final_temperature = -1 / log(final_probability);
reduction_fraction = (final_temperature / initial_temperature)^(1/
(n_cycles-1));
% Initial position
% Generate random number in the range of each dimension
range_min=1;
range_max=40;
v0 = floor(4*(rand(n_dimensions,1)))
%
*****
% print initial values
sprintf('initial_temperature = %f', initial_temperature)
sprintf('final_temperature = %f', final_temperature)
sprintf('reduction_fraction = %f', reduction_fraction)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% Calculate error for initial point%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%%% Se establecen las unidades de carga f_1, f_2, \dots, f_{20} y el flujo de cada producto C_1, C_2, \dots, C_{20}

```
f1=[250 100 100 250 250 500];  
2  
f2=[250 200 200 100];  
f3=[100 200 100];  
f4=[2000 400 200 400];  
f5=[250 250 500 100 200 250 500];  
f6=[250 100 200];  
f7=[250 200 200];  
f8=[500 200 200 200];  
f9=[500 100 100];  
f10=[250 100 100 200 200 100];  
f11=[200 100 200 200];  
f12=[250 200 100 200];  
f13=[250 100 250];  
f14=[250 50 25 62.5];  
f15=[250 100 200 200 200 100 200];  
f16=[200 200 100 100];  
f17=[2000 400 400 200];  
f18=[500 100 200 200 100 100 200  
500];  
f19=[200 200 250 100];  
f20=[250 250 100 100];  
f21=[250 100 200 100];  
f22=[1000 200 100];  
C1=[40 1 9 10 11 2];  
C2=[40 1 3 4];  
C3=[40 1 5];  
C4=[40 1 5 6];  
C5=[40 1 12 13 14 15 2];  
C6=[40 16 17];  
C7=[40 18 19];  
C8=[40 1 3 4];  
C9=[40 1 5];  
C10=[40 20 21 22 23 8];
```

```

C11=[40 7 24 25];
C12=[40 1 3 4];
C13=[40 1 5];
C14=[40 1 5];
C15=[40 26 27 28 29 8 2];
C16=[40 1 3 4];
C17=[40 1 5 6];
C18=[40 1 30 31 32 33 34 2];
C19=[40 1 36 37];
C20=[40 1 38 39];
C21=[40 1 3 4];
3
C22=[40 1 5];
% Tabla sobre la respectiva area de cada departamento.
A=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40;
8 16 12 12 16 8 8 16 4 16 16 4 16 16 8 8 4 4 4 4 16 16 8 8 4 4 8 16 12
4 4 16 16 4 4 4 4 8 4 32];
v0 =
3
3
ans =
initial_temperature = 2.803673
ans =
final_temperature = 0.144765
ans =
reduction_fraction = 0.719437

```

Generación del punto de partida

```

while 2>1
A1=A(2,40);
A2=0;
A3=0;
dept1=40;
dept2=0;
dept3=0;

```

```

X0=randperm(39);
for i=1:N-1
v0 = floor(3*(rand(1,1)))+1;
if v0==1
dept1=[dept1 X0(i)];
A1=A(2,X0(i))+A1;
else if v0==2
dept2=[dept2 X0(i)];
A2=A(2,X0(i))+A2;
else
4
dept3=[dept3 X0(i)];
A3=A(2,X0(i))+A3;
end
end
end
if (A1<=Amax && A2<=Amax && A3<=Amax)
break;
end
end
%dept1=[1,3,4,9,10,12,28,30,31,32,33,36,37,38,39,40];
%dept2=[0 2, 5 6,7 ,8,11, 15,23,24,25,26,27,29,34,35];
%dept3=[0 13,14,16,17,19,20,21,22];
error1=0;
error0=0;
sigma=16; % distancia entre pisos adyacentes
Cv=0.08; % factor de costo vertical
Ch=0.016; % factor de costo Horizontal
No_f=22; % numero de flujos establecidos en todo los productos
for k=1:No_f
for i=1:length(eval(strcat('C',num2str(k))))-1
Dk=0;
Dg=0;
for j=1:length(dept1)
if
dept1((j))==eval(strcat('C',num2str(k),'(',num2str(i),')'))

```

```

Dk=1;
end
if dept1(j)==(eval(strcat('C',num2str(k),'(',num2str(i
+1),')')))
Dg=1;
end
end
for j=2:length(dept2)
if
dept2(j)==(eval(strcat('C',num2str(k),'(',num2str(i),')')))
Dk=2;
end
5
if dept2(j)==(eval(strcat('C',num2str(k),'(',num2str(i
+1),')')))
Dg=2;
end
end
for j=2:length(dept3)
if
dept3(j)==(eval(strcat('C',num2str(k),'(',num2str(i),')')))
Dk=3;
end
if dept3(j)==(eval(strcat('C',num2str(k),'(',num2str(i
+1),')')))
Dg=3;
end
end
error0=abs(Dg-
Dk)*sigma*Cv*(eval(strcat('f',num2str(k),'(',num2str(i),')')))
+error0;
end
end
dept1=sort(dept1);
dept2=sort(dept2(2:end));
dept3=sort(dept3(2:end));
disp('Departamentos en el piso 1:');

```

```

disp(dept1)
disp('Departamentos en el piso 2:');
disp(dept2)
disp('Departamentos en el piso 3:');
disp(dept3)
disp('Funcion de costo vertical inicial:');
disp(error0)
cont=1;
vector(cont)=error0;
Departamentos en el piso 1:
3 4 10 14 19 26 32 33 35 36 39
40
6
Departamentos en el piso 2:
1 2 5 9 12 13 20 22 24 27 30
34 37
Departamentos en el piso 3:
Columns 1 through 13
6 7 8 11 15 16 17 18 21 23 25
28 29
Columns 14 through 15
31 38
Funcion de costo vertical inicial:
21568
% Iterate over temperature
X(1)=initial_temperature;
cont=2;
temperature=initial_temperature;
while (temperature>=final_temperature)
temperature=temperature*reduction_fraction;
X(cont)=temperature;
cont=cont+1;
end;
for temperature=X
sprintf('Temperature = %f', temperature);
% Iterations for each temperature

```

```

for n=1:n_tries_for_cicle
% Generate new point of exploration
while 2>1
A1=A(2,40);
A2=0;
A3=0;
dept1_f=40;
dept2_f=0;
dept3_f=0;
X1=randperm(39);
for i=1:N-1
v0 = floor(3*(rand(1,1)))+1;
if v0==1
dept1_f=[dept1_f X1(i)];
7
A1=A(2,X1(i))+A1;
else if v0==2
dept2_f=[dept2_f X1(i)];
A2=A(2,X1(i))+A2;
else
dept3_f=[dept3_f X1(i)];
A3=A(2,X1(i))+A3;
end
end
end
if (A1<=Amax && A2<=Amax && A3<=Amax)
break;
end
end
for k=1:No_f
for i=1:length(eval(strcat('C',num2str(k))))-1
Dk=0;
Dg=0;
for j=1:length(dept1_f)
if
dept1_f((j))==eval(strcat('C',num2str(k),'(',num2str(i),')'))

```

```

Dk=1;
end
if dept1_f((j))==eval(strcat('C',num2str(k),'(',num2str(i
+1),')'))
Dg=1;
end
end
for j=2:length(dept2_f)
if
dept2_f((j))==eval(strcat('C',num2str(k),'(',num2str(i),')'))
Dk=2;
end
if dept2_f((j))==eval(strcat('C',num2str(k),'(',num2str(i
+1),')'))
Dg=2;
end
8
end
for j=2:length(dept3_f)
if
dept3_f((j))==eval(strcat('C',num2str(k),'(',num2str(i),')'))
Dk=3;
end
if dept3_f((j))==eval(strcat('C',num2str(k),'(',num2str(i
+1),')'))
Dg=3;
end
end
error1=abs(Dg-
Dk)*sigma*Cv*(eval(strcat('f',num2str(k),'(',num2str(i),')')))+error1;
end
end
delta_error = error1 - error0;
if (delta_error <0)
for i=1:n_dimensions
dept1 = dept1_f;

```

```

dept2 = dept2_f;
dept3 = dept3_f;
end
error0 = error1;
else
p = exp(-(delta_error/temperature));
r = rand;
if (p>r)
for i=1:n_dimensions
dept1 = dept1_f;
dept2 = dept2_f;
dept3 = dept3_f;
end
error0 = error1;
end
end
disp('-----')
dept1=sort(dept1);
dept2=sort(dept2);
dept3=sort(dept3);
disp('Departamentos en el piso 1:');
9
disp(dept1)
disp('Departamentos en el piso 2:');
disp(dept2(2:end))
disp('Departamentos en el piso 3:');
disp(dept3(2:end))
error1=0;
disp('error0:');
disp(error0)
vector=[vector error0];
cont=cont+1;
disp('-----')
end
end
disp(' -----')

```

```

disp(' resultado -----')
dept1=sort(dept1)
dept2=sort(dept2(2:end))
dept3=sort(dept3(2:end))
disp('Departamentos en el piso 1:');
disp(dept1)
disp('Departamentos en el piso 2:');
disp(dept2(1:end))
disp('Departamentos en el piso 3:');
disp(dept3(1:end))
disp('Funcion de costo vertical final:');
disp(error0)
figure, plot(vector)
title('Funcion Costo - Primera Etapa');
Ac=A;
% dept1=[1,3,4,9,10,12,28,30,31,32,33,36,37,38,39,40];
% dept2=[2, 5 6,7 ,8,11, 15,23,24,25,26,27,29,34,35];
% dept2=[24,23,11,5,6,7,25,34,8,2,26,35,29,15,27];
% dept3=[19,13,14,18,17,16,21,20,22];
% dept3=[13,14,16,17,19,20,21,22];
-----
Departamentos en el piso 1:
10
3 4 10 14 19 26 32 33 35 36 39
40
Departamentos en el piso 2:
2 5 9 12 13 20 22 24 27 30 34
37
Departamentos en el piso 3:
Columns 1 through 13
7 8 11 15 16 17 18 21 23 25 28
29 31
Column 14
38
error0:
21568

```


Departamentos en el piso 1:
3 4 10 14 19 26 32 33 35 36 39
40
Departamentos en el piso 2:
2 5 9 12 13 20 22 24 27 30 34
37
Departamentos en el piso 3:
Columns 1 through 13
7 8 11 15 16 17 18 21 23 25 28
29 31
Column 14
38
error0:
21568

Departamentos en el piso 1:
3 4 10 14 19 26 32 33 35 36 39
40
Departamentos en el piso 2:
2 5 9 12 13 20 22 24 27 30 34
37
Departamentos en el piso 3:
11
Columns 1 through 13
7 8 11 15 16 17 18 21 23 25 28
29 31
Column 14
38
error0:
21568

Departamentos en el piso 1:

1 3 7 8 10 13 17 25 28 34 35
40

Departamentos en el piso 2:

Columns 1 through 13

11 12 14 16 19 20 21 22 23 26 27
30 33

Column 14

37

Departamentos en el piso 3:

Columns 1 through 13

2 4 5 6 9 15 18 24 29 31 32
36 38

Column 14

39

error0:

15424

Departamentos en el piso 1:

1 3 7 8 10 13 17 25 28 34 35
40

Departamentos en el piso 2:

Columns 1 through 13

11 12 14 16 19 20 21 22 23 26 27
30 33

Column 14

12

37

Departamentos en el piso 3:

Columns 1 through 13

2 4 5 6 9 15 18 24 29 31 32
36 38

Column 14

39

error0:

15424

Departamentos en el piso 1:

1 3 7 8 10 13 17 25 28 34 35

40

Departamentos en el piso 2:

Columns 1 through 13

11 12 14 16 19 20 21 22 23 26 27

30 33

Column 14

37

Departamentos en el piso 3:

Columns 1 through 13

2 4 5 6 9 15 18 24 29 31 32

36 38

Column 14

39

error0:

15424

Departamentos en el piso 1:

1 3 7 8 10 13 17 25 28 34 35

40

Departamentos en el piso 2:

Columns 1 through 13

13

11 12 14 16 19 20 21 22 23 26 27

30 33

Column 14

37

Departamentos en el piso 3:

Columns 1 through 13

2 4 5 6 9 15 18 24 29 31 32

36 38

Column 14

39
error0:
15424

Departamentos en el piso 1:
1 3 7 8 10 13 17 25 28 34 35
40
Departamentos en el piso 2:
Columns 1 through 13
11 12 14 16 19 20 21 22 23 26 27
30 33
Column 14
37
Departamentos en el piso 3:
Columns 1 through 13
2 4 5 6 9 15 18 24 29 31 32
36 38
Column 14
39
error0:
15424

Departamentos en el piso 1:
1 3 7 8 10 13 17 25 28 34 35
40
14
Departamentos en el piso 2:
Columns 1 through 13
11 12 14 16 19 20 21 22 23 26 27
30 33
Column 14
37
Departamentos en el piso 3:
Columns 1 through 13

2 4 5 6 9 15 18 24 29 31 32

36 38

Column 14

39

error0:

15424

Departamentos en el piso 1:

1 3 7 8 10 13 17 25 28 34 35

40

Departamentos en el piso 2:

Columns 1 through 13

11 12 14 16 19 20 21 22 23 26 27

30 33

Column 14

37

Departamentos en el piso 3:

Columns 1 through 13

2 4 5 6 9 15 18 24 29 31 32

36 38

Column 14

39

error0:

15424

15

Departamentos en el piso 1:

1 3 7 8 10 13 17 25 28 34 35

40

Departamentos en el piso 2:

Columns 1 through 13

11 12 14 16 19 20 21 22 23 26 27

30 33

Column 14

37

Departamentos en el piso 3:

Columns 1 through 13

2 4 5 6 9 15 18 24 29 31 32

36 38

Column 14

39

error0:

15424

Departamentos en el piso 1:

1 3 7 8 10 13 17 25 28 34 35

40

Departamentos en el piso 2:

Columns 1 through 13

11 12 14 16 19 20 21 22 23 26 27

30 33

Column 14

37

Departamentos en el piso 3:

Columns 1 through 13

2 4 5 6 9 15 18 24 29 31 32

36 38

Column 14

39

16

error0:

15424

Departamentos en el piso 1:

Columns 1 through 13

1 3 6 9 20 22 23 24 27 29 30

34 36

Columns 14 through 15

39 40

Departamentos en el piso 2:

2 7 8 10 11 12 13 14 15 16 25

31 35

Departamentos en el piso 3:

4 5 17 18 19 21 26 28 32 33 37

38

error0:

14976

Departamentos en el piso 1:

Columns 1 through 13

1 4 8 9 11 12 14 17 18 23 26

34 38

Column 14

40

Departamentos en el piso 2:

Columns 1 through 13

3 6 7 13 16 19 20 22 25 27 28

30 31

Columns 14 through 16

35 36 37

Departamentos en el piso 3:

2 5 10 15 21 24 29 32 33 39

error0:

14464

17

Departamentos en el piso 1:

Columns 1 through 13

1 4 8 9 11 12 14 17 18 23 26

34 38

Column 14

40

Departamentos en el piso 2:
Columns 1 through 13
3 6 7 13 16 19 20 22 25 27 28
30 31
Columns 14 through 16
35 36 37
Departamentos en el piso 3:
2 5 10 15 21 24 29 32 33 39
error0:
14464

Departamentos en el piso 1:
Columns 1 through 13
1 4 8 9 11 12 14 17 18 23 26
34 38
Column 14
40

Departamentos en el piso 2:
Columns 1 through 13
3 6 7 13 16 19 20 22 25 27 28
30 31
Columns 14 through 16
35 36 37
Departamentos en el piso 3:
2 5 10 15 21 24 29 32 33 39
error0:
18
14464

Departamentos en el piso 1:
Columns 1 through 13
1 4 8 9 11 12 14 17 18 23 26
34 38
Column 14

40

Departamentos en el piso 2:

Columns 1 through 13

3 6 7 13 16 19 20 22 25 27 28

30 31

Columns 14 through 16

35 36 37

Departamentos en el piso 3:

2 5 10 15 21 24 29 32 33 39

error0:

14464

Departamentos en el piso 1:

Columns 1 through 13

1 4 8 9 11 12 14 17 18 23 26

34 38

Column 14

40

Departamentos en el piso 2:

Columns 1 through 13

3 6 7 13 16 19 20 22 25 27 28

30 31

Columns 14 through 16

35 36 37

Departamentos en el piso 3:

2 5 10 15 21 24 29 32 33 39

19

error0:

14464

Departamentos en el piso 1:

Columns 1 through 13

1 4 8 9 11 12 14 17 18 23 26

34 38

Column 14

40

Departamentos en el piso 2:

Columns 1 through 13

3 6 7 13 16 19 20 22 25 27 28

30 31

Columns 14 through 16

35 36 37

Departamentos en el piso 3:

2 5 10 15 21 24 29 32 33 39

error0:

14464

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

Columns 14 through 16

34 37 39

error0:

12544

20

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

Columns 14 through 16

34 37 39

error0:

12544

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

Columns 14 through 16

34 37 39

error0:

12544

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

21

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

Columns 14 through 16

34 37 39

error0:

12544

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

Columns 14 through 16

34 37 39

error0:

12544

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

22

Columns 14 through 16

34 37 39
error0:
12544

Departamentos en el piso 1:
1 2 15 19 20 22 27 31 32 35 38
40
Departamentos en el piso 2:
3 4 5 7 12 16 21 23 28 30 33
36
Departamentos en el piso 3:
Columns 1 through 13
6 8 9 10 11 13 14 17 18 24 25
26 29
Columns 14 through 16
34 37 39
error0:
12544

Departamentos en el piso 1:
1 2 15 19 20 22 27 31 32 35 38
40
Departamentos en el piso 2:
3 4 5 7 12 16 21 23 28 30 33
36
Departamentos en el piso 3:
Columns 1 through 13
6 8 9 10 11 13 14 17 18 24 25
26 29
Columns 14 through 16
34 37 39
error0:
12544
23

Departamentos en el piso 1:
1 2 15 19 20 22 27 31 32 35 38
40
Departamentos en el piso 2:
3 4 5 7 12 16 21 23 28 30 33
36
Departamentos en el piso 3:
Columns 1 through 13
6 8 9 10 11 13 14 17 18 24 25
26 29
Columns 14 through 16
34 37 39
error0:
12544

Departamentos en el piso 1:
1 2 15 19 20 22 27 31 32 35 38
40
Departamentos en el piso 2:
3 4 5 7 12 16 21 23 28 30 33
36
Departamentos en el piso 3:
Columns 1 through 13
6 8 9 10 11 13 14 17 18 24 25
26 29
Columns 14 through 16
34 37 39
error0:
12544

Departamentos en el piso 1:
1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

24

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

Columns 14 through 16

34 37 39

error0:

12544

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

Columns 14 through 16

34 37 39

error0:

12544

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

25

Columns 14 through 16

34 37 39

error0:

12544

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

Columns 14 through 16

34 37 39

error0:

12544

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25
26 29

Columns 14 through 16

34 37 39

error0:

12544

26

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

Columns 14 through 16

34 37 39

error0:

12544

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38

40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33

36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25

26 29

Columns 14 through 16

```
34 37 39
error0:
12544
-----
-----
Departamentos en el piso 1:
1 2 15 19 20 22 27 31 32 35 38
40
Departamentos en el piso 2:
27
3 4 5 7 12 16 21 23 28 30 33
36
Departamentos en el piso 3:
Columns 1 through 13
6 8 9 10 11 13 14 17 18 24 25
26 29
Columns 14 through 16
34 37 39
error0:
12544
-----
-----
Departamentos en el piso 1:
1 2 15 19 20 22 27 31 32 35 38
40
Departamentos en el piso 2:
3 4 5 7 12 16 21 23 28 30 33
36
Departamentos en el piso 3:
Columns 1 through 13
6 8 9 10 11 13 14 17 18 24 25
26 29
Columns 14 through 16
34 37 39
error0:
12544
```


Departamentos en el piso 1:
1 2 15 19 20 22 27 31 32 35 38
40
Departamentos en el piso 2:
3 4 5 7 12 16 21 23 28 30 33
36
Departamentos en el piso 3:
Columns 1 through 13
6 8 9 10 11 13 14 17 18 24 25
26 29
28
Columns 14 through 16
34 37 39
error0:
12544

Departamentos en el piso 1:
1 2 15 19 20 22 27 31 32 35 38
40
Departamentos en el piso 2:
3 4 5 7 12 16 21 23 28 30 33
36
Departamentos en el piso 3:
Columns 1 through 13
6 8 9 10 11 13 14 17 18 24 25
26 29
Columns 14 through 16
34 37 39
error0:
12544

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38
40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33
36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25
26 29

Columns 14 through 16

34 37 39

error0:

12544

29

Departamentos en el piso 1:

1 2 15 19 20 22 27 31 32 35 38
40

Departamentos en el piso 2:

3 4 5 7 12 16 21 23 28 30 33
36

Departamentos en el piso 3:

Columns 1 through 13

6 8 9 10 11 13 14 17 18 24 25
26 29

Columns 14 through 16

34 37 39

error0:

12544

Departamentos en el piso 1:

1 5 6 8 13 15 19 23 29 34 35
36 40

Departamentos en el piso 2:

2 7 10 14 22 24 26 27 30 31 37
38

Departamentos en el piso 3:

Columns 1 through 13

3 4 9 11 12 16 17 18 20 21 25
28 32

Columns 14 through 15

33 39

error0:

12160

Departamentos en el piso 1:

1 5 6 8 13 15 19 23 29 34 35
36 40

Departamentos en el piso 2:

30

2 7 10 14 22 24 26 27 30 31 37
38

Departamentos en el piso 3:

Columns 1 through 13

3 4 9 11 12 16 17 18 20 21 25
28 32

Columns 14 through 15

33 39

error0:

12160

Departamentos en el piso 1:

1 5 6 8 13 15 19 23 29 34 35
36 40

Departamentos en el piso 2:

2 7 10 14 22 24 26 27 30 31 37
38

Departamentos en el piso 3:

Columns 1 through 13
3 4 9 11 12 16 17 18 20 21 25
28 32

Columns 14 through 15

33 39

error0:

12160

Departamentos en el piso 1:

1 5 6 8 13 15 19 23 29 34 35

36 40

Departamentos en el piso 2:

2 7 10 14 22 24 26 27 30 31 37

38

Departamentos en el piso 3:

Columns 1 through 13

3 4 9 11 12 16 17 18 20 21 25

28 32

31

Columns 14 through 15

33 39

error0:

12160

Departamentos en el piso 1:

1 5 6 8 13 15 19 23 29 34 35

36 40

Departamentos en el piso 2:

2 7 10 14 22 24 26 27 30 31 37

38

Departamentos en el piso 3:

Columns 1 through 13

3 4 9 11 12 16 17 18 20 21 25

28 32

Columns 14 through 15
33 39
error0:
12160

Departamentos en el piso 1:
1 5 6 8 13 15 19 23 29 34 35
36 40
Departamentos en el piso 2:
2 7 10 14 22 24 26 27 30 31 37
38
Departamentos en el piso 3:
Columns 1 through 13
3 4 9 11 12 16 17 18 20 21 25
28 32
Columns 14 through 15
33 39
error0:
12160
32

Departamentos en el piso 1:
1 5 6 8 13 15 19 23 29 34 35
36 40
Departamentos en el piso 2:
2 7 10 14 22 24 26 27 30 31 37
38
Departamentos en el piso 3:
Columns 1 through 13
3 4 9 11 12 16 17 18 20 21 25
28 32
Columns 14 through 15
33 39
error0:

12160

Departamentos en el piso 1:

1 5 6 8 13 15 19 23 29 34 35

36 40

Departamentos en el piso 2:

2 7 10 14 22 24 26 27 30 31 37

38

Departamentos en el piso 3:

Columns 1 through 13

3 4 9 11 12 16 17 18 20 21 25

28 32

Columns 14 through 15

33 39

error0:

12160

Departamentos en el piso 1:

1 5 6 8 13 15 19 23 29 34 35

36 40

Departamentos en el piso 2:

33

2 7 10 14 22 24 26 27 30 31 37

38

Departamentos en el piso 3:

Columns 1 through 13

3 4 9 11 12 16 17 18 20 21 25

28 32

Columns 14 through 15

33 39

error0:

12160

Departamentos en el piso 1:
1 5 6 8 13 15 19 23 29 34 35
36 40

Departamentos en el piso 2:
2 7 10 14 22 24 26 27 30 31 37
38

Departamentos en el piso 3:
Columns 1 through 13
3 4 9 11 12 16 17 18 20 21 25
28 32

Columns 14 through 15
33 39

error0:
12160

Departamentos en el piso 1:
1 5 6 8 13 15 19 23 29 34 35
36 40

Departamentos en el piso 2:
2 7 10 14 22 24 26 27 30 31 37
38

Departamentos en el piso 3:
Columns 1 through 13
3 4 9 11 12 16 17 18 20 21 25
28 32

34
Columns 14 through 15
33 39

error0:
12160

Departamentos en el piso 1:
1 5 6 8 13 15 19 23 29 34 35
36 40

Departamentos en el piso 2:
2 7 10 14 22 24 26 27 30 31 37
38

Departamentos en el piso 3:
Columns 1 through 13
3 4 9 11 12 16 17 18 20 21 25
28 32
Columns 14 through 15
33 39

error0:
12160

Departamentos en el piso 1:
1 5 6 8 13 15 19 23 29 34 35
36 40

Departamentos en el piso 2:
2 7 10 14 22 24 26 27 30 31 37
38

Departamentos en el piso 3:
Columns 1 through 13
3 4 9 11 12 16 17 18 20 21 25
28 32
Columns 14 through 15
33 39

error0:
12160

35

Departamentos en el piso 1:
Columns 1 through 13
1 2 7 9 15 16 24 25 29 30 31
35 38

Columns 14 through 15
39 40

Departamentos en el piso 2:
4 5 6 8 10 14 17 18 21 26 27
36 37

Departamentos en el piso 3:
3 11 12 13 19 20 22 23 28 32 33
34

error0:
11904

Departamentos en el piso 1:
Columns 1 through 13
1 2 7 9 15 16 24 25 29 30 31
35 38
Columns 14 through 15
39 40

Departamentos en el piso 2:
4 5 6 8 10 14 17 18 21 26 27
36 37

Departamentos en el piso 3:
3 11 12 13 19 20 22 23 28 32 33
34

error0:
11904

Departamentos en el piso 1:
Columns 1 through 13
1 2 7 9 15 16 24 25 29 30 31
35 38
36

Columns 14 through 15
39 40

Departamentos en el piso 2:
4 5 6 8 10 14 17 18 21 26 27
36 37

Departamentos en el piso 3:
3 11 12 13 19 20 22 23 28 32 33
34

error0:

11904

Departamentos en el piso 1:

Columns 1 through 13

1 2 7 9 15 16 24 25 29 30 31

35 38

Columns 14 through 15

39 40

Departamentos en el piso 2:

4 5 6 8 10 14 17 18 21 26 27

36 37

Departamentos en el piso 3:

3 11 12 13 19 20 22 23 28 32 33

34

error0:

11904

Departamentos en el piso 1:

Columns 1 through 13

1 2 7 9 15 16 24 25 29 30 31

35 38

Columns 14 through 15

39 40

Departamentos en el piso 2:

4 5 6 8 10 14 17 18 21 26 27

36 37

37

Departamentos en el piso 3:

3 11 12 13 19 20 22 23 28 32 33

34

error0:

11904

Departamentos en el piso 1:

Columns 1 through 13

1 2 7 9 15 16 24 25 29 30 31

35 38

Columns 14 through 15

39 40

Departamentos en el piso 2:

4 5 6 8 10 14 17 18 21 26 27

36 37

Departamentos en el piso 3:

3 11 12 13 19 20 22 23 28 32 33

34

error0:

11904

Departamentos en el piso 1:

Columns 1 through 13

1 2 7 9 15 16 24 25 29 30 31

35 38

Columns 14 through 15

39 40

Departamentos en el piso 2:

4 5 6 8 10 14 17 18 21 26 27

36 37

Departamentos en el piso 3:

3 11 12 13 19 20 22 23 28 32 33

34

error0:

11904

38

Departamentos en el piso 1:
Columns 1 through 13
1 2 7 9 15 16 24 25 29 30 31
35 38
Columns 14 through 15
39 40
Departamentos en el piso 2:
4 5 6 8 10 14 17 18 21 26 27
36 37
Departamentos en el piso 3:
3 11 12 13 19 20 22 23 28 32 33
34
error0:
11904

Departamentos en el piso 1:
Columns 1 through 13
1 2 7 9 15 16 24 25 29 30 31
35 38
Columns 14 through 15
39 40
Departamentos en el piso 2:
4 5 6 8 10 14 17 18 21 26 27
36 37
Departamentos en el piso 3:
3 11 12 13 19 20 22 23 28 32 33
34
error0:
11904

Departamentos en el piso 1:
Columns 1 through 13
1 2 7 9 15 16 24 25 29 30 31

35 38

Columns 14 through 15

39

39 40

Departamentos en el piso 2:

4 5 6 8 10 14 17 18 21 26 27

36 37

Departamentos en el piso 3:

3 11 12 13 19 20 22 23 28 32 33

34

error0:

11904

Departamentos en el piso 1:

Columns 1 through 13

1 2 7 9 15 16 24 25 29 30 31

35 38

Columns 14 through 15

39 40

Departamentos en el piso 2:

4 5 6 8 10 14 17 18 21 26 27

36 37

Departamentos en el piso 3:

3 11 12 13 19 20 22 23 28 32 33

34

error0:

11904

Departamentos en el piso 1:

Columns 1 through 13

1 2 7 9 15 16 24 25 29 30 31

35 38

Columns 14 through 15

39 40

Departamentos en el piso 2:
4 5 6 8 10 14 17 18 21 26 27
36 37

Departamentos en el piso 3:
40
3 11 12 13 19 20 22 23 28 32 33
34

error0:
11904

Departamentos en el piso 1:
Columns 1 through 13
1 2 7 9 15 16 24 25 29 30 31
35 38

Columns 14 through 15
39 40

Departamentos en el piso 2:
4 5 6 8 10 14 17 18 21 26 27
36 37

Departamentos en el piso 3:
3 11 12 13 19 20 22 23 28 32 33
34
error0:

11904

Departamentos en el piso 1:
Columns 1 through 13
1 2 7 9 15 16 24 25 29 30 31
35 38

Columns 14 through 15
39 40

Departamentos en el piso 2:
4 5 6 8 10 14 17 18 21 26 27
36 37

Departamentos en el piso 3:
3 11 12 13 19 20 22 23 28 32 33
34

error0:

11904

41

Departamentos en el piso 1:

Columns 1 through 13

1 2 7 9 15 16 24 25 29 30 31
35 38

Columns 14 through 15

39 40

Departamentos en el piso 2:

4 5 6 8 10 14 17 18 21 26 27
36 37

Departamentos en el piso 3:

3 11 12 13 19 20 22 23 28 32 33
34

error0:

11904

Departamentos en el piso 1:

1 5 6 9 12 14 15 17 22 24 29
31 40

Departamentos en el piso 2:

Columns 1 through 13

2 7 10 11 18 19 20 21 23 25 28
36 37

Columns 14 through 15

38 39

Departamentos en el piso 3:

3 4 8 13 16 26 27 30 32 33 34
35

error0:

11008

Departamentos en el piso 1:

1 5 6 9 12 14 15 17 22 24 29

31 40

Departamentos en el piso 2:

Columns 1 through 13

42

2 7 10 11 18 19 20 21 23 25 28

36 37

Columns 14 through 15

38 39

Departamentos en el piso 3:

3 4 8 13 16 26 27 30 32 33 34

35

error0:

11008

Departamentos en el piso 1:

1 5 6 9 12 14 15 17 22 24 29

31 40

Departamentos en el piso 2:

Columns 1 through 13

2 7 10 11 18 19 20 21 23 25 28

36 37

Columns 14 through 15

38 39

Departamentos en el piso 3:

3 4 8 13 16 26 27 30 32 33 34

35

error0:

11008

Departamentos en el piso 1:
1 5 6 9 12 14 15 17 22 24 29
31 40
Departamentos en el piso 2:
Columns 1 through 13
2 7 10 11 18 19 20 21 23 25 28
36 37
Columns 14 through 15
38 39
Departamentos en el piso 3:
43
3 4 8 13 16 26 27 30 32 33 34
35
error0:
11008

Departamentos en el piso 1:
Columns 1 through 13
1 5 6 7 10 23 24 25 26 28 30
35 36
Column 14
40
Departamentos en el piso 2:
Columns 1 through 13
3 9 11 12 13 16 17 19 21 22 29
31 33
Column 14
37
Departamentos en el piso 3:
2 4 8 14 15 18 20 27 32 34 38
39
error0:
9280

Departamentos en el piso 1:
Columns 1 through 13
1 5 6 7 10 23 24 25 26 28 30
35 36
Column 14
40
Departamentos en el piso 2:
Columns 1 through 13
3 9 11 12 13 16 17 19 21 22 29
31 33
Column 14
44
37
Departamentos en el piso 3:
2 4 8 14 15 18 20 27 32 34 38
39
error0:
9280

Departamentos en el piso 1:
Columns 1 through 13
1 5 6 7 10 23 24 25 26 28 30
35 36
Column 14
40
Departamentos en el piso 2:
Columns 1 through 13
3 9 11 12 13 16 17 19 21 22 29
31 33
Column 14
37
Departamentos en el piso 3:
2 4 8 14 15 18 20 27 32 34 38
39

error0:

9280

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 7 10 23 24 25 26 28 30

35 36

Column 14

40

Departamentos en el piso 2:

Columns 1 through 13

45

3 9 11 12 13 16 17 19 21 22 29

31 33

Column 14

37

Departamentos en el piso 3:

2 4 8 14 15 18 20 27 32 34 38

39

error0:

9280

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35

36 37

Columns 14 through 15

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24

33

Departamentos en el piso 3:

2 7 8 13 17 23 26 27 29 31 32

34 39

```
error0:
8064
-----
-----
Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37
Columns 14 through 15
38 40
Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33
Departamentos en el piso 3:
46
2 7 8 13 17 23 26 27 29 31 32
34 39
error0:
8064
-----
-----
Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37
Columns 14 through 15
38 40
Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33
Departamentos en el piso 3:
2 7 8 13 17 23 26 27 29 31 32
34 39
error0:
8064
-----
```

Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37
Columns 14 through 15
38 40
Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33
Departamentos en el piso 3:
2 7 8 13 17 23 26 27 29 31 32
34 39
error0:
8064

47
Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37
Columns 14 through 15
38 40
Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33
Departamentos en el piso 3:
2 7 8 13 17 23 26 27 29 31 32
34 39
error0:
8064

Departamentos en el piso 1:
Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35
36 37

Columns 14 through 15

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24
33

Departamentos en el piso 3:

2 7 8 13 17 23 26 27 29 31 32
34 39

error0:

8064

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35
36 37

Columns 14 through 15

48

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24
33

Departamentos en el piso 3:

2 7 8 13 17 23 26 27 29 31 32
34 39

error0:

8064

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35
36 37

Columns 14 through 15

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24

33

Departamentos en el piso 3:

2 7 8 13 17 23 26 27 29 31 32

34 39

error0:

8064

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35

36 37

Columns 14 through 15

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24

33

Departamentos en el piso 3:

49

2 7 8 13 17 23 26 27 29 31 32

34 39

error0:

8064

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35

36 37

Columns 14 through 15

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24

33

Departamentos en el piso 3:

2 7 8 13 17 23 26 27 29 31 32

34 39

error0:

8064

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35

36 37

Columns 14 through 15

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24

33

Departamentos en el piso 3:

2 7 8 13 17 23 26 27 29 31 32

34 39

error0:

8064

50

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35

36 37

Columns 14 through 15

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24

33

Departamentos en el piso 3:

2 7 8 13 17 23 26 27 29 31 32

34 39
error0:
8064

Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37
Columns 14 through 15
38 40
Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33
Departamentos en el piso 3:
2 7 8 13 17 23 26 27 29 31 32
34 39
error0:
8064

Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37
Columns 14 through 15
51
38 40
Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33
Departamentos en el piso 3:
2 7 8 13 17 23 26 27 29 31 32
34 39
error0:
8064

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35

36 37

Columns 14 through 15

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24

33

Departamentos en el piso 3:

2 7 8 13 17 23 26 27 29 31 32

34 39

error0:

8064

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35

36 37

Columns 14 through 15

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24

33

Departamentos en el piso 3:

52

2 7 8 13 17 23 26 27 29 31 32

34 39

error0:

8064

Departamentos en el piso 1:

Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37

Columns 14 through 15
38 40

Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33

Departamentos en el piso 3:
2 7 8 13 17 23 26 27 29 31 32
34 39

error0:
8064

Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37

Columns 14 through 15
38 40

Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33

Departamentos en el piso 3:
2 7 8 13 17 23 26 27 29 31 32
34 39

error0:
8064

53

Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37

Columns 14 through 15
38 40
Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33
Departamentos en el piso 3:
2 7 8 13 17 23 26 27 29 31 32
34 39
error0:
8064

Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37
Columns 14 through 15
38 40
Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33
Departamentos en el piso 3:
2 7 8 13 17 23 26 27 29 31 32
34 39
error0:
8064

Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37
Columns 14 through 15
54
38 40
Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24
33

Departamentos en el piso 3:

2 7 8 13 17 23 26 27 29 31 32
34 39

error0:

8064

Departamentos en el piso 1:

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35
36 37

Columns 14 through 15

38 40

Departamentos en el piso 2:

3 4 9 10 11 12 14 19 21 22 24
33

Departamentos en el piso 3:

2 7 8 13 17 23 26 27 29 31 32
34 39

error0:

8064

resultado -----

dept1 =

Columns 1 through 13

1 5 6 15 16 18 20 25 28 30 35
36 37

Columns 14 through 15

38 40

dept2 =

55

3 4 9 10 11 12 14 19 21 22 24
33

```

dept3 =
2 7 8 13 17 23 26 27 29 31 32
34 39
Departamentos en el piso 1:
Columns 1 through 13
1 5 6 15 16 18 20 25 28 30 35
36 37
Columns 14 through 15
38 40
Departamentos en el piso 2:
3 4 9 10 11 12 14 19 21 22 24
33
Departamentos en el piso 3:
2 7 8 13 17 23 26 27 29 31 32
34 39
Funcion de costo vertical final:
8064
56

```

Second Stage

```

pisol=[dept1'];
piso2=[dept2'];
piso3=[dept3'];
sec_p2y3=[8 6 5 6 6 8 4 8 6 8 4 4 5 4 8 8 8 6 5 6 6 8 4 8 6 8 4 4
5 4 8 8 8 6 5 6 8 6 5 6 8 6 5 6 8 6 5 5 5 4 8 4 4 5 6 5 4 5 6 6 8 6 5
5 5 4 8 4 4 5 6 5 4 5 6 6 8 6 5 6 6 8 4 8 8 6 5 6 8 6 5 5 4 5 6 6 6 8
4 8 6 8 4 8 6 8 4 8 6 8 4 4 4 5 6 5 5 4 8 4 5 4 8 8 6 8 4 8 8 6 5 6 8
6 5 6 8 6 5];
sec_p1=[4 8 6 8 4 8 6 8 4 8 6 8 4 8 6 8 4 8 6 6 5 6 8 6 6 5 4 5 6
5 4 4 8 4 5 5 5 6 8 6 6 5 4 5 6 5 4 4 8 4 5 5 5 6 8 6 5 6 8 6 5 6 8 6
5 6 8 8 8 4 5 4 4 8 6 8 4 8 6 6 5 6 8 8 8 4 5 4 4 8 6 8 4 8 6 6 5 6 8
6 5 6 8 6 5 6 8];
p_inicial1=[10,2];
p_inicial2y3=[10,1];
[Ac, m_layout1, posiciones1]= layout(Ac, p_inicial1, pisol,
sec_p1,1);

```

```

figure,
imagesc(m_layout1), caxis([-10 35])
title('Primer Piso - Inicial');
57
for ii=1:1:(size(posiciones1,1))
text(posiciones1(ii,2),posiciones1(ii,1),strcat('\fontsize{15}',num2str(p
osiciones1([0.5 0.5 0.5]));
end
text(12,7,'\fontsize{15} 40','color',[0.5 0.5 0.5]);
axis square
[Ac, m_layout2,posiciones2]= layout(Ac, p_inicial2y3, piso2,
sec_p2y3,2);
figure,
imagesc(m_layout2)
title('Segundo Piso - Inicial');
for ii=1:1:(size(posiciones2,1))
text(posiciones2(ii,2),posiciones2(ii,1),strcat('\fontsize{15}',num2str(p
osiciones2([0.5 0.5 0.5]));
end
axis square
[Ac, m_layout3, posiciones3]= layout(Ac, p_inicial2y3, piso3,
sec_p2y3,3);
figure,
imagesc(m_layout3)
title('Tercer Piso - Inicial');
for ii=1:1:(size(posiciones3,1))
text(posiciones3(ii,2),posiciones3(ii,1),strcat('\fontsize{15}',num2str(p
osiciones3([0.5 0.5 0.5]));
end
axis square
posiciones1(length(posiciones1(:,3))+1,1:3)=0;
posiciones1(length(posiciones1(:,3)),3)=40;
error1=0;
error0=0;
cont=1;
for k=1:No_f

```

```

for i=1:length(eval(strcat('C',num2str(k))))-1
% for i=1:1
[ik1,jk1]=find(posiciones1(:,3)==(eval(strcat('C',num2str(k),'(',num2str(
i),')'))));
58
[ik2,jk2]=find(posiciones1(:,3)==(eval(strcat('C',num2str(k),'(',num2str(
i
+1),')'))));
[ik3,jk3]=find(posiciones2(:,3)==(eval(strcat('C',num2str(k),'(',num2str(
i),')'))));
[ik4,jk4]=find(posiciones2(:,3)==(eval(strcat('C',num2str(k),'(',num2str(
i
+1),')'))));
[ik5,jk5]=find(posiciones3(:,3)==(eval(strcat('C',num2str(k),'(',num2str(
i),')'))));
[ik6,jk6]=find(posiciones3(:,3)==(eval(strcat('C',num2str(k),'(',num2str(
i
+1),')'))));
if isempty(ik1)==0
dk=1;
end
if isempty(ik2)==0
dg=1;
end
if isempty(ik3)==0
dk=2;
end
if isempty(ik4)==0
dg=2;
end
if isempty(ik5)==0
dk=3;
end
if isempty(ik6)==0
dg=3;
end

```

```

[ixx1
jxx1]=find(eval(strcat('C',num2str(k),'(',num2str(i),')'))==Ac(1,:));
[ixx2 jxx2]=find(eval(strcat('C',num2str(k),'(',num2str(i
+1),')'))==Ac(1,:));
if (dk-dg==0)
% disp('son el mismo piso');
distance=sqrt((Ac(3,jxx1)-Ac(3,jxx2))^2+(Ac(4,jxx1)-
Ac(4,jxx2))^2);
else
% disp('son pisos distintos');
distance=sqrt((Ac(3,jxx1)-lift(1))^2+(Ac(4,jxx1)-
lift(2))^2)+sqrt((Ac(3,jxx1)-lift(1))^2+(Ac(4,jxx1)-lift(2))^2));
end
vector2(cont)=distance;
cont=cont+1;
59
error0=distance*Ch*(eval(strcat('f',num2str(k),'(',num2str(i),')')))+erro
r0;
end
end
disp('Funcion de costo horizontal inicial:');
disp(error0);
lift=[11 1]; % posiciones del ascensor
n_cycles = 10;
n_tries_for_cycle = 10;
n_dimensions = 2;
% Probabilities of accept the worst solution
initial_probability = 0.7;
final_probability = 0.001;
% Calculate temperature and steps
initial_temperature = -1 / log(initial_probability);
final_temperature = -1 / log(final_probability);
reduction_fraction = (final_temperature / initial_temperature)^(1/
(n_cycles-1));
% Initial position
% Generate random number in the range of each dimension

```

```

range_min=1;
range_max=40;
%
*****
% print initial values
sprintf('initial_temperature = %f', initial_temperature)
sprintf('final_temperature = %f', final_temperature)
sprintf('reduction_fraction = %f', reduction_fraction)
conta=1;
X(1)=initial_temperature;
cont=2;
temperature=initial_temperature;
while (temperature>=final_temperature)
temperature=temperature*reduction_fraction;
X(cont)=temperature;
cont=cont+1;
end;
for temperature=X
sprintf('Temperature = %f', temperature);
% Iterations for each temperature
60
for n=1:n_tries_for_cicle
dept_1=dept1(randperm(length(dept1)));
dept_2=dept2(randperm(length(dept2)));
dept_3=dept3(randperm(length(dept3)));
Ac=A;
piso1=[dept_1'];
piso2=[dept_2'];
piso3=[dept_3'];
[Ac, m_layout1, posiciones1]= layout(Ac, p_inicial1, piso1, sec_p1,1);
% figure
% imagesc(m_layout1), caxis([-10 35])
% for ii=1:1:(size(posiciones1,1))
%
text(posiciones1(ii,2),posiciones1(ii,1),strcat('\fontsize{15}',num2str(p
osiciones1([0.5 0.5 0.5]));

```

```

% end
% text(12,7,'\fontsize{15} 40','color',[0.5 0.5 0.5]);
% axis square
% posiciones=[];
[Ac, m_layout2, posiciones2]= layout(Ac, p_inicial2y3, piso2,
sec_p2y3,2);
% figure
% imagesc(m_layout2)
%
% for ii=1:1:(size(posiciones2,1))
%
text(posiciones2(ii,2),posiciones2(ii,1),strcat('\fontsize{15}',num2str(p
osiciones2([0.5 0.5 0.5]));
% end
% axis square
% posiciones=[];
[Ac, m_layout3, posiciones3]= layout(Ac, p_inicial2y3, piso3,
sec_p2y3,3);
% figure
% imagesc(m_layout3)
%
% for ii=1:1:(size(posiciones3,1))
%
text(posiciones3(ii,2),posiciones3(ii,1),strcat('\fontsize{15}',num2str(p
osiciones3('color', [0.5 0.5 0.5]));
% end
% axis square
cont=1;
61
posiciones1(length(posiciones1(:,3))+1,1:3)=0;
posiciones1(length(posiciones1(:,3)),3)=40;
for k=1:No_f
for i=1:length(eval(strcat('C',num2str(k))))-1
% for i=1:2
[ik1,jk1]=find(posiciones1(:,3)==(eval(strcat('C',num2str(k),'(',num2str(
i),')'))));

```

```

[ik2,jk2]=find(posiciones1(:,3)==(eval(strcat('C',num2str(k),'(',num2str(i
+1),')'))));
[ik3,jk3]=find(posiciones2(:,3)==(eval(strcat('C',num2str(k),'(',num2str(i
),')'))));
[ik4,jk4]=find(posiciones2(:,3)==(eval(strcat('C',num2str(k),'(',num2str(i
+1),')'))));
[ik5,jk5]=find(posiciones3(:,3)==(eval(strcat('C',num2str(k),'(',num2str(i
),')'))));
[ik6,jk6]=find(posiciones3(:,3)==(eval(strcat('C',num2str(k),'(',num2str(i
+1),')'))));
if isempty(ik1)==0
dk=1;
end
if isempty(ik2)==0
dg=1;
end
if isempty(ik3)==0
dk=2;
end
if isempty(ik4)==0
dg=2;
end
if isempty(ik5)==0
dk=3;
end
if isempty(ik6)==0
dg=3;
end
[ixx1
jxx1]=find(eval(strcat('C',num2str(k),'(',num2str(i),')'))==Ac(1,:));
[ixx2 jxx2]=find(eval(strcat('C',num2str(k),'(',num2str(i
+1),')'))==Ac(1,:));
if (dk-dg==0)

```

62

```
% disp('son el mismo piso');
distance=sqrt((Ac(3,jxx1)-Ac(3,jxx2))^2+(Ac(4,jxx1)-
Ac(4,jxx2))^2);
else
% disp('son pisos distintos');
distance=sqrt((Ac(3,jxx1)-lift(1))^2+(Ac(4,jxx1)-
lift(2))^2)+sqrt((Ac(3,jxx1)-lift(1))^2+(Ac(4,jxx1)-lift(2))^2));
end
% distance
vector3(cont)=distance;
cont=cont+1;
error1=distance*Ch*(eval(strcat('f',num2str(k),'(',num2str(i),')')))+erro
r1;
end
end
delta_error = error1 - error0;
if (delta_error <0)
for i=1:n_dimensions
dept_11 = dept_1;
dept_22 = dept_2;
dept_33 = dept_3;
end
error0 = error1;
else
p = exp(-(delta_error/temperature));
r = rand;
if (p>r)
for i=1:n_dimensions
dept_11 = dept_1;
dept_22 = dept_2;
dept_33 = dept_3;
end
error0 = error1;
end
end
```

```

disp('-----')
error1=0;
disp('error0:');
disp(error0)
Error2(conta)=error0;
conta=conta+1;
disp('-----')
end
end
63
pisol=[dept_11'];
piso2=[dept_22'];
piso3=[dept_33'];
sec_p2y3=[8 6 5 6 6 8 4 8 6 8 4 4 5 4 8 8 8 6 5 6 6 8 4 8 6 8 4 4 5 4
8 8 8 6 5 6 8 6 5 6 8 6 5 6 8 6 5 5 5 4 8 4 4 5 6 5 4 5 6 6 8 6 5 5 5
4 8 4 4 5 6 5 4 5 6 6 8 6 5 6 6 8 4 8 8 6 5 6 8 6 5 5 4 5 6 6 6 8 4 8
6 8 4 8 6 8 4 8 6 8 4 4 4 5 6 5 5 4 8 4 5 4 8 8 6 8 4 8 8 6 5 6 8 6 5
6 8 6 5];
sec_p1=[4 8 6 8 4 8 6 8 4 8 6 8 4 8 6 8 4 8 6 6 5 6 8 6 6 5 4 5 6 5 4
4 8 4 5 5 5 6 8 6 6 5 4 5 6 5 4 4 8 4 5 5 5 6 8 6 5 6 8 6 5 6 8 6 5 6
8 8 8 4 5 4 4 8 6 8 4 8 6 6 5 6 8 8 8 4 5 4 4 8 6 8 4 8 6 6 5 6 8 6 5
6 8 6 5 6 8];
p_inicial1=[10,2];
p_inicial2y3=[10,1];
[Ac, m_layout1,posiciones1]= layout(A, p_inicial1, piso1, sec_p1,1);
figure
imagesc(m_layout1), caxis([-10 35]),
title('Primer Piso - Final');
for ii=1:1:(size(posiciones1,1))
text(posiciones1(ii,2),posiciones1(ii,1),strcat('\fontsize{15}',num2str(p
osiciones1([0.5 0.5 0.5]));
end
text(12,7,'\fontsize{15} 40','color',[0.5 0.5 0.5]);
axis square
% posiciones=[];
[Ac, m_layout2,posiciones2]= layout(A, p_inicial2y3, piso2,

```

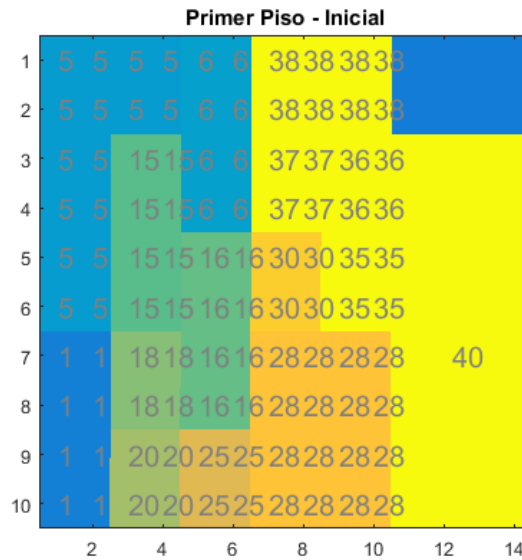
```

sec_p2y3,2);
figure,
imagesc(m_layout2)
title('Segundo Piso - Final');
for ii=1:1:(size(posiciones2,1))
text(posiciones2(ii,2),posiciones2(ii,1),strcat('\fontsize{15}',num2str(p
osiciones2([0.5 0.5 0.5]));
end
axis square
% posiciones=[];
[Ac, m_layout3, posiciones3]= layout(A, p_inicial2y3, piso3,
sec_p2y3,3);
figure,
imagesc(m_layout3)
title('Tercer Piso - Final');
64
for ii=1:1:(size(posiciones3,1))
text(posiciones3(ii,2),posiciones3(ii,1),strcat('\fontsize{15}',num2str(p
osiciones3([0.5 0.5 0.5]));
end
axis square
disp('Funcion de costo horizontal final:');
disp(error0);
figure
plot(Error2);
title('Funcion de costo - Segunda Etapa')

```

ANEXO C. Información obtenida cada vez que se corre el algoritmo (3 pisos, 40 departamentos)

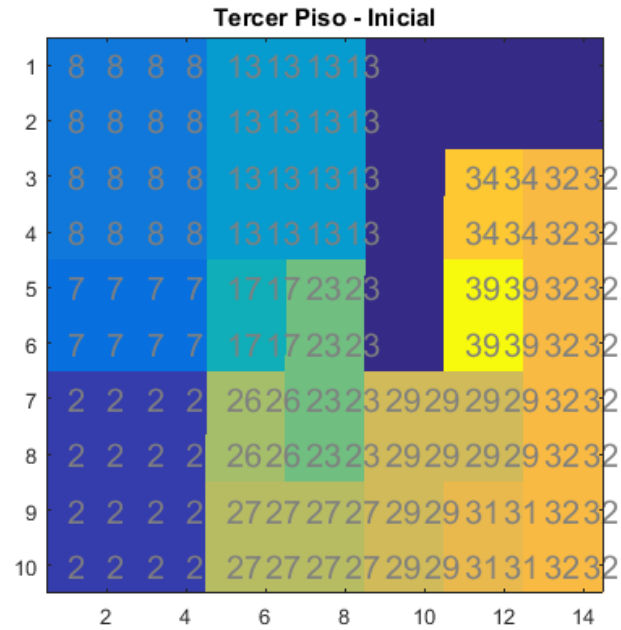
- Layout Primer Piso – Etapa 1



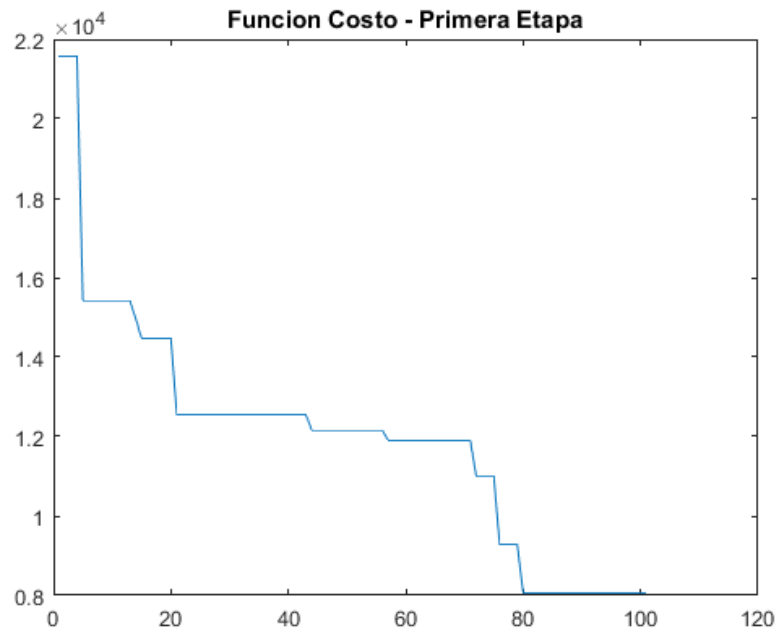
- Layout Segundo Piso – Etapa 1



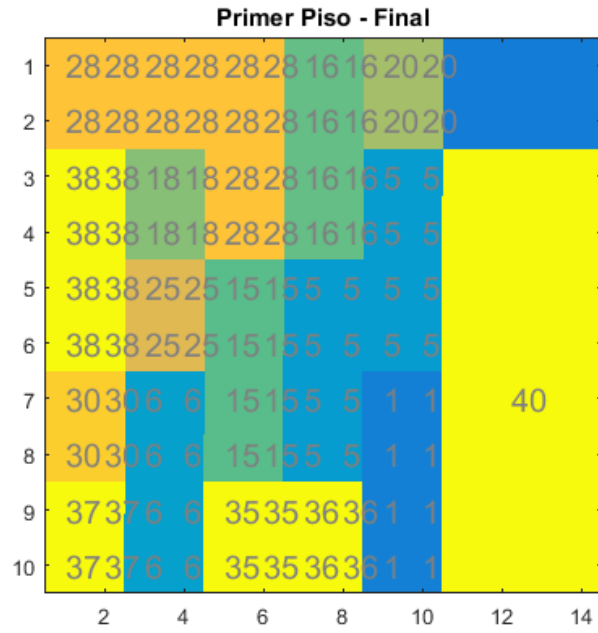
- Layout Tercer Piso – Etapa 1



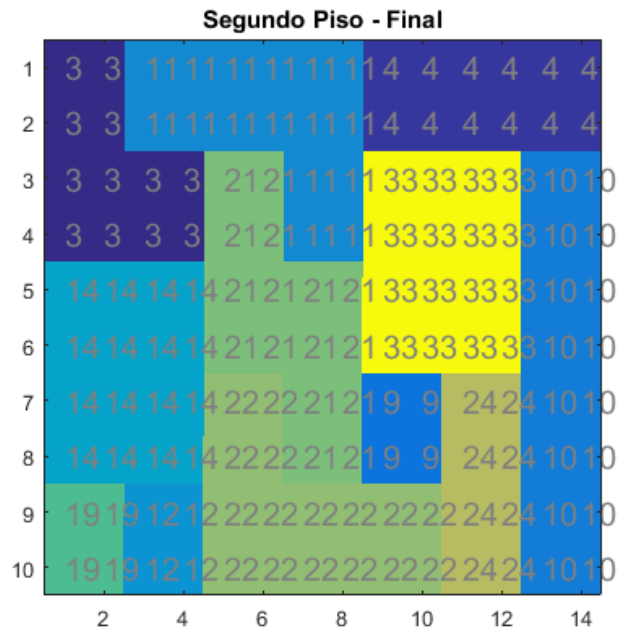
- Funcion Costo – Etapa 1



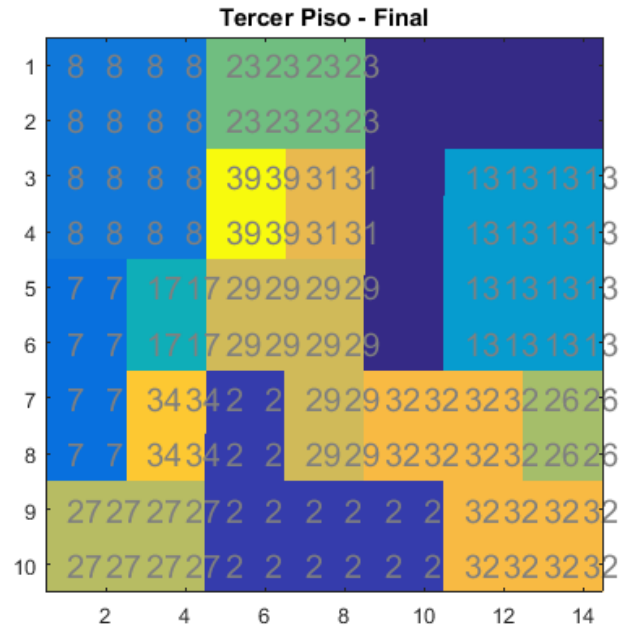
- Layout Piso 1 – Etapa 2



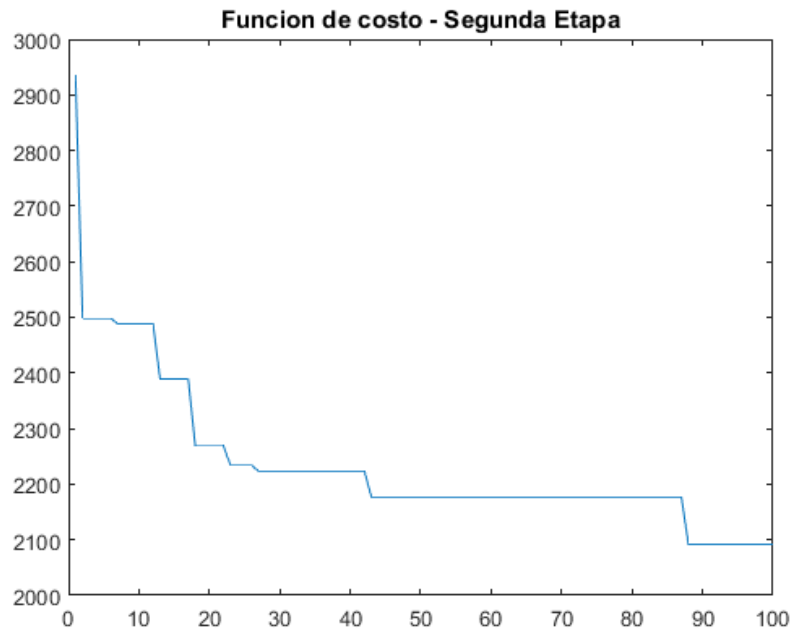
- Layout Piso 2 – Etapa 2



- Layout Piso 3 – Etapa 2



- Función Costo – Etapa 2



- Tiempos De Cómputo

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
num2str	1332304	68.152 s	29.802 s	
int2str	1332304	38.350 s	38.350 s	
strcat	668734	65.400 s	59.208 s	
blanks	668734	6.192 s	6.192 s	
layout	306	0.114 s	0.114 s	
newplot	8	0.419 s	0.289 s	
newplot>ObserveFigureNextPlot	8	0.004 s	0.004 s	
cla	8	0.087 s	0.062 s	
graphics\private\claNotify	8	0.003 s	0.003 s	
graphics\private\clo	8	0.021 s	0.021 s	
newplot>ObserveAxesNextPlot	8	0.125 s	0.038 s	
ishold	8	0.004 s	0.004 s	
title	8	3.687 s	3.684 s	
graph2d\private\labelcheck	8	0.002 s	0.002 s	
imagesc	6	0.057 s	0.019 s	
axis	6	0.007 s	0.006 s	
axis>allAxes	6	0.001 s	0.001 s	
dlmwrite>parseinput	5	0.012 s	0.008 s	
cell_ismember	5	0.002 s	0.002 s	
dlmwrite>setdlm	5	0.001 s	0.001 s	
dlmwrite>setoffset	5	0.000 s	0.000 s	
dlmwrite>setcoffset	5	0.000 s	0.000 s	
dlmwrite	5	0.044 s	0.032 s	
csvwrite	5	0.060 s	0.016 s	
caxis	2	0.002 s	0.002 s	
close	1	0.008 s	0.005 s	
close>getEmptyHandleList	1	0.001 s	0.001 s	
close>safefetchchildren	1	0.001 s	0.001 s	
close>checkfigs	1	0.000 s	0.000 s	
close>request_close	1	0.000 s	0.000 s	
simulated_annealing_matlab_v3	1	245.715 s	107.843 s	

ANEXO D. Iteraciones del algoritmo con n=78

Costo Vertical Inicial	Costo Horizontal Inicial	Costo Vertical Final	Costo Horizontal Final	Costo Inicial Total	Costo Final Total
\$ 21,568.0	\$ 8,064.0	\$ 3,103.4	\$ 2,091.6	\$ 29,632.0	\$ 5,195.0
\$ 35,840.0	\$ 9,152.0	\$ 3,069.8	\$ 2,237.2	\$ 44,992.0	\$ 5,307.0
\$ 22,336.0	\$ 9,664.0	\$ 3,317.3	\$ 2,354.3	\$ 32,000.0	\$ 5,671.6
\$ 20,800.0	\$ 7,424.0	\$ 3,027.9	\$ 2,064.7	\$ 28,224.0	\$ 5,092.6
\$ 24,064.0	\$ 8,576.0	\$ 3,427.7	\$ 2,159.3	\$ 32,640.0	\$ 5,587.0
\$ 24,384.0	\$ 9,344.0	\$ 3,341.7	\$ 2,359.1	\$ 33,728.0	\$ 5,700.8
\$ 35,264.0	\$ 9,280.0	\$ 2,966.1	\$ 2,204.0	\$ 44,544.0	\$ 5,170.1
\$ 24,384.0	\$ 9,088.0	\$ 3,287.9	\$ 2,264.5	\$ 33,472.0	\$ 5,552.4
\$ 14,720.0	\$ 10,176.0	\$ 3,075.2	\$ 2,212.4	\$ 24,896.0	\$ 5,287.6
\$ 22,784.0	\$ 8,896.0	\$ 3,007.3	\$ 2,130.6	\$ 31,680.0	\$ 5,137.9
\$ 13,184.0	\$ 7,680.0	\$ 3,069.4	\$ 2,022.6	\$ 20,864.0	\$ 5,092.0
\$ 24,000.0	\$ 8,704.0	\$ 3,312.9	\$ 2,252.2	\$ 32,704.0	\$ 5,565.1
\$ 36,288.0	\$ 9,216.0	\$ 3,108.8	\$ 2,119.7	\$ 45,504.0	\$ 5,228.5
\$ 24,512.0	\$ 9,792.0	\$ 3,301.6	\$ 2,462.2	\$ 34,304.0	\$ 5,763.8
\$ 23,680.0	\$ 8,640.0	\$ 3,346.7	\$ 2,216.2	\$ 32,320.0	\$ 5,562.9
\$ 23,616.0	\$ 8,832.0	\$ 3,056.7	\$ 2,217.1	\$ 32,448.0	\$ 5,273.8
\$ 34,816.0	\$ 9,472.0	\$ 3,231.0	\$ 2,283.5	\$ 44,288.0	\$ 5,514.5
\$ 36,416.0	\$ 8,128.0	\$ 3,055.5	\$ 2,090.6	\$ 44,544.0	\$ 5,146.1
\$ 11,456.0	\$ 8,832.0	\$ 3,288.2	\$ 2,240.3	\$ 20,288.0	\$ 5,528.5
\$ 13,824.0	\$ 8,320.0	\$ 2,962.6	\$ 2,081.6	\$ 22,144.0	\$ 5,044.2
\$ 20,416.0	\$ 10,112.0	\$ 3,283.7	\$ 2,380.1	\$ 30,528.0	\$ 5,663.8
\$ 23,808.0	\$ 7,488.0	\$ 3,181.9	\$ 2,057.9	\$ 31,296.0	\$ 5,239.8
\$ 14,144.0	\$ 8,128.0	\$ 3,068.9	\$ 2,105.1	\$ 22,272.0	\$ 5,174.0
\$ 14,720.0	\$ 10,176.0	\$ 3,075.2	\$ 2,212.4	\$ 24,896.0	\$ 5,287.6
\$ 22,784.0	\$ 8,896.0	\$ 3,007.3	\$ 2,130.6	\$ 31,680.0	\$ 5,137.9
\$ 13,184.0	\$ 7,680.0	\$ 3,069.4	\$ 2,022.6	\$ 20,864.0	\$ 5,092.0
\$ 24,000.0	\$ 8,704.0	\$ 3,312.9	\$ 2,252.2	\$ 32,704.0	\$ 5,565.1
\$ 36,288.0	\$ 9,216.0	\$ 3,108.8	\$ 2,119.7	\$ 45,504.0	\$ 5,228.5
\$ 24,512.0	\$ 9,792.0	\$ 3,301.6	\$ 2,462.2	\$ 34,304.0	\$ 5,763.8
\$ 23,680.0	\$ 8,640.0	\$ 3,346.7	\$ 2,216.2	\$ 32,320.0	\$ 5,562.9

Costo Vertical Inicial	Costo Horizontal Inicial	Costo Vertical Final	Costo Horizontal Final	Costo Inicial Total	Costo Final Total
\$ 23,616.0	\$ 8,832.0	\$ 3,056.7	\$ 2,217.1	\$ 32,448.0	\$ 5,273.8
\$ 34,816.0	\$ 9,472.0	\$ 3,231.0	\$ 2,283.5	\$ 44,288.0	\$ 5,514.5
\$ 36,416.0	\$ 8,128.0	\$ 3,055.5	\$ 2,090.6	\$ 44,544.0	\$ 5,146.1
\$ 11,456.0	\$ 8,832.0	\$ 3,288.2	\$ 2,240.3	\$ 20,288.0	\$ 5,528.5
\$ 34,496.0	\$ 9,536.0	\$ 3,164.6	\$ 2,153.6	\$ 44,032.0	\$ 5,318.2
\$ 13,824.0	\$ 8,320.0	\$ 2,962.6	\$ 2,081.6	\$ 22,144.0	\$ 5,044.2
\$ 23,808.0	\$ 7,488.0	\$ 3,181.9	\$ 2,057.9	\$ 31,296.0	\$ 5,239.8
\$ 21,568.0	\$ 8,064.0	\$ 3,103.4	\$ 2,091.6	\$ 29,632.0	\$ 5,195.0
\$ 35,840.0	\$ 9,152.0	\$ 3,069.8	\$ 2,237.2	\$ 44,992.0	\$ 5,307.0
\$ 22,336.0	\$ 9,664.0	\$ 3,317.3	\$ 2,354.3	\$ 32,000.0	\$ 5,671.6
\$ 20,800.0	\$ 7,424.0	\$ 3,027.9	\$ 2,064.7	\$ 28,224.0	\$ 5,092.6
\$ 24,384.0	\$ 9,344.0	\$ 3,341.7	\$ 2,359.1	\$ 33,728.0	\$ 5,700.8
\$ 35,264.0	\$ 9,280.0	\$ 2,966.1	\$ 2,204.0	\$ 44,544.0	\$ 5,170.1
\$ 24,384.0	\$ 9,088.0	\$ 3,287.9	\$ 2,264.5	\$ 33,472.0	\$ 5,552.4
\$ 14,720.0	\$ 10,176.0	\$ 3,075.2	\$ 2,212.4	\$ 24,896.0	\$ 5,287.6
\$ 22,784.0	\$ 8,896.0	\$ 3,007.3	\$ 2,130.6	\$ 31,680.0	\$ 5,137.9
\$ 13,184.0	\$ 7,680.0	\$ 3,069.4	\$ 2,022.6	\$ 20,864.0	\$ 5,092.0
\$ 36,288.0	\$ 9,216.0	\$ 3,108.8	\$ 2,119.7	\$ 45,504.0	\$ 5,228.5
\$ 14,912.0	\$ 9,984.0	\$ 3,492.3	\$ 2,168.6	\$ 24,896.0	\$ 5,660.9
\$ 24,512.0	\$ 9,792.0	\$ 3,301.6	\$ 2,462.2	\$ 34,304.0	\$ 5,763.8
\$ 23,680.0	\$ 8,640.0	\$ 3,346.7	\$ 2,216.2	\$ 32,320.0	\$ 5,562.9
\$ 21,568.0	\$ 8,064.0	\$ 3,103.4	\$ 2,091.6	\$ 29,632.0	\$ 5,195.0
\$ 37,440.0	\$ 8,960.0	\$ 3,490.9	\$ 2,245.3	\$ 46,400.0	\$ 5,736.2
\$ 20,800.0	\$ 7,424.0	\$ 3,027.9	\$ 2,064.7	\$ 28,224.0	\$ 5,092.6
\$ 24,384.0	\$ 9,344.0	\$ 3,341.7	\$ 2,359.1	\$ 33,728.0	\$ 5,700.8
\$ 35,264.0	\$ 9,280.0	\$ 2,966.1	\$ 2,204.0	\$ 44,544.0	\$ 5,170.1
\$ 9,792.0	\$ 8,320.0	\$ 3,408.5	\$ 2,228.0	\$ 18,112.0	\$ 5,636.5
\$ 24,384.0	\$ 9,088.0	\$ 3,287.9	\$ 2,264.5	\$ 33,472.0	\$ 5,552.4
\$ 14,144.0	\$ 8,128.0	\$ 3,068.9	\$ 2,105.1	\$ 22,272.0	\$ 5,174.0
\$ 14,720.0	\$ 10,176.0	\$ 3,075.2	\$ 2,212.4	\$ 24,896.0	\$ 5,287.6
\$ 22,784.0	\$ 8,896.0	\$ 3,007.3	\$ 2,130.6	\$ 31,680.0	\$ 5,137.9
\$ 13,184.0	\$ 7,680.0	\$ 3,069.4	\$ 2,022.6	\$ 20,864.0	\$ 5,092.0

Costo Vertical Inicial	Costo Horizontal Inicial	Costo Vertical Final	Costo Horizontal Final	Costo Inicial Total	Costo Final Total
\$ 24,000.0	\$ 8,704.0	\$ 3,312.9	\$ 2,252.2	\$ 32,704.0	\$ 5,565.1
\$ 22,528.0	\$ 9,664.0	\$ 3,449.7	\$ 2,426.0	\$ 32,192.0	\$ 5,875.7
\$ 36,288.0	\$ 9,216.0	\$ 3,108.8	\$ 2,119.7	\$ 45,504.0	\$ 5,228.5
\$ 24,512.0	\$ 9,792.0	\$ 3,301.6	\$ 2,462.2	\$ 34,304.0	\$ 5,763.8
\$ 23,680.0	\$ 8,640.0	\$ 3,346.7	\$ 2,216.2	\$ 32,320.0	\$ 5,562.9
\$ 21,568.0	\$ 8,064.0	\$ 3,103.4	\$ 2,091.6	\$ 29,632.0	\$ 5,195.0
\$ 35,840.0	\$ 9,152.0	\$ 3,069.8	\$ 2,237.2	\$ 44,992.0	\$ 5,307.0
\$ 35,840.0	\$ 9,152.0	\$ 3,069.8	\$ 2,237.2	\$ 44,992.0	\$ 5,307.0
\$ 24,064.0	\$ 8,576.0	\$ 3,427.7	\$ 2,159.3	\$ 32,640.0	\$ 5,587.0
\$ 24,384.0	\$ 9,344.0	\$ 3,341.7	\$ 2,359.1	\$ 33,728.0	\$ 5,700.8
\$ 35,264.0	\$ 9,280.0	\$ 2,966.1	\$ 2,204.0	\$ 44,544.0	\$ 5,170.1
\$ 9,792.0	\$ 8,320.0	\$ 3,408.5	\$ 2,228.0	\$ 18,112.0	\$ 5,636.5
\$ 24,384.0	\$ 9,088.0	\$ 3,287.9	\$ 2,264.5	\$ 33,472.0	\$ 5,552.4
\$ 22,336.0	\$ 9,664.0	\$ 3,317.3	\$ 2,354.3	\$ 32,000.0	\$ 5,671.6
\$ 37,440.0	\$ 8,960.0	\$ 3,490.9	\$ 2,245.3	\$ 46,400.0	\$ 5,736.2
\$ 20,800.0	\$ 7,424.0	\$ 3,027.9	\$ 2,064.7	\$ 28,224.0	\$ 5,092.6