

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**ESCUELA DE POSGRADO**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**UNIDAD DE POSGRADO**

**“UN ALGORITMO GREEDY PARA EL DISEÑO DE  
DISTRIBUCIÓN DE PLANTA CON MULTI PRODUCTOS  
Y RUTAS DE PROCESAMIENTO EN LAS  
INSTALACIONES INDUSTRIALES”**

**TESIS**

Para obtener el grado académico de Magíster en Ingeniería de Sistemas e Informática,  
con mención en Ingeniería de Software

**AUTOR**

Máximo Medardo Leyva Caballero

Lima – Perú

2015

**Máximo Medardo Leyva Caballero**

**UN ALGORITMO GREEDY PARA EL DISEÑO DE DISTRIBUCIÓN DE PLANTA  
CON MULTI PRODUCTOS Y RUTAS DE PROCESAMIENTO EN LAS  
INSTALACIONES INDUSTRIALES**

“Tesis presentada a la Universidad Nacional  
Mayor de San Marcos, para obtener el Grado de  
Magíster en Ingeniería de Sistemas e  
Informática, con la mención de Ingeniería de  
Software”.

**Asesor: Dr. David S. Mauricio Sánchez**

UNMSM – LIMA

MARZO 2015

Este trabajo está dedicado a mi familia, en especial a mi madre, por darme la vida y apoyarme incondicionalmente en todo momento.

Con todo mi corazón a mi hija, por ser la fuerza que me motiva en todo momento.

## **AGRADECIMIENTOS**

Al profesor Dr. David S. Mauricio Sánchez, por su orientación y dedicación para que este trabajo cumpla con los objetivos trazados.

Al profesor Dr. Luis Rivera E., por su constante apoyo y consejos permanentes durante las clases de Maestría.

A los profesores de la UNMSM, principalmente al Ing. Julio Salas Bacalla, docente del curso de Distribución de Planta, por sus observaciones teóricas que me sirvieron de mucho.

A todas aquellas personas que indirectamente me ayudaron a culminar este trabajo y que constituyen un invalorable apoyo.

Y por encima de todo, doy gracias a Dios.

# **Un algoritmo greedy para el diseño de distribución de planta con multi productos y rutas de procesamiento en las instalaciones industriales**

## **RESUMEN**

Esta tesis aborda el problema de distribución en planta, el cual, en líneas generales, pretende asignar o distribuir instalaciones en un sistema de producción. Existen muchos planteamientos diferentes dependiendo de la naturaleza de los talleres de trabajo consideradas de la planta industrial, por ejemplo el tipo de producto que se fabrica, la flexibilidad requerida en las rutas de procesamiento, la forma de las instalaciones, el número de niveles, etcétera. Uno de los planteamientos más abordados ha sido el problema de distribución en planta con instalaciones de áreas iguales, generalizándose como un problema de asignación cuadrática. Para solucionar este tipo de problemas, el presente trabajo desarrolla un algoritmo greedy para la distribución de las instalaciones en la planta con los siguientes tipos de naturaleza: sistema productivo por procesos, con rutas de procesamiento, presencia de equipos para el manejo de materiales en fila múltiple, distribución en un solo nivel y horizonte de planeamiento estático. El método propuesto **AlgoDist** minimiza el costo total del manejo de materiales considerando la métrica de distancia Manhattan y logra obtener un equilibrio adecuado entre los algoritmos tipo constructor que generan la distribución ocupando paulatinamente las posiciones del dominio de ubicación, y los algoritmos tipo mejoría que promueven la mejor solución a partir de una distribución inicial de planta.

El algoritmo heurístico es bastante eficaz para la resolución de problemas de optimización combinatoria de múltiples filas, con uso de tiempo computacional mínimo.

Finalmente, el rendimiento del algoritmo ha sido probado en varios casos de estudio y sus resultados fueron comparados con los obtenidos en otros trabajos citados en la bibliografía, mostrando un grado de eficiencia del 96%, pero con un bajo costo computacional.

**Palabras clave:** Problema de asignación cuadrática, Heurísticas, Algoritmo goloso.

# **A greedy algorithm for the design of plant layout with multi-products and processing routes in industrial facilities**

## **ABSTRACT**

This thesis addresses the problem of plant distribution, which, in general, attempts to assign or distribute facilities in a production system. There are many different approaches depending on the nature of the workshops considered the plant, for example the type of product being manufactured, the flexibility in processing routes, the way of facilities, the number of levels, etc. One of the approaches discussed was the problem of plant distribution facilities equal areas, as a generalized quadratic assignment problem. To solve such problems, this paper develops a greedy algorithm for facility layout on the ground with the following types of nature by processes production system with processing paths, presence of equipment for material handling multi-line, level and distribution in a static planning horizon. The proposed AlgoDist method minimizes the total cost of material handling considering the Manhattan distance metric and manages to get the right balance between type constructor algorithms that generate the distribution gradually occupying the positions of the domain location, and type algorithms that promote improvement best solution from an initial distribution of plant.

The heuristic algorithm is quite effective to solve combinatorial optimization problems in multiple rows, with minimal use of computer time.

Finally, the performance of the algorithm has been tested with several case studies and their results compared with those obtained in other works cited in the literature, showing an efficiency of 96%, but with a low computational cost.

**Keywords:** Quadratic assignment problem, Heuristic, Greedy algorithm.

# ÍNDICE

Índice de figuras .....	x
Índice de tablas .....	xii
 CAPÍTULO 1: Introducción.....	1
1.1 Antecedentes .....	2
1.1.1 Análisis de la secuencia de operaciones.....	3
1.1.2 Análisis del diagrama de bloques.....	3
1.1.3 Análisis de carga-distancia.....	4
1.1.4 Costos de comunicación.....	4
1.1.5 Problema de Asignación Cuadrática .....	5
1.1.6 Planeación Sistemática de la Distribución .....	7
1.2 Taxonomía de los problemas de distribución de planta .....	10
1.3 Planteamiento del problema .....	13
1.4 Objetivos .....	14
1.4.1 Objetivo general .....	14
1.4.2 Objetivos específicos.....	14
1.5 Justificación.....	15
 CAPÍTULO 2: Estado del arte de la representación del layout y revisión de técnicas	
2.1. Representación del layout .....	16
2.1.1 Representación discreta.....	17
2.1.2 Representación continua .....	17
2.1.3 Representación cartesiana .....	19
2.2 Revisión de técnicas .....	20
2.2.1 Buffa E.S., Armour G.C. & Vollman T.E., Allocating Facilities with CRAFT .....	20
2.2.2 Ravindra K. Ahuja, James B. Orlin y Ashish Tiwari, A greedy genetic algorithm for the quadratic assignment problem .....	23

2.2.3	Yongzhong Wu & Ping Ji, Solving the Quadratic Assignment Problems by a Genetic Algorithm with New Replacement Strategy .....	24
2.2.4	Josef Kratica, Dusan Tomic, Vladimir Filipovic & Dordje Dugosija, A new genetic representation for quadratic assignment problem .....	25
2.2.5	Berna Haktanirlar Ulutas & Sadan Kultutrel-Konak, An artificial immune system based algorithm to solve unequal area facility layout problem.....	27
2.2.6	Giuseppe Aiello, Giada La Scalia & Mario Enea, A non dominated ranking Multi Objective Genetic Algorithm and electre method for unequal area facility layout problems .....	29
2.2.7	Ravi Kothari and Diptesh Ghosh, Scatter search algorithms for the single row facility layout problem .....	31
2.2.8	M. Adel El-Baz, A genetic algorithm for facility layout problems of different manufacturing environments .....	34
2.2.9	Maghsud Solimanpur & Mehdi A. Kamran, Solving facilities location problem in the presence of alternative processing routes using a genetic algorithm .....	38
2.2.10	Amir Sadrzadeh, A genetic algorithm with the heuristic procedure to solve the multiline layout problem.....	40
2.3	Herramientas para el diseño de distribución de planta.....	45

### CAPÍTULO 3: Un algoritmo greedy para el diseño de distribución de planta multi línea

3.1	Formulación del problema.....	47
3.2	Algoritmo greedy .....	50
3.2.1	Criterio greedy.....	50
3.2.2	La primera pareja de máquinas y su ubicación en el layout.....	51
3.2.3	Métrica de distancia Manhattan .....	52
3.2.4	Algoritmo AlgoDist.....	53
3.3	Ilustración del algoritmo .....	55
3.4	Análisis de complejidad .....	58



CAPÍTULO 4: Experimentos numéricos	
4.1 Hardware y software empleado .....	59
4.2 Casos de prueba.....	59
4.2.1 Caso 1: Curso de Distribución de Planta FII-UNMSM (Salas 2014-2) .....	59
4.2.2 Caso 2: Chase <i>et al.</i> (2009), ejercicio 11, página 249 .....	61
4.2.3 Caso 3: El-Baz (2004), Mak <i>et al.</i> (1998) y Chan & Tansri (1994).....	63
4.2.4 Caso 4: Buffa <i>et al.</i> (1964), Oficina con 10 departamentos .....	65
4.2.5 Caso 5: Red Crystal Glass Products Company (Gaither <i>et al.</i> , 2000).....	66
4.2.6 Caso 6: El-Baz (2004) .....	68
4.2.7 Caso 7: Amir Sadrzadeh (2012) .....	70
4.3 Evaluación comparativa .....	74
 CAPÍTULO 5: Conclusiones y trabajos futuros.....	76
 Referencias bibliográficas .....	77

## Índice de figuras

1.1	Secuencia de operaciones .....	3
1.2	Diagrama de bloques .....	3
1.3	Análisis de carga-distancia .....	3
1.4	Esquema del Planeamiento Sistemático de la Distribución .....	8
1.5	Ejemplo del texto p. 226 (Chase et al., 2009) .....	9
1.6	Naturaleza de los talleres de trabajo y su influencia en el layout.....	10
1.7	Distribución de planta por procesos .....	11
1.8	Distribución orientada al producto .....	12
2.1	Técnicas de representación del layout.....	16
2.2	Rejilla de cuadrículas.....	17
2.3	Representación continua .....	18
2.4	Una distribución de los departamentos de la Oficina .....	20
2.5	Evaluación de la distancia total viajada .....	22
2.6	Algoritmo CRAFT.....	22
2.7	Pseudocódigo del algoritmo genético greedy .....	23
2.8	Pseudocódigo de la nueva estrategia de reemplazo .....	24
2.9	Pseudocódigo para la función objetivo .....	25
2.10	Etapas del algoritmo CSA .....	28
2.11	Marco código de procesos de clasificación .....	30
2.12	Combinación alternada entre permutaciones P1 y P2 .....	32
2.13	Combinación parcialmente coincidente entre P1 y P2 .....	33
2.14	Tipos de distribución de planta .....	34
2.15	Proceso de búsqueda genética .....	35
2.16	Representación de los cromosomas de los tipos de distribución.....	36

2.17	Proceso de búsqueda del Algoritmo Genético .....	38
2.18	Cromosoma, operadores de mutación y casamiento .....	39
2.19	Diagrama de flujo del procedimiento heurístico .....	42
3.1	Una solución del ejemplo .....	48
3.2	Otra solución para el ejemplo .....	49
3.3	Proceso de distribución de las máquinas en el grid de localizaciones.....	50
3.4	Pseudocódigo del algoritmo propuesto AlgoDist .....	54

## Índice de tablas

1.1	Tipificación del problema de investigación .....	14
2.1	Matriz Desde – A para una Oficina .....	21
2.2	Similaridades y diferencias básicas de EA y CSA .....	27
2.3	Cuatro algoritmos de búsqueda dispersa .....	31
2.4	Estrategias de cada propuesta en la aplicación de Algoritmos Genéticos en FLP.....	44
2.5	Resumen comparativo de los métodos: CRAFT y CORELAP .....	46
3.1	Un ejemplo de 6 instalaciones con 4 productos .....	48
3.2	Evaluación del costo para las relaciones del ejemplo .....	48
3.3	Matriz de distancias Manhattan y las localizaciones .....	49
3.4	Datos de entrada del ejemplo .....	55
3.5	Resumen de los cálculos de la ilustración del algoritmo .....	57
4.1	Lista de los productos, rutas de proceso y volúmenes de producción del Caso 1 .....	59
4.2	Matriz de entrada del Caso 1 .....	60
4.3	Iteraciones del algoritmo, función objetiva y la mejor distribución del Caso 1 .....	60
4.4	Matriz de flujos interdepartamentales .....	61
4.5	Matriz de entrada y cuadrícula de 2 x 4 .....	61
4.6	Iteraciones del algoritmo, función objetivo y la mejor distribución del Caso 2 .....	62
4.7	Datos generales del Caso 3 .....	63
4.8	Datos de entrada del Caso 3 .....	63
4.9	Iteraciones del algoritmo, la función objetivo y la mejor distribución del Caso 3 .....	64
4.10	Matriz de entrada y la cuadrícula de distribución del Caso 4 .....	65
4.11	Iteraciones del algoritmo y la función objetivo del Caso 4.....	66
4.12	Lista de productos y los volúmenes de producción del Caso 5.....	66
4.13	Matriz de entrada y cuadrícula de 2 x 5 del Caso 5 .....	67

4.14 Iteraciones del algoritmo y la función objetivo del Caso 5.....	67
4.15 Datos generales del Caso 6. ....	68
4.16 Iteraciones del algoritmo y la función objetivo del Caso 6 .....	69
4.17 Lista de artículos y el volumen de producción del Caso 7 .....	70
4.18 Iteraciones del algoritmo y la función objetivo del Caso 7 .....	71
4.19 Distribución de planta de las mejores soluciones del Caso 7 .....	73
4.20 Comparación entre el AlgoDist propuesto y otros métodos .....	74

# Capítulo 1: Introducción

Por medio de la distribución en planta se consigue el mejor funcionamiento de las instalaciones. Se aplica a todos aquellos casos en los que es necesaria la disposición de unos medios físicos (departamentos, máquinas y otros) en un espacio determinado, ya esté prefijado o no. Su utilidad se extiende tanto a procesos industriales como de servicios. La distribución en planta es un problema ineludible para todas las plantas industriales; por lo tanto, no es posible evitarlo. El solo hecho de colocar un equipo en el interior del edificio ya representa un problema de instalación (Muther, 1961).

La ordenación de las áreas de trabajo se ha desarrollado desde hace muchos años. Las primeras distribuciones las realizaba el hombre que llevaba a cabo el trabajo, o el arquitecto que proyectaba el edificio. Con la llegada de la revolución industrial se transformó el pensamiento que se tenía hacia ésta, buscando entonces los propietarios un objetivo económico al estudiar las diferentes alternativas de distribución.

Cuando una empresa determina la localización de sus instalaciones, la próxima decisión operativa más importante es la distribución de las unidades productivas y logísticas; es decir, la *distribución de los recursos en la planta*, denominado como el problema de distribución en planta.

“El problema de la distribución en planta consiste en localizar la disposición óptima de un grupo de instalaciones sujetas a restricciones cualitativas o cuantitativas” (Shayan *et al.*, 2004).

“Una instalación es cualquier entidad que ocupa espacio, por ejemplo, una persona o grupo de personas, un área de recepción de clientes, la ventanilla de un cajero, una máquina, una estación de trabajo, un departamento, un pasillo o un cuarto de almacenamiento” (Krajewski *et al.*, 2008).

Obsérvese que cada una de estas decisiones afecta, entre otros aspectos, el flujo de materiales, los costos de manejo y mantenimiento, la utilización de los equipos, la productividad de la planta y la efectividad de los trabajadores y la administración. La distribución de los recursos en la planta también afecta la satisfacción del cliente, y debe corresponder a un objetivo específico o a un conjunto de objetivos.

En general, la distribución se planea para minimizar determinado criterio: distancias de recorrido, tiempo total de fabricación, flujo y costo del manejo de materiales, retrasos o reprocesamientos y manejos físicos.

También se presentan situaciones en las cuales la distribución se diseña para maximizar un criterio: espacio de utilización, flexibilidad, eficiencia, satisfacción y seguridad de los trabajadores.

Por consiguiente, determinar el objetivo constituye el primer paso en el proceso de distribución de los recursos en las instalaciones.

El presente siglo representa un interesante contraste con los intentos de la década de los años ochenta para modelar las preferencias de los planeadores de distribución de planta, a través de sistemas masivos de *inteligencia artificial* (Loiola *et al.*, 2004).

Es de gran importancia hacer un reconocimiento total del tema de distribución en planta, puesto que será la base para implementar nuevos procedimientos y técnicas en la ejecución de los procesos productivos, lo que dará como resultado una distribución adecuada y, asimismo, un beneficio óptimo para el mejoramiento continuo de las empresas industriales.

## **1.1 Antecedentes**

Las primitivas distribuciones eran principalmente la creación del hombre en su industria particular; había escasos objetivos específicos o procedimientos reconocidos, de distribución en planta.

Con el advenimiento de la revolución industrial, hace unos 150 años, se transformó en objetivo económico, para los propietarios, el estudiar la ordenación de sus fábricas. Las primeras mejoras fueron dirigidas hacia la mecanización del equipo. Se dieron cuenta también de que un taller limpio y ordenado era una ayuda tangible. Durante los primeros años del siglo pasado la especialización del trabajo empezó a ser tan grande que el manejo de los materiales comenzó también a recibir una mayor atención en lo que se refiere a su movimiento entre dos operaciones. Con el tiempo, los propietarios o sus administradores empezaron a crear conjuntos de especialistas para solucionar los problemas de distribución. Con ellos llegaron los principios que se conocen hoy en día.

En sus inicios, el problema de distribuciones nació con la preocupación únicamente de “secuencia de operaciones” de forma para permitir un mejor flujo de operaciones en una planta (Buffa, 1955). Seguidamente, fueron añadidos los “diagramas de bloque” en donde fueron consideradas las dimensiones del área de instalación. Luego se implementaron los análisis de “carga-distancia”, donde se procura reducir al *mínimo* el movimiento total considerando no sólo la cantidad de *movimientos* interdepartamentales de un producto, sino también las *distancias* sobre las cuales se realizan los movimientos y luego la “minimización de los costos de comunicación”.

### 1.1.1 Análisis de la secuencia de operaciones

El análisis de la secuencia de operaciones desarrolla un buen esquema para la distribución física, al analizar de manera gráfica el flujo de productos que se mueven entre los departamentos; el método fue desarrollado por Buffa en 1955 (Gaither *et al.*, 2000), a quien puede considerarse precursor del planeamiento sistemático de la distribución. La Figura 1.1 representa el arreglo de diez departamentos en una instalación de manufactura. El diagrama desarrollado es un grafo en el que los diferentes departamentos son representados mediante círculos. Cada círculo está unido a los demás mediante líneas que representan el flujo de materiales entre los respectivos departamentos.

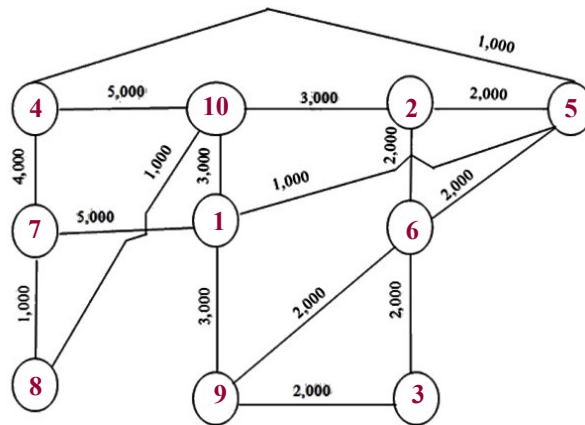


Figura 1.1 Secuencia de operaciones

La generación de este grafo debe tratar de minimizar el número de cruzamientos de las líneas de transporte de materiales, la propuesta se estudia y se rehace mejorándola en los sucesivos intentos. Es decir, demuestra la manera en que pudiéramos determinar la ubicación de los departamentos de operación, en relación uno con otro, *cuando ni la forma ni las dimensiones externas del edificio son factores limitantes*.

### 1.1.2 Análisis del diagrama de bloques

Este análisis establece la forma y dimensiones generales de las instalaciones (rectángulos) y dónde quedarán las fronteras interdepartamentales. De importancia vital para esta distribución física de las instalaciones son las áreas que requiere cada uno de los departamentos (Gaither *et al.*, 2000). La Figura 1.2 representa el arreglo de diez departamentos de la misma instalación anterior (Figura 1.1).

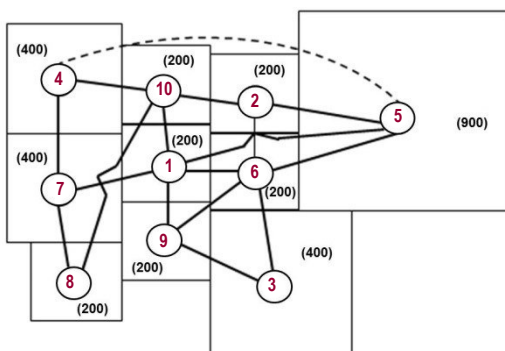


Figura 1.2 Diagrama de bloques

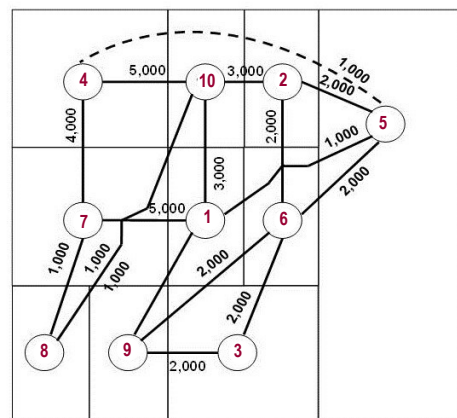


Figura 1.3 Análisis de carga-distancia



El diagrama representa una distribución inicial, pero los departamentos han sido representados por sus áreas relativas indicadas entre paréntesis. En este análisis predomina más la distribución ordenada de áreas de las estaciones de trabajo que el flujo interdepartamental.

Observe que esta disposición física también establece las relaciones generales entre departamentos a través de las líneas, pero el objetivo es establecer la forma y dimensiones generales de las fronteras externas de la planta manufacturera.

### 1.1.3 Análisis de carga-distancia

El análisis de la secuencia de operaciones y el análisis del diagrama de bloques no desarrollan distribuciones físicas óptimas, es decir las mejores, sino solamente buenas distribuciones físicas. No resulta extraño para estos análisis desarrollar dos o más diagramas de bloques alternativos, cada uno de los cuales parece igualmente bueno. El análisis carga-distancia es útil para comparar distribuciones físicas alternas, para identificar cuál tiene el mínimo recorrido del producto o de materiales en un determinado periodo. Es importante contar con la secuencia del proceso de los productos, carga o flujo de materiales y las distancias entre departamentos para cada alternativa de disposición física (Gaither *et al.*, 2000).

La Figura 1.3 representa el arreglo de diez departamentos de la misma instalación de manufactura anterior. En el diagrama desarrollado los departamentos se han representado por sus áreas relativas indicadas entre paréntesis y, además, se establece la carga o flujo interdepartamental mediante un número sobre cada línea.

### 1.1.4 Costos de comunicación

Con frecuencia los departamentos o los centros de trabajo se organizan de modo que se minimicen los costos de comunicación. Por ejemplo, si se presenta un tráfico considerable entre dos departamentos, es probable que la empresa quiera minimizar el costo de recorrido entre dichos departamentos (Konak *et al.*, 2006).

Costo total a minimizar:

$$Z = \sum_{i=1}^{N-1} \sum_{j=i+1}^N f_{ij} c_{ij} d_{ij}$$

N Número de departamentos o centros de trabajo

$c_{ij}$  Costo de mover una unidad de carga por unidad de distancia entre los departamentos  $i, j$

$f_{ij}$  Número de cargas (movimientos) entre los departamentos  $i, j$

$d_{ij}$  Distancia entre los departamentos  $i, j$

Este criterio de distribución de planta se torna complejo cuando la naturaleza y el costo de las comunicaciones varían en los diversos departamentos. Por ejemplo, la comunicación entre dos departamentos puede realizarse sólo para intercambiar diferentes formatos de papel, mientras que entre otros dos pueden transportarse grandes lotes de componentes manufacturados. En estas circunstancias, los costos se asignan a las diversas clases de comunicación y el objetivo es minimizar los costos totales de comunicación.

### 1.1.5 Problema de Asignación Cuadrática (QAP)

El problema de la asignación cuadrática, que se denota por sus siglas en inglés QAP (Quadratic Assignment Problem), fue planteado por Koopmans y Beckmann en 1957 como un modelo matemático para un conjunto de actividades económicas indivisibles.

Las rentabilidades de  $n^2$  pares posibles planta–localización puede establecerse en la forma de una matriz cuadrada, el elemento típico  $a_{ki}$  representa el beneficio esperado de la operación de la planta  $k$  en la localización  $i$ . Una posible matriz de rentabilidad de orden  $n = 4$  sería

$$\text{Planta} \begin{matrix} & \begin{matrix} \text{Localización} \\ 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 25 & \mathbf{20} & 5 & 19 \\ 18 & 3 & \mathbf{0} & 12 \\ \mathbf{22} & 4 & 2 & 12 \\ 16 & 7 & -2 & \mathbf{10} \end{bmatrix} \end{matrix} = [a_{ki}] = A$$

Las entradas en negrita representan una asignación máxima, con un beneficio total de 52 unidades. Tenga en cuenta que el par más rentable, planta 1 en la localización 1, no se produce en una asignación máxima.

La asignación desconocida con la que el problema se refiere, puede ser representado por una matriz de permutación. Esta es una matriz  $P = [p_{ki}]$  de los cuales cada fila y cada columna contiene un solo elemento 1, mientras que todos los demás elementos son 0. La matriz de permutación particular que representa la solución del problema es

$$\text{Planta} \begin{matrix} & \begin{matrix} \text{Localización} \\ 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} = [p_{ki}] = P$$

Cada fila muestra la ubicación asignada a la planta correspondiente por el lugar del elemento de unidad. El beneficio de esta asignación se puede escribir como una suma doble

$$\pi = \sum_{k,i=1}^4 a_{ki} \hat{p}_{ki} = 22 + 20 + 0 + 10 = 52$$

La formulación matemática del problema entonces es encontrar una matriz de permutación  $P = [\hat{p}_{ki}]$  del mismo orden  $n$  como la matriz  $A$ , de modo (omitiendo los límites de suma  $k, i = 1, \dots, n$ ):

$$\sum_{k,i} a_{ki} p_{ki} \leq \sum_{k,i} a_{ki} \hat{p}_{ki}$$

Desde entonces los investigadores han presentado muchas formulaciones de los QAP. Una de las presentaciones más populares es la formulación basada en la permutación (Wu & Ji, 2008). Se asigna  $n$  instalaciones a  $n$  localizaciones con un costo que será proporcional al flujo entre las instalaciones multiplicado por sus distancias. El objetivo es localizar cada instalación en una ubicación, tal que minimice el costo total. Por lo tanto se tiene dos matrices de  $n \times n$ , la matriz de flujos  $A = (a_{ij})$ , y la matriz de distancias  $B = (b_{kl})$ . El QAP en la forma de Koopmans-Beckmann puede ahora escribirse como

$$\min_{\pi \in S_n} C = \sum_{i=1}^n \sum_{j=1}^n a_{\pi(i)\pi(j)} b_{ij}$$

Donde  $S_n$  es el conjunto de permutaciones de  $\{1, 2, \dots, n\}$ . Cada producto individual  $a_{\pi(i)\pi(j)} b_{ij}$  es el costo causado por asignar la instalación  $\pi(i)$  a la localización  $i$  y la instalación  $\pi(j)$  a la localización  $j$ . A veces una instancia QAP con matrices de entrada  $A$  y  $B$  es denotado por  $\text{QAP}(A, B)$ . Si cualquiera de los coeficientes de las matrices  $A, B$  es simétrica,  $\text{QAP}(A, B)$  se denomina como QAP simétrica. En caso contrario,  $\text{QAP}(A, B)$  se dice que es asimétrico.

### Aplicaciones para el Problema de Asignación Cuadrática

En los siguientes ejemplos de aplicaciones se puede observar que resolver este problema para un gran número de instancias es de vital importancia, y a la vez, tratar de resolver el problema mediante técnicas completas puede resultar infactible por el alto número de instancias.

- Resolución de problemas de distribución en instalaciones industriales y hospitales.
- Diseño de centros comerciales donde se quiere que el público recorra la menor distancia para llegar a tiendas de intereses comunes para un sector del público.
- Diseño de terminales en aeropuertos, en donde se quiere que los pasajeros que deban hacer un transbordo recorran la distancia mínima entre una y otra terminal, teniendo en cuenta el flujo de personas entre ellas.
- Procesos de comunicaciones.
- Diseño de teclados de computadoras, en donde se quiere, por ejemplo, ubicar las teclas de una forma tal que el desplazamiento de los dedos para escribir textos regulares sea el mínimo.
- Diseño de circuitos eléctricos, en donde es de relevante importancia saber dónde se ubican ciertas partes o chips con el fin de minimizar la distancia entre ellos, ya que las conexiones son de alto costo.

### **1.1.6 Planeación Sistemática de la Distribución (PSD)**

El Planeamiento Sistemático de la Distribución, que se identifica por sus siglas en inglés SLP (Systematic Layout Planning), fue desarrollado por Richard Muther (1961) como un procedimiento sistemático multicriterio y relativamente simple, para la resolución de problemas de distribución en planta de diversa naturaleza. El método es aplicable a problemas de distribución en instalaciones industriales, locales comerciales, hospitales, etc. Establece una serie de etapas y técnicas que, como el propio Muther describe, permiten identificar, valorar y visualizar todos los elementos involucrados en la implantación y las relaciones existentes entre ellos; cuya estructura esquemática de diez pasos se presenta en la Figura 1.4.

Paso 1. Recolectar y analizar información sobre los procesos que se llevan a cabo en la planta y los productos que se elaboran. Seleccionar una distribución por procesos, por productos o por grupo de tecnología.

Paso 2. Analizar las interacciones que se presentan en las áreas o los departamentos que van a organizarse. Evaluar la conveniencia de colocar áreas o departamentos muy cercanos. El cuadro contendrá un código de evaluación que refleje la conveniencia de colocar juntos dos departamentos. Por ejemplo, el código de evaluación A indica que es absolutamente necesario que estos dos departamentos o funciones estén juntos, mientras que el código X señala que es absolutamente necesario que no estén juntos. También puede añadirse el código de razones. Como se deriva del nombre, estos códigos indican por qué debe asignarse un código particular de evaluación a una pareja de departamentos.

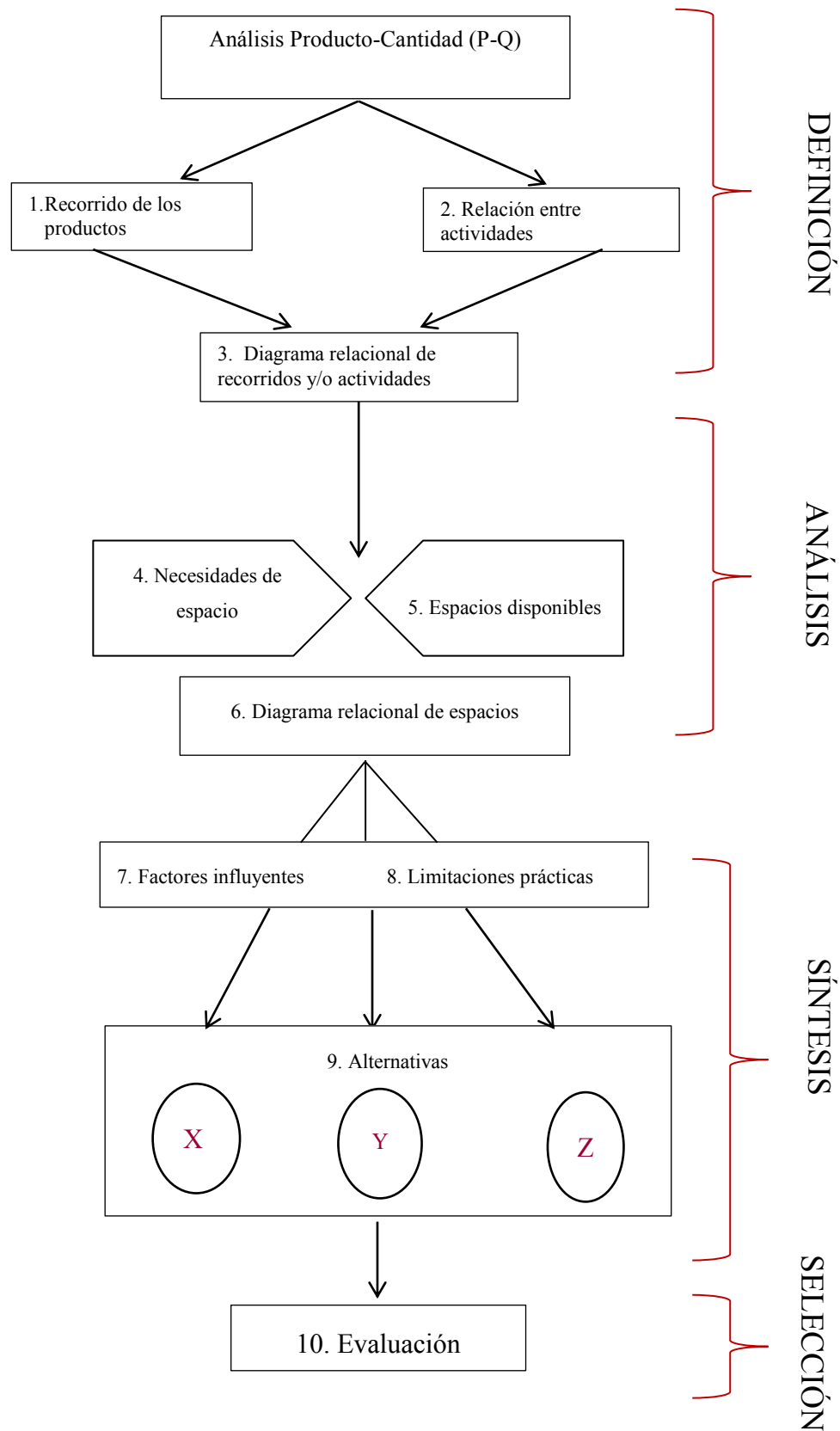
Paso 3. Construir un diagrama de relación de la actividad basado en el cuadro de relación de la actividad y la información relacionada con el flujo de comunicación, materiales, etc.

Básicamente, el diagrama de relación de la actividad es la ilustración de la información contenida en el cuadro de relación de la actividad. En este diagrama, los departamentos se representan mediante rectángulos de igual tamaño. El diagrama de relaciones de las actividades se va ajustando por prueba y error hasta encontrar un patrón satisfactorio de adyacencia.

Paso 4. Estimar el espacio que requiere cada departamento y el total de la distribución.

Paso 5. Determinar el espacio total disponible.

Paso 6. Convertir el cuadro de relación de la actividad en un cuadro de relación del espacio. El tamaño de cada rectángulo debe reflejar el tamaño relativo del departamento representado en este cuadro.



**Figura 1.4** Esquema del Planeamiento Sistemático de la Distribución

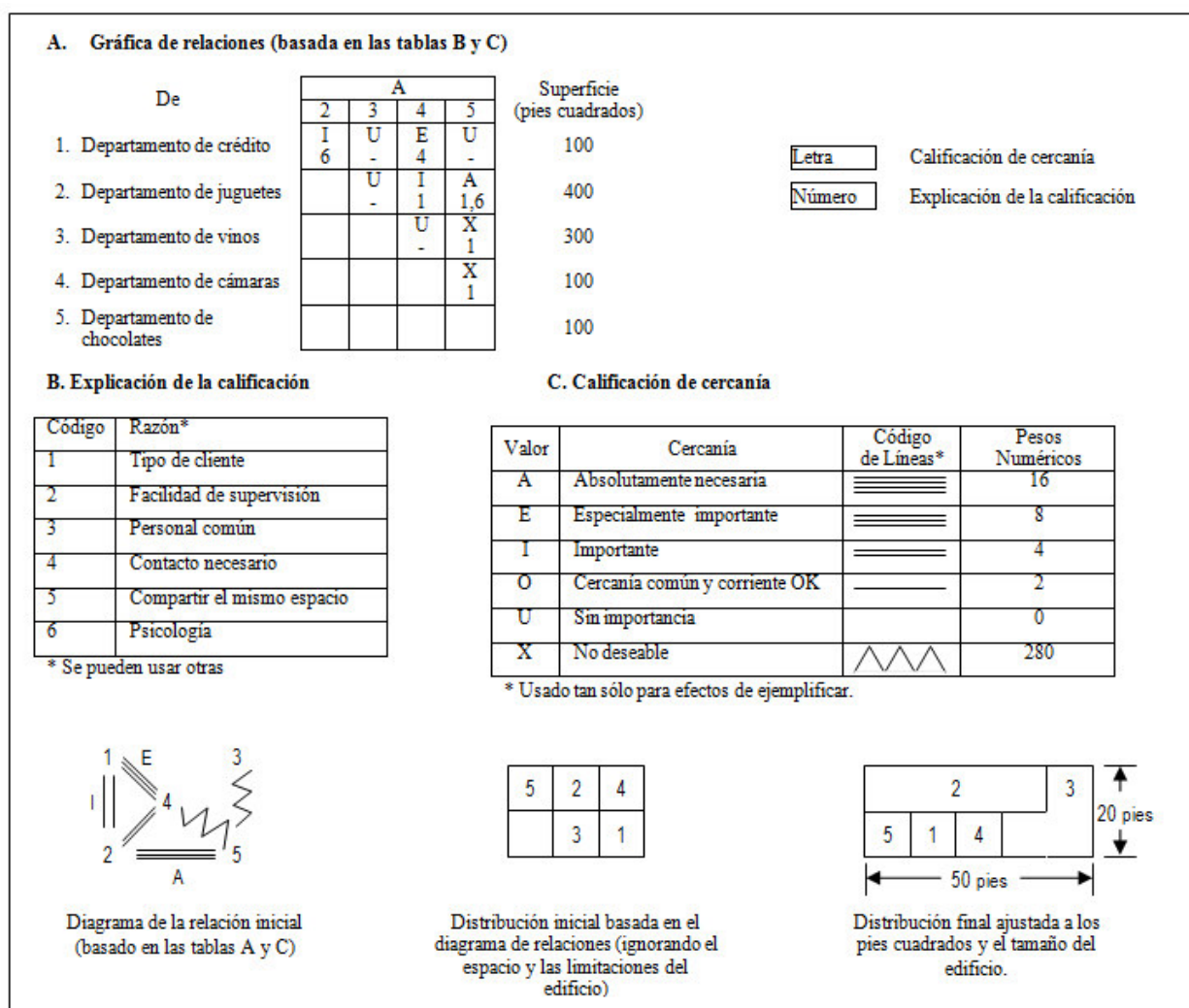
Paso 7. Tomar en cuenta todas las consideraciones de modificación a corto plazo.

Paso 8. No olvidar las limitaciones prácticas y/o restricciones.

Paso 9. Desarrollar y evaluar las distribuciones alternativas. En general, los departamentos se sitúan uno a continuación del otro, empezando por el que tenga el puntaje A más elevado, seguido del que tenga el puntaje X más alto. Cuando esto se ha llevado a cabo, los otros departamentos se sitúan teniendo en cuenta las posiciones polares. Es característico situar en primer lugar todas las relaciones A, luego todas las relaciones E y así sucesivamente.

Paso 10. El SLP es en el fondo un método de ensayo y error; en consecuencia, a menos que se tengan en cuenta todas las distribuciones posibles, no existe manera de saber si se ha logrado la solución óptima. Pero se tiene que evaluar una solución final viable.

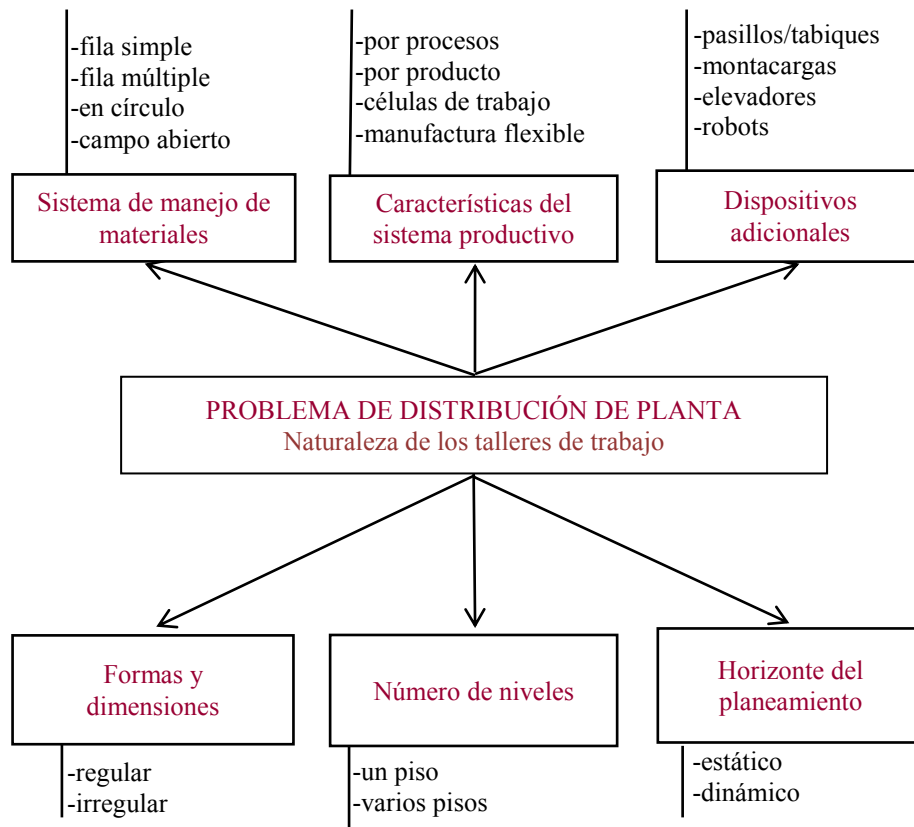
En la Figura 1.5 se ilustra la técnica empleando un sencillo problema de cinco centros de trabajo o departamentos presentado en la p. 226 del texto (Chase *et al.*, 2009).



**Figura 1.5** Ejemplo del texto p. 226 (Chase *et al.*, 2009)

## 1.2 Taxonomía de los problemas de distribución de planta

En la terminología del análisis del proceso productivo, se entiende por taller de trabajo (workshop) a la ubicación física donde se realiza un conjunto particular de tareas. Laura García H. (2011) ha encontrado que en la problemática de distribución de planta, más conocido con el término técnico *layout*, influyen muchos factores según la *naturaleza de los talleres de trabajo*. Luego de la revisión exhaustiva de la literatura, toda la problemática de la distribución en planta se puede esquematizar en función de dichos factores y formas, tal como se muestra en la Figura 1.6.



**Figura 1.6** Naturaleza de los talleres de trabajo y su influencia en el layout

Los factores más predominantes son: sistema de manejo de materiales, características del sistema productivo, dispositivos adicionales, formas y dimensiones de la instalación, número de niveles (pisos) a considerar y el horizonte del planeamiento. A continuación, una breve descripción de cada uno de ellos.

### 1.2.1 Sistema de manejo de materiales

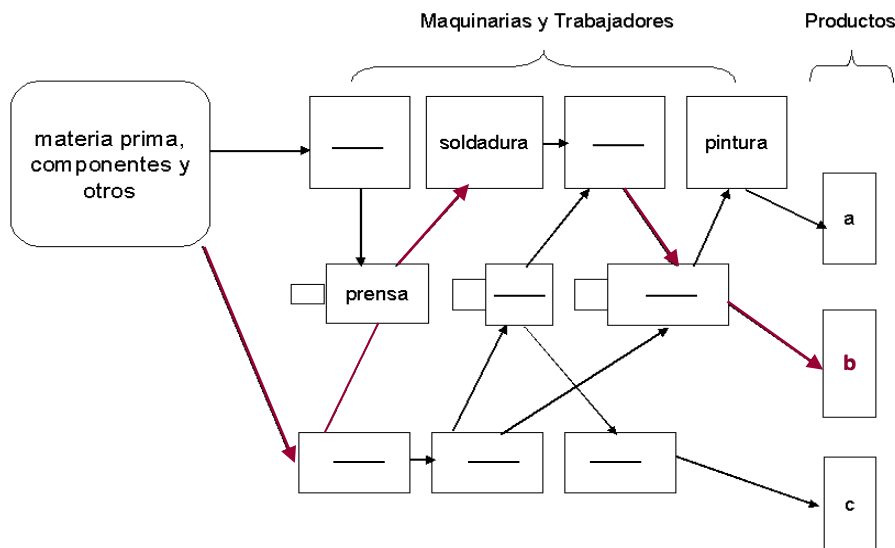
El sistema de manejo de materiales también determina diferentes tipos de problemas de distribución, dependiendo de la existencia de las diferentes rutas de manejo de materiales que existen entre las secciones de trabajo.

### 1.2.2 Características del sistema productivo

Hay diferentes distribuciones de planta dependiendo del volumen y variedad de productos. Los formatos según los cuales se arreglan los departamentos están definidos por el *patrón general del flujo de trabajo* (Chase *et al.*, 2009); existen tres tipos básicos: la distribución por proceso, la distribución por producto y la distribución celular.

Una distribución por proceso es un formato según el cual los departamentos o estaciones de trabajo involucrados en el proceso de producción se agrupan por el tipo de función que realizan, tales como soldadura, tratamiento térmico, pintura y otros, como se puede apreciar en la Figura 1.7.

A continuación, la pieza que se está trabajando avanza, en una secuencia preestablecida de operaciones, de un área a otra, donde se encuentran las máquinas necesarias para cada operación.



**Figura 1.7** Distribución de planta por procesos

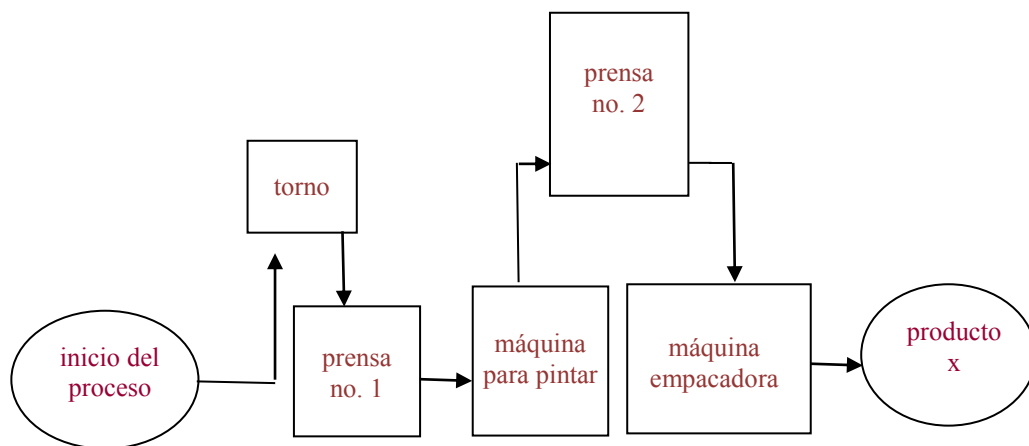
La distribución por proceso, frente a otros tipos de disposiciones, ofrece una gran flexibilidad en cuanto a tipos de productos, siendo su eficiencia dependiente del tamaño del lote producido.

Este tipo de distribución es típica en la industria farmacéutica en la producción de cápsulas, tabletas y tabletas especiales y también en la industria de bebidas.



Una **distribución por producto** se adopta cuando se fabrica un producto estandarizado, por lo común en gran volumen. Cada una de las unidades en producción requiere de la misma secuencia de operaciones de principio a fin, como se puede apreciar en la Figura 1.8. Si se considera en exclusiva la secuencia de operaciones, la distribución es relativamente sencilla, pues se trata de colocar cada operación tan cerca como sea posible de su predecesora. Este sistema permite reducir tiempos de fabricación ya que posee una elevada *eficiencia*.

Las líneas de producción para los zapatos, las plantas de envasados de líquidos o embotelladoras y las empresas de ensamblaje automotriz son todas distribuciones por producto.



**Figura 1.8** Distribución orientada al producto

Una **distribución celular** o distribución de manufactura celular (MC) agrupa máquinas y trabajadores en centros de trabajo (o células), para realizar operaciones en productos que tengan formas y requisitos de procesamiento similares. El objetivo general es ganar los beneficios de la distribución por producto en un sistema de producción por equipos.

La distribución de MC es ahora ampliamente utilizada en la manufactura metálica, en la fabricación de chips de computadoras y en el trabajo de ensamblaje.

**Sistemas de manufactura flexible:** es una estructura con manejo de material automático que mueve unidades individuales de productos entre procesos automatizados. Una gran variedad de productos puede ser manufacturada. Todos estos serán realizados con tiempo de arranque mínimo y muy poca intervención manual.

### 1.2.3 Dispositivos adicionales

Los problemas de distribución de planta pueden incorporar dispositivos adicionales, tales como tabiquerías, elevadores, montacargas y robots.

#### 1.2.4 Formas y dimensiones

Hay dos formas comunes de las instalaciones. Por un lado la forma regular, la cual es usualmente una instalación rectangular. Por el otro lado tenemos la forma irregular, la cual es usualmente un polígono cubriendo un ángulo de 270 grados.

#### 1.2.5 Número de niveles para la distribución

La ubicación de las instalaciones en una planta manufacturera puede abarcar varios pisos o niveles. En este tipo de configuración, el material no sólo fluye horizontalmente en un piso, sino también de un piso a otro en dirección vertical.

#### 1.2.6 El horizonte del planeamiento

Si la demanda del producto cambia en periodos cortos de tiempo, es necesario que la planta manufacturera sea flexible y capaz de adaptarse rápidamente a la nueva configuración. Si el requerimiento de los productos es constante en un periodo de tiempo largo, será recomendable una distribución de planta estática, tal como ocurre en las refinerías de petróleo.

### 1.3 Planteamiento del problema

En una instalación orientada al proceso, como se muestra en la Figura 1.7, se fabrican productos diversificados, los trabajos fluyen en diferentes esquemas de jornada, y es preciso manejar una cantidad relativamente grande de materiales. La fabricación de un lote de producción puede requerir que ésta transite en  $n$  diferentes centros de trabajo (workshop) conforme su proceso de fabricación avanza, desde las materias primas hasta el producto terminado.

Planteamiento:

*Dada una región rectangular como el área de localización de una planta manufacturera de dimensiones fijas de  $W \times H$  y una relación de  $N$  departamentos de área conocida. Cada par ordenado de instalaciones  $(i, j)$  es asociado a un flujo de material  $f_{ij}$  que también es conocido. El problema de diseño de distribución de planta por procesos con multi productos y rutas de procesamiento consiste en particionar la región en  $N$  sub regiones de áreas apropiadas para distribuir ahí las  $N$  instalaciones sin ninguna restricción, con tal de minimizar el costo total del manejo del flujo de materiales.*

El problema de investigación, de acuerdo con la taxonomía mostrada en la Figura 1.5, se tipifica con las características como el indicado en la Tabla 1.1.

**Tabla 1.1** Tipificación del problema de investigación

Tipo de naturaleza	Naturaleza
Características del sistema productivo	por procesos
Sistema de manejo de materiales	fila múltiple
Dispositivos adicionales	montacargas
Formas y dimensiones	regular
Distribución en varios niveles	un piso
Horizonte del planeamiento	estático

## 1.4 Objetivos

### 1.4.1 Objetivo general

El objetivo primordial de la investigación es proponer e implementar un algoritmo goloso para diseñar distribuciones de planta por procesos con multi productos y rutas de procesamiento en las instalaciones industriales, minimizando el costo total del manejo de materiales.

### 1.4.2 Objetivos específicos

Para lograr el objetivo general se deben alcanzar previamente los siguientes objetivos particulares:

- Llevar a cabo un análisis de los métodos de representación del problema de distribución en planta, los diferentes enfoques y planteamientos, y los resultados que se obtienen con cada uno de ellos.
- Estudiar el funcionamiento y el estado del arte de las técnicas analíticas, heurísticas y metaheurísticas empleadas en la resolución del problema de distribución en planta por procesos.
- Revisar los métodos más conocidos de evaluación de algoritmos por *eficiencia* en cuanto al uso de recursos de computación y la *calidad* de la solución obtenida.

- Realizar un profundo análisis de la teoría en la que se asientan los algoritmos golosos y evolutivos, como forma de llevar a cabo una correcta aplicación al problema de distribución de instalaciones industriales.
- Proponer indicadores de la *calidad de los resultados* obtenidos, fundamentalmente desde el punto de vista post instalación, que permitan evaluar la flexibilidad en cuanto a tipos de productos y su dependencia de la ruta de procesamiento.

## 1.5 Justificación

Desde el punto de vista teórico, la distribución de planta por procesos es altamente compleja debido a su carácter multicriterio, y a la naturaleza cuadrática de la función objetivo resultante de la formulación matemática del mismo, es decir, es un problema NP (Non-deterministic Polynomial) completo (Meller *et al.*, 2007).

En un mercado dominado por la competencia y la demanda, que exige gran variedad de productos con ciclos de vida cada vez más cortos, las empresas manufactureras tratan de adaptar sus procesos de fabricación implementando sistemas flexibles que permitan hacer frente a la incertidumbre que esta situación genera. En la práctica, lograr la *eficiencia y la flexibilidad* demandada en un sistema de producción por procesos con multi productos pasa necesariamente por una correcta distribución de los recursos en la planta (Díaz *et al.*, 2008).

$$\text{eficiencia} = \frac{\text{Producción real}}{\text{Capacidad del sistema}} \times 100$$

La *capacidad de diseño* es el resultado de la definición del tamaño de planta, que resulta del diseño del proceso y está limitada por la capacidad de la tecnología implementada.

La *capacidad del sistema* es la que resulta de la reducción de la capacidad de diseño por la mezcla de productos y condiciones de mercado a largo plazo.

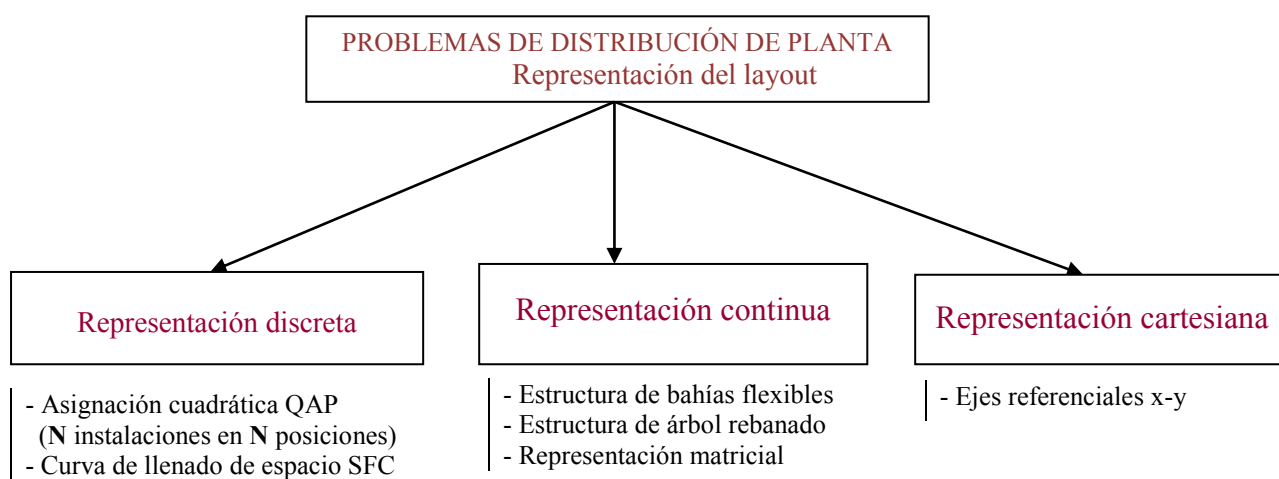
La capacidad de *producción real* está definida como el número de unidades que produce una instalación determinada en un periodo de tiempo, y la definición de los periodos de trabajo.

## Capítulo 2 Estado del arte de la representación del layout y revisión de técnicas

En el presente capítulo se identifica y describe la taxonomía en que se ubica la presente investigación, los antecedentes históricos de los diferentes algoritmos investigados, sus autores, motivaciones, objetivos y aportaciones. Cada uno de los métodos está debidamente resumido y estructurado, de tal modo que puedan distinguirse gráficamente sus estructuras clave.

### 2.1 Representación del layout

Modelar la solución de un problema de distribución de planta implica definir la representación de las diferentes instalaciones en el área de la localización. Como indican los autores García (2011) y García *et al.* (2013), los problemas de distribución de los departamentos pueden modelarse usando varias representaciones, las mismas que pueden clasificarse y agruparse dentro de tres categorías. Al primer tipo se le denomina representación discreta, al segundo representación continua y al tercero representación cartesiana, tal como se esquematiza en la Figura 2.1

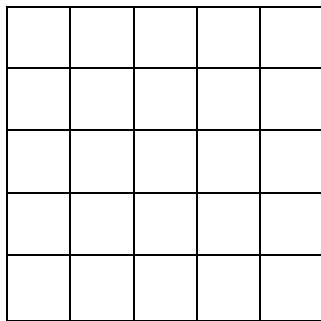


**Figura 2.1** Técnicas de representación del layout

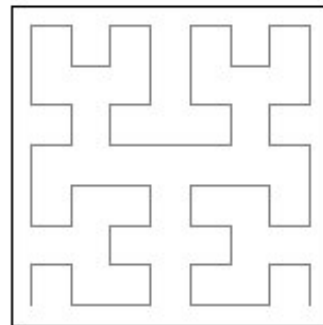
D. José Antonio Diego-Mas (2006), en su tesis considera que las actividades no son simples entelequias adimensionales, sino que se les debe asignar una superficie con un área mínima y con ciertas restricciones en cuanto a la forma y presenta otra clasificación de la representación según el modelo espacial empleado para situar las actividades en el dominio de la ubicación: unidimensionales, bidimensionales, multi plantas y tridimensionales.

### 2.1.1 Representación discreta

Las técnicas que formulan el problema de la distribución de instalaciones bajo modelos discretos, parten de una subdivisión del área de planta en una rejilla de cuadrículas de igual área, tal como se puede observar en la Figura 2.2 a). Las instalaciones serán situadas en la localización asignándoles el número suficiente de subdominios para cubrir sus necesidades espaciales. Si las instalaciones tienen dimensiones iguales y formas regulares, tenemos el problema de ubicar  $N$  instalaciones dentro de  $N$  posiciones, a la que se denomina Problema de Asignación Cuadrática (QAP). Otros autores han aplicado la representación discreta, por ejemplo Buffa *et al.* (1964) y Tate & Smith (1995).



a) Rejilla de cuadrículas



b) Curva de llenado de espacio

**Figura 2.2** Representación discreta

Sin embargo, si las dimensiones son desiguales es necesario adoptar otra estructura adicional. En ese sentido, Balakrishnan *et al.* (2003) usaron otra estructura adicional llamada Curva de Llenado de Espacio (SFC), ver Figura 2.2 b).

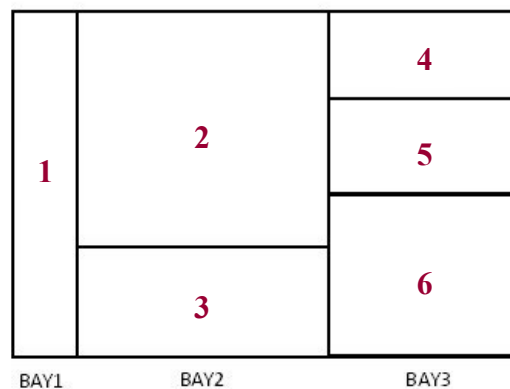
### 2.1.2 Representación continua

De manera general, los modelos continuos construyen las diferentes distribuciones de las instalaciones en el área de la planta mediante el corte recursivo del mismo. Las instalaciones tienen forma rectangular y geométrica flexible, lo que significa que el ratio entre la altura y la anchura del área asignada a la actividad puede variar dentro de ciertos márgenes (restricciones geométricas).

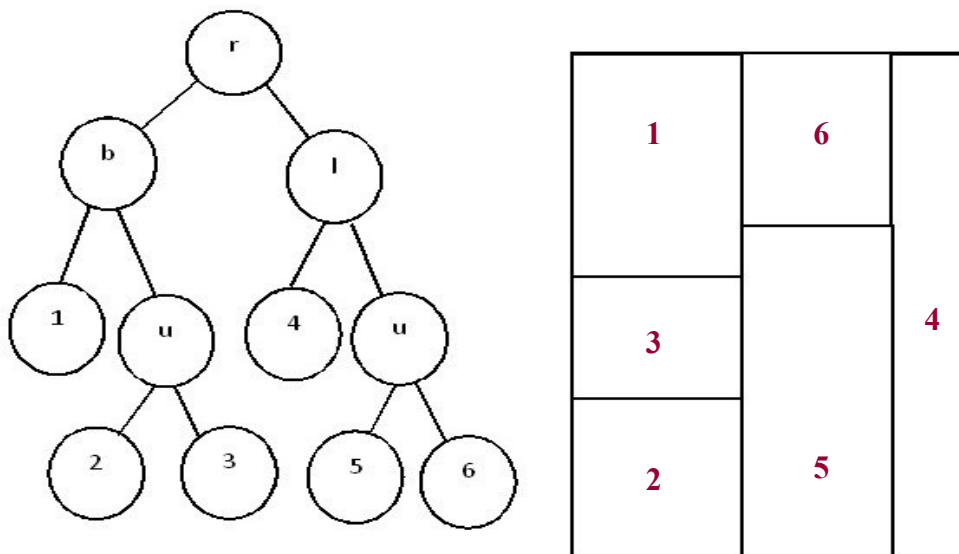
En la representación continua, básicamente existen dos posibles estructuras: las que emplean árboles de corte (Slicing Tree Structure STS) y los modelos de bahías flexibles (Flexible Bay Structure FBS). Últimamente se ha adicionado la representación matricial.

## Estructura de bahías flexibles

La generación de layouts mediante bahías flexibles fue propuesto por Tong (1991), permite la distribución de instalaciones rectangulares de área desigual en áreas habitualmente rectangulares. La localización es dividida en una dirección determinada en “bahías” de anchura variable, tal como se aprecia en la Figura 2.3 a). Dentro de cada una de estas bahías se coloca un número también variable de instalaciones que poseerán igual anchura, y altura proporcional al área requerida por cada una. Actualmente está recibiendo mayor atención por parte de los investigadores, tal como Wong y Komarudin (2010).



a) Representación FBS



b) Representación STS

G	H	B	D	F	C	E	A
1	1	0	0	0	1	1	0
1	3	0	1	2	0	3	1

c) Representación matricial

**Figura 2.3** Representación continua

Un ejemplo de la representación FBS es dado en la Figura 2.3 a). Los departamentos son localizados de izquierda a derecha y de arriba hacia abajo. En la distribución propuesta el departamento 1 está en la primera bahía; las instalaciones 2 y 3 están en la segunda bahía; los departamentos 4, 5 y 6 son ubicados en la bahía tres.

### **Estructura de árbol rebanado**

Los árboles de corte como forma de representación de la distribución en planta, aparecen en el inicio de la década de los noventa tal como se recoge en Tam (1992a, b). Usa la representación del árbol para describir un layout, donde cada hoja representa una instalación y cada nudo interno es el operador del rebanado. Estos operadores pueden ser ‘b’ corte abajo, ‘u’ corte arriba, ‘r’ corte a la derecha y ‘l’ corte a la izquierda, tal como se puede observar en la Figura 2.3 b).

### **Representación matricial**

Esta nueva alternativa de representación de los problemas de distribución de planta ha sido propuesto por García (2011) y consta de una matriz de 3 filas con N columnas (N es el número de instalaciones), tal como se puede apreciar en la Figura 2.3 c). La primera fila indica el orden en el cual las instalaciones han sido distribuidas en el área de superficie disponible (GHBDFCEA). La segunda fila contiene ceros y unos aleatorios que indican los grupos seleccionados de instalaciones a distribuirse. La tercera fila corresponde a las orientaciones: 0 línea vertical izquierda, 1 línea vertical derecha, 2 línea horizontal abajo y 3 línea horizontal arriba.

La matriz de la Figura 2.3 c) representa una distribución de planta similar a la Figura 2.3 a) ver p. 62 en la tesis del autor.

Si el área disponible fuera mayor que el necesario, el espacio sobrante podría permanecer en el centro o en los bordes.

#### **2.1.3 Representación cartesiana**

La distribución de instalaciones bajo este esquema de representación parte del origen (0) del sistema de coordenadas cartesianas. Los departamentos adoptan generalmente formas rectangulares o cuadradas, delimitadas por su ancho ( $a_i$ ) y su alto ( $b_i$ ). La ubicación de una instalación  $i$  queda definida mediante las coordenadas de su centro de gravedad ( $x_i$ ,  $y_i$ ) respecto al origen. Todos los modelos analíticos usan esta representación.



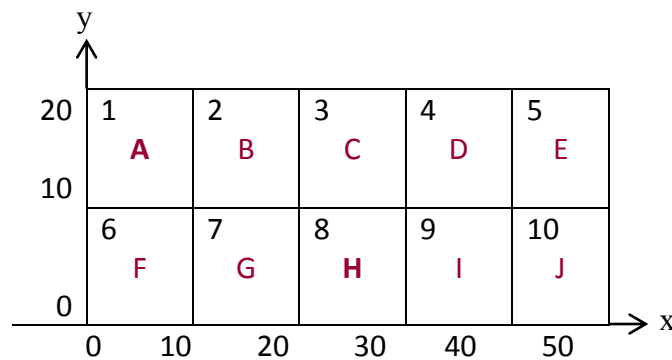
## 2.2 Revisión de técnicas

### 2.2.1 Buffa E.S., Armour G.C. & Vollman T.E., Allocating Facilities with CRAFT, Harvard Business Review, vol. 42, No. 2, March-April, 1964, pp. 136-58

Los autores consideran el problema de arreglar varios departamentos en una planta en un solo piso y dimensiones fijas. Cierta información es necesaria para describir el problema de distribución de planta:

- Número de departamentos,  $n$
- Áreas físicas de cada departamento,  $A_i$  para  $i=1 \dots n$
- Dimensiones físicas de la planta en la cual los departamentos serán ubicados: longitud,  $L(x)$  y ancho  $W(y)$ .
- Flujo de productos entre cada par de departamentos:  $f_{ij}$  para  $i=1 \dots n$  y  $j=1 \dots n$
- Costo del manejo de material entre cada par de departamentos medidas en dólares/pie:  $c_{ij}$  para  $i=1 \dots n$  y  $j=1 \dots n$
- La distancia de un departamento a otro:  $d_{ij}$

Para ilustrar consideran una oficina con 10 departamentos (A, B, C, D, E, F, G, H, I, J), cada instalación tiene un área de 100 pies<sup>2</sup>. Las diez instalaciones son colocadas en un área de 50 pies de longitud (x) por 20 pies de ancho (y). Una distribución cualquiera se muestra en la Figura 2.4.



**Figura 2.4** Una distribución de los departamentos de la oficina

La distribución presentada en la Figura 2.4 es sólo una de muchas distribuciones posibles. Si se asume que los departamentos deben mantener una forma cuadrada, cada permutación de las letras desde A hasta J es una distribución diferente. Entonces habrá  $n!$  permutaciones.

La matriz que describe el flujo entre los departamentos se da en la Tabla 2.1. Este se denomina matriz De – A porque un elemento  $(i, j)$  contiene  $f_{ij}$ , el flujo desde el departamento  $i$  hasta el departamento  $j$ .

**Tabla 2.1** Matriz Desde – A para una oficina

De	A									
	A	B	C	D	E	F	G	H	I	J
A					5				1	9
B				3	4		1			
C				1			9			2
D							6		1	
E						8	3			
F							2	5		
G								6		
H									8	
I										
J										

Los problemas de distribución de planta involucran un criterio para medir la distancia entre los departamentos. Primero asumen que las distancias son medidas entre los centroides o centros de gravedad de los departamentos. Segundo debe especificarse la ruta del viaje; una posibilidad es que el flujo siga una línea recta, esta es la medida euclidiana. Los autores asumen que el flujo viaja en caminos paralelos a los ejes del plano de distribución, esta es la medida rectilínea, también conocida como la distancia Manhattan.

Criterio para comparación:

El flujo y la distancia separados por una diagonal se encuentran registrados en las celdas, ver Figura 2.5. El costo del flujo de i a j se calcula como el producto del costo del manejo del material, el flujo y la distancia entre los departamentos.

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} f_{ij} d_{ij}$$

El costo asociado con cada fila del cuadro se muestra en el lado derecho de la Figura 2.5. El costo de la distribución de planta es la suma de los costos de las filas (2,100).

Los factores  $c_{ij}$  y  $f_{ij}$  son datos del problema que no cambian, pero  $d_{ij}$  dependerá de cada alternativa de distribución de planta.

### Evaluación del layout con el método CRAFT

El ejemplo planteado es bien dificultoso debido al gran número de cálculos requeridos para evaluar el costo de una alternativa en la búsqueda de un arreglo óptimo, en el espacio de  $n!$  permutaciones ( $10! = 3\,628\,800$ ). Como alternativa proponen un método heurístico, que trata de encontrar una buena solución próximo al óptimo; este procedimiento se denomina método CRAFT, cuyo algoritmo se presenta en la Figura 2.6. Las siglas corresponden a la terminología en inglés Computerized Relative Allocation of Facilities Technique.

[illegible]

Debe notarse que cuando dos departamentos son conmutados, solamente los centroides de dichos ambientes son afectados. Por ejemplo, en la Figura 2.4, si el departamento A es conmutado con el departamento H, el centroide de A será (25,5), mientras el centroide de H será (5,15). Todos los otros departamentos permanecen fijos. El costo del layout solamente es afectado por los flujos que entran o salen de los dos departamentos conmutados.

### Figura 2.6 Algoritmo CRAFT

**2.2.2 Ravindra K. Ahuja, James B. Orlin y Ashish Tiwari, A greedy genetic algorithm for the quadratic assignment problem, Computers & Operations Research 27 (2000) 917-934**

El Problema de Asignación Cuadrática (QAP) es uno de los casos de optimización combinatoria clásica y es conocido por sus diversas aplicaciones. El algoritmo que proponen los autores incorpora muchos principios codiciosos en su diseño y por eso denominan a su trabajo como algoritmo genético goloso. Las ideas que incorporan en su algoritmo novedoso incluyen (i) la generación de la población inicial usando una heurística de construcción aleatoria; (ii) nuevos esquemas de casamiento; (iii) un régimen especial de inmigración que promueve la diversidad; (iv) optimización local periódica de un subconjunto de la población; (v) alternamiento entre las diferentes poblaciones iniciales con opciones de eliminación del 50% de dos poblaciones, y (vi) un diseño general que trata de encontrar un equilibrio entre la diversidad y un sesgo hacia los individuos más aptos. En el algoritmo propuesto la ejecución del ciclo repetitivo se llama *proceso* y su descripción de alto nivel se presenta en la Figura 2.7.

```
algoritmo genético;  
inicio  
    obtener población inicial;  
    repetir  
        seleccionar 2 individuos  $I_1$  y  $I_2$  de la población  
        aplicar el operador crossover en  $I_1$  y  $I_2$  para producir  $I_3$ ;  
        reemplazar uno de los 2 individuos  $I_1$  o  $I_2$  por  $I_3$   
        ocasionalmente hacer inmigración;  
    hasta población converja;  
fin
```

**Figura 2.7** Pseudocódigo del algoritmo genético greedy

El algoritmo propuesto es contrastado con todos los casos de referencia de QAPLIB, una biblioteca bien conocida de las instancias QAP. De los 132 casos totales en QAPLIB de diferentes tamaños, el algoritmo genético goloso obtiene la mejor solución conocida para 103 instancias, y en los casos restantes encontraron soluciones dentro del 1% de las soluciones más conocidas.

Los resultados computacionales son presentados en la Tabla 1 del artículo (pp. 928-929), donde se comparan con las 132 instancias del QAPLIB. Los autores sólo presentan el porcentaje de desviación de su algoritmo implementado en tres versiones respecto a la mejor solución conocida, en las columnas señaladas como GA-1, GA-2 y GA-3.

### 2.2.3 Yongzhong Wu & Ping Ji, Solving the Quadratic Assignment Problems by a Genetic Algorithm with New Replacement Strategy, International Journal of Information and Mathematical Sciences 4:3 2008

El trabajo propone un algoritmo genético basado en una nueva estrategia de sustitución para resolver los problemas de asignación cuadrática que son NP-difícil. La estrategia de reemplazo define la forma de seleccionar a los miembros de la próxima generación a partir de los descendientes y los miembros de la última generación. El algoritmo genético implementa dos políticas diferentes de reemplazo, la primera consiste en reemplazar el peor cada  $T_c$  generaciones y la segunda es reemplazar a los similares a sus precursores una vez en cada generación.

El rendimiento del algoritmo genético es prometedor. El significado es que este algoritmo genético es genérico; no se basa en los operadores genéticos de problemas específicos, y se puede aplicar fácilmente a varios tipos de problemas combinatorios.

El esquema del algoritmo genético basado en la nueva estrategia de reemplazo puede describirse a nivel generalizado en la Figura 2.8.

```
Inicio  
  crear población inicial;  
  para cada generación, repetir  
    seleccionar los individuos aleatoriamente desde lugar de apareamiento;  
    aplicar los operadores genéticos para generar descendientes;  
    aplicar la heurística post-crossover en los descendientes;  
    si no es i x  $T_c$  generaciones  
      comparar cada descendiente con su precursor similar y remover el peor;  
    si_no  
      comparar cada descendiente con el peor miembro en la población actual  
      y remover el peor;  
    fin si  
  hasta que algún criterio de parada se cumpla  
fin
```

Figura 2.8 Pseudocódigo de la nueva estrategia de reemplazo

Los resultados computacionales son presentados en la Tabla I del artículo (p. 227), donde se compara con los resultados obtenidos por Ahuja *et al.*, (2000) y con los obtenidos por Drezner (2002). Las instancias del QAPLIB son contrastadas para todos aquellos en el rango de tamaño de 30 a 100 departamentos, la mayoría de las cuales aún no tienen solución óptima. Los valores mostrados son promedios, ya que los autores hacen corridas en el rango de 20 a 200 veces. El nuevo algoritmo genético sólo presenta el porcentaje de desviación respecto a la mejor solución conocida y es registrado en la columna Gap de dicha tabla.

**2.2.4 Josef Kratica, Dusan Tosic, Vladimir Filipovic & Dorde Dugosija, A new genetic representation for quadratic assignment problem, Yugoslav Journal of Operations Research, 21 (2011), Number 2, 225-238**

Los autores proponen una nueva codificación genética para los Problemas de Asignación Cuadrática (QAP) bien conocidos. El enfoque de la nueva representación parte del supuesto de que el QAP es un problema de minimización; entonces, es obvio que la permutación óptima corresponde a pares de instalaciones con flujos altos, ubicados en pares de localizaciones con distancias menores entre ellos; por tanto, este comportamiento tiene que considerarse en un nuevo esquema de codificación. En la nueva representación cada individuo consiste de  $n-1$  genes. La longitud del individuo es  $n-1$  porque ya no es necesario recordar el último elemento de la permutación, cuando todos los otros elementos ya han sido fijados. Cada gen es representado por un entero que corresponde a un elemento de la permutación (no es el índice de una localización asignada a la instalación  $i$ ). Para un individuo codificado la correspondiente permutación para asignar localizaciones a las instalaciones es obtenido por un procedimiento iterativo, cuya codificación se resume en la Figura 2.9.

```
1. S= [ ]; gv := Take_Gene (1) ; pi [ 1 ] := gv+1;
2. for i := 2 to n-1 do begin
3.   gv := Take_Gene (i);
4.   for j := 1 to n do
5.     if not (j in S) then begin
6.       w [j] = F (i,i) * D (j,j);
7.       for k:=1 to i-1 do begin
8.         w [j] := w [j] + F (k,i) * D (pi [k], j) ;
9.         w [j] := w [j] + F (i,k) * D (j, pi [k]) ;
10.      endfor
11.    endif
12.  endfor
13.  Quick_Sort (n-i+1, w, w_index);
14.  pi [i] := w_index [gv+1] ;
15.  S := S ∪ S { pi [i] };
16. endfor
17. pi [n] := {1,2,..,n} \ S ;
```

**Figura 2.9** Pseudocódigo para la función objetivo

Descripción del pseudocódigo:

S representa el set de localizaciones asignadas hasta el momento

pi es la correspondiente permutación de localización

w es el arreglo que guarda las ponderaciones actuales calculadas

$w\_index$  es el vector que denota el índice de  $w$  de acuerdo con los pesos de manera creciente ordenado con la función `Quick_Sort()`.

`Take_Gene(i)` es la función que retorna el valor de  $i$ -ésimo gen.

En cada iteración del procedimiento, para cada localización no asignada su ponderación es calculada como la suma parcial de los productos de flujo y distancias desde esta localización a todas las localizaciones previamente asignadas.

Suponga que se trata de resolver un problema de QAP con  $n=4$ , con las siguientes matrices de flujo y distancia:

$$F = \begin{bmatrix} 0 & 3 & 0 & 2 \\ 2 & 0 & 0 & 7 \\ 3 & 4 & 0 & 0 \\ 5 & 2 & 5 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

En este caso, cada individuo en la población tiene representación genética con longitud 3. Suponga que el individuo tiene la siguiente representación:

2	0	1
---	---	---

Los cálculos de permutaciones del algoritmo para este individuo serán:

- Al empezar, la localización 3 es asignada a la instalación 1,  $Pi[1] = g_{v+1} = 3$
- En la siguiente etapa, donde  $i = 2$ , las localizaciones no asignadas son  $\{1,2,4\}$ . Para cada una de estas localizaciones calculamos sus pesos  $w[1] = 10$ ,  $w[2] = w[4] = 5$ , e inmediatamente ordenamos el vector de índices  $w\_index$  en 2,4,1

Puesto que el valor de  $g_v = 0$ , entonces  $pi[2] = w\_index[1] = 2$

- En la tercera etapa se aplica el mismo procedimiento, de modo que  $pi[3] = w\_index[2] = 1$
- En la última etapa,  $pi[4] = 4$ , la cual es sólo la localización remanente

Los resultados del algoritmo propuesto han sido contrastados con las soluciones de las instancias del QAPLIB. Para cada prueba se ha corrido 20 veces y para todas las instancias se encontró la misma solución óptima. Sin embargo, el objetivo principal de los autores fue desarrollar un método con nueva codificación genética y generar soluciones de alta calidad en tiempo.

**2.2.5 Berna Haktanirlar Ulutas & Sadan Kultutrel-Konak, An artificial immune system based algorithm to solve unequal area facility layout problem, Expert System with Applications 39 (2012) 5384-5395**

Este estudio introduce el Algoritmo de Selección Clonal (CSA) con una nueva codificación y un procedimiento novedoso para cubrir con departamentos ficticios, que son introducidos para completar los espacios vacíos en el área de distribución en planta, con representación de Estructura de Bahías Flexibles (FBS). Los principios fundamentales de los Algoritmos Evolucionarios (EA) y Algoritmos de Selección Clonal (CSA) tienen algunas formas comunes, aunque tienen diferente inspiración biológica; se presenta un resumen de las similitudes y diferencias básicas (ver Tabla 2.2).

**Tabla 2.2** Similaridades y diferencias básicas de EA y CSA

	EA	CSA
Espacio de búsqueda	Set de cromosomas	Set de anticuerpos
Candidatos	Cromosomas	Anticuerpos
Individuos	Cualquiera (vectores reales, cadenas, etc.)	Cualquiera
Tamaño de población	Fijo	Fijo
Rendimiento medida	Aptitudes	Afinidad
Operadores	Selección de cromosomas, mutación y casamiento	Selección de clones, hipermutación
Tasa de mutación	Pequeño (-1%)	Grande (depende del valor de la afinidad del anticuerpo)

Casamiento y mutación son herramientas básicas para crear nuevas soluciones para un estándar EA como los Algoritmos Genéticos (GA), mientras hipermutación y operadores de edición receptor son distintivos para CSA.

En los GA, la selección de los padres reproductores se hace en función de sus aptitudes. Un operador de casamiento se aplica y se crean los nuevos descendientes. Principalmente la diferenciación se consigue por cruce. Después de aplicar el operador mutación, los individuos para la próxima generación son seleccionados desde la población total. En el CSA se toman en cuenta los valores de la afinidad (Función Objetivo) durante la hipermutación. Los anticuerpos mutados son inmediatamente testeados para ser aceptados o rechazados.

La tasa de hipermutación es inversamente proporcional a la afinidad de los anticuerpos. En un estándar GA, la tasa de mutación no cambia para diferentes individuos y es usualmente considerado como un valor pequeño para la población. La característica más notable del CSA es la autoadaptación.



Las etapas del algoritmo CSA-FBS para los FLP de áreas desiguales se presenta en la Figura 2.10.

- Paso 1. Inicialización: Crear una población inicial de  $p$  individuos. Las cadenas de soluciones candidatos son nombrados como los anticuerpos y corresponden a diferentes diseños.
- Paso 2. Selección: Calcular el valor de la función objetivo de los anticuerpos (afinidad) y aplicar la selección con la técnica de la ruleta. Los anticuerpos con alta afinidad tienen mayores probabilidades de ser seleccionados.
- Paso 3. Clonación: Clonar los anticuerpos que se seleccionan a partir de la población inicial. El número de clones en la población clon de anticuerpo puede variar dependiendo de la probabilidad de selección de cada anticuerpo.
- Paso 4. Hipermutación: Aplicar hipermutación, que se detalla en la sección 3.5 (p. 5387), y calcular el valor de afinidad. Si hay una mejora en el valor objetivo, sustituir el anticuerpo mutado; en caso contrario, mantener al individuo actual.
- Paso 5. Edición Receptor: reemplazar un porcentaje predeterminado de anticuerpos que tienen un valor bajo de afinidad con otros nuevos generados aleatoriamente. Los anticuerpos con afinidades más bajas tienen mayor probabilidad de ser reemplazados. Este proceso tiene como objetivo la diversidad en la población y la exploración de nuevas regiones de búsqueda.
- Paso 6. Terminación: Si el criterio de terminación, se considera como el número de soluciones que no son mejoradas hasta el momento, si cumple, detener; de lo contrario, vaya al paso 2.

**Figura 2.10** Etapas del algoritmo CSA

El algoritmo muestra un performance consistente para casos estudiados de 25 problemas prueba.

Los problemas de tamaño 100 y 125 fueron estudiados por primera vez con la estructura FBS. El CSA provee cuatro nuevas soluciones FBS mejores y encontró 16 mejores soluciones para casos FBS. Por eso, los dos problemas de prueba de gran tamaño fueron resueltos por primera vez usando la representación FBS y resultados significativamente mejorados a las soluciones conocidas anteriormente. El estado del resultado global del CSA con representación FBS fue satisfactorio en 95.65% de los problemas de prueba cuando se compara con los mejores resultados de FBS, y 90.90% comparado con las mejores soluciones conocidas que no usan la representación FBS.

El performance del CSA-FBS ha sido contrastado con los problemas de prueba de la literatura y agrupados en dos grupos en función del número de departamentos, que se muestra en las Tablas 2 (p. 5387) y 9 (p. 5389), respectivamente.

Es preciso resaltar el problema de prueba de Armour & Buffa (1963), considerado como un modelo precursor de los UA-FLP de tamaño 20, que se describe en la Tabla 7 del artículo (p. 5388).

**2.2.6 Giuseppe Aiello, Giada La Scalia & Mario Enea, A non dominated ranking Multi Objective Genetic Algorithm and electre method for unequal area facility layout problems, *Expert Systems with Applications* 40 (2013) 4812-4819**

El Problema de Distribución de las Instalaciones de área desigual (UA-FLP) comprende una clase de problemas de optimización extremadamente difíciles y ampliamente aplicables, en diversas áreas del mundo real. Los Algoritmos Genéticos (AGs) han demostrado recientemente su eficacia en la búsqueda de (sub) soluciones óptimas a muchos problemas NP-duro, como los UA-FLP. Un problema principal en tal enfoque está relacionado con la codificación genética y el mecanismo evolutivo implementado que debe permitir la exploración eficiente en el gran espacio de las soluciones y garantizar la convergencia hacia el óptimo. Además, en situaciones reales en las que deben tenerse en cuenta varios factores, el problema de diseño cae en el marco más amplio del problema de optimización multi-objetivo.

Hasta la fecha sólo hay algunos enfoques FLP multi-objetivo y la mayoría de ellos emplean técnicas de optimización simplistas, que finalmente influyen en la calidad de las soluciones obtenidas y los resultados del procedimiento de optimización. En este trabajo, esta dificultad es superada al abordar el problema en dos etapas: en la primera etapa, las soluciones Pareto-óptimo son determinadas empleando el algoritmo genético multi objetivo (MOGA) implementando cuatro funciones de rendimiento, con un procedimiento evolutivo Pareto, siguiendo la estructura general del Algoritmo Genético No dominado (NRGA), y la posterior selección de la solución óptima se lleva a cabo mediante el procedimiento de toma de decisiones multi criterio Electre. Este procedimiento permite al tomador de decisiones expresar sus preferencias sobre la base del conocimiento del conjunto de soluciones candidato.

**Mecanismo de diversificación y procedimiento de ranking**

Un aspecto crucial que afecta drásticamente la convergencia de una MOGA es el procedimiento para la selección de los mejores individuos de la población. Al ser un enfoque basado en el sentido de Pareto, el procedimiento de clasificación aquí utilizado se refiere al grado de dominancia. De acuerdo con este enfoque, primero los individuos no dominados dentro de la población son identificados; se les da el rango 1, y son removidos de la población. Entonces, los individuos no dominados dentro de la población reducida se identifican y se les da el rango 2, seguidos de su expulsión de la población. Este procedimiento se repite hasta que toda la población queda clasificada.

El código de macro correspondiente se presenta en la Figura 2.11.

Las soluciones de menos dominadas, así determinadas, sobreviven para hacer la población de la próxima generación. Hay que señalar que las individuos que pertenecen a la parte delantera no dominada no pueden diferenciarse aún más (y clasificados), a menos que se introduzca un mecanismo de elitismo adicional. Por un lado, esto significa que el tamaño de la población debe ser lo suficientemente grande como para involucrar a todo el conjunto de soluciones no dominadas, y para mantener a la población bien diferenciada; mientras que por otro lado, esto sugiere el empleo de operadores específicos para mantener una buena distribución de soluciones. Por tal razón, el mecanismo de elitismo adicional puede ser introducido; por ejemplo, en NSGA-II el método de comparación-concurrida se utiliza durante el proceso, con el operador comparación-concurrida. El algoritmo que aquí se presenta, por tanto, incluye un mecanismo elitista basado en la distancia y una rutina de control clon que cuenta los clones en la población y opera una mutación recursiva hasta que el elemento mutado es diferente de todos los demás. Esto se realiza hasta que el número de clones es nulo o inferior a un umbral de aceptación preestablecido.

```
i=0, Rank =0, dom=0; Nranked =0;
While (Nranked<N)
    {Increase rank;
    For each element I in the population}
        {dom=0;
        For each j element in the population If (j dominates i) then dom=1;}
        If dom=0
            {assign rank to element i;
            increase Nranked;
            }
        }
    }
```

**Figura 2.11** Macro código del proceso de clasificación

Los objetivos cuantitativos y cualitativos se consideran con referencia al esquema de representación en la Estructura de Árbol Rebanado (STS).

En orden de validar el algoritmo propuesto, consideran el modelo precursor de Armour & Buffa (1963) de 20 departamentos. El costo total del manejo de materiales obtenido es comparado con la solución obtenida por Wanga *et al.*, (2005), según la Tabla 5 del artículo (p. 4816).

**2.2.7 Ravi Kothari and Diptesh Ghosh (2012), Scatter search algorithms for the single row facility layout problem, *P & QM Area*, IIM Ahmedabad, Vastrapur, Ahmedabad 380015, Gujarat, INDIA**

El problema de distribución de instalaciones en una sola fila (SRFLP) es el problema de acomodamiento de instalaciones con longitudes dadas en una línea, con el objetivo de minimizar la suma ponderada de las distancias entre todos los pares de instalaciones. El problema es NP-duro y la investigación se ha centrado en heurísticas para resolver grandes instancias de los SRFLP. En este artículo presentan cuatro algoritmos de búsqueda dispersa para resolver casos de gran tamaño.

Estos algoritmos heredan la estructura básica de la plantilla de búsqueda dispersa presentada en Glover (1998). Cada uno de los algoritmos usa el método de generación diversificada para generar una población inicial que contiene buena calidad y diversas permutaciones, un método de mejoramiento para mejorar las permutaciones en el algoritmo, un método para actualizar el conjunto de referencia para construir y mantener el conjunto de referencia con permutaciones de buena calidad, un método de generación de subconjuntos y un método de solución combinada para actualizar el conjunto de referencia con nuevas permutaciones. Presentan dos métodos de generación diversificada llamados DIV-1 y DIV-2 y dos métodos de solución combinada llamados método de combinación alterna y método de combinación parcialmente coincidente, cuyas combinaciones dan lugar a cuatro algoritmos de búsqueda dispersa, tal como se muestra en la Tabla 2.3.

**Tabla 2.3** Cuatro algoritmos de búsqueda dispersa

Algoritmo	Métodos de generación diversificada	Métodos de solución combinada
SS-1A	DIV-1	Combinación alternada
SS-1P	DIV-1	Combinación parcialmente coincidente
SS-2A	DIV-2	Combinación alternada
SS-2P	DIV-2	Combinación parcialmente coincidente

*Métodos de generación diversificada*

Este elemento de búsqueda dispersa determina la calidad de permutaciones en el *conjunto de referencia* y garantiza la diversificación en el proceso de búsqueda. Utiliza el concepto de distancias de desviación para medir la diversificación en un conjunto de permutaciones. En un SRFLP, una permutación es idéntica a la permutación obtenida invirtiendo las posiciones de todas las instalaciones de la permutación.

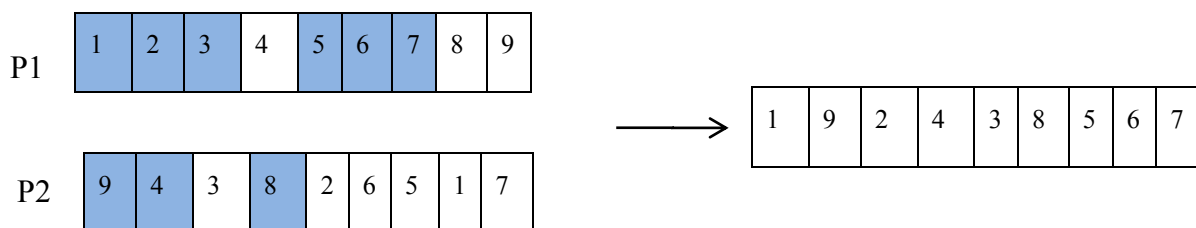
La distancia entre dos permutaciones  $\Pi_1$  y  $\Pi_2$  se define como el mínimo de la distancia de desviación entre  $\Pi_1$  y  $\Pi_2$  y la distancia de desviación entre  $\Pi_1$  y la permutación obtenida mediante la inversión de  $\Pi_2$ .

Utilizando esta medida de distancias de desviación, desarrollan dos métodos DIV-1 y DIV-2 para generar un conjunto diversificado de buenas permutaciones, que sirven como la población inicial de los algoritmos de búsqueda dispersa. Ambos métodos requieren de una permutación semilla inicial de instalaciones en F como insumo para la generación de la población inicial.

#### *Métodos de solución combinada*

Estos métodos combinan, en una nueva permutación, las permutaciones de los subconjuntos generados por el método de generación de subconjuntos. La nueva permutación así generada se somete al método de mejoramiento hasta que converge en un óptimo local. La permutación localmente óptima es, finalmente, añadida al conjunto de referencia, o se descarta basado en su costo. En los algoritmos de búsqueda dispersa utilizan dos métodos de combinación, uno de los cuales es puramente determinista en la aplicación y el otro que tiene un componente aleatorio.

El *primer método* de combinación es el método de combinación alterna. Se crea una nueva permutación al seleccionar alternativamente las instalaciones de la primera y segunda permutación en el subconjunto, omitiendo las instalaciones que ya han sido ubicadas en la nueva permutación. La operación se muestra en la Figura 2.12 con las células sombreadas indicando las instalaciones seleccionadas para su inclusión en la nueva permutación, en las posiciones correspondientes.



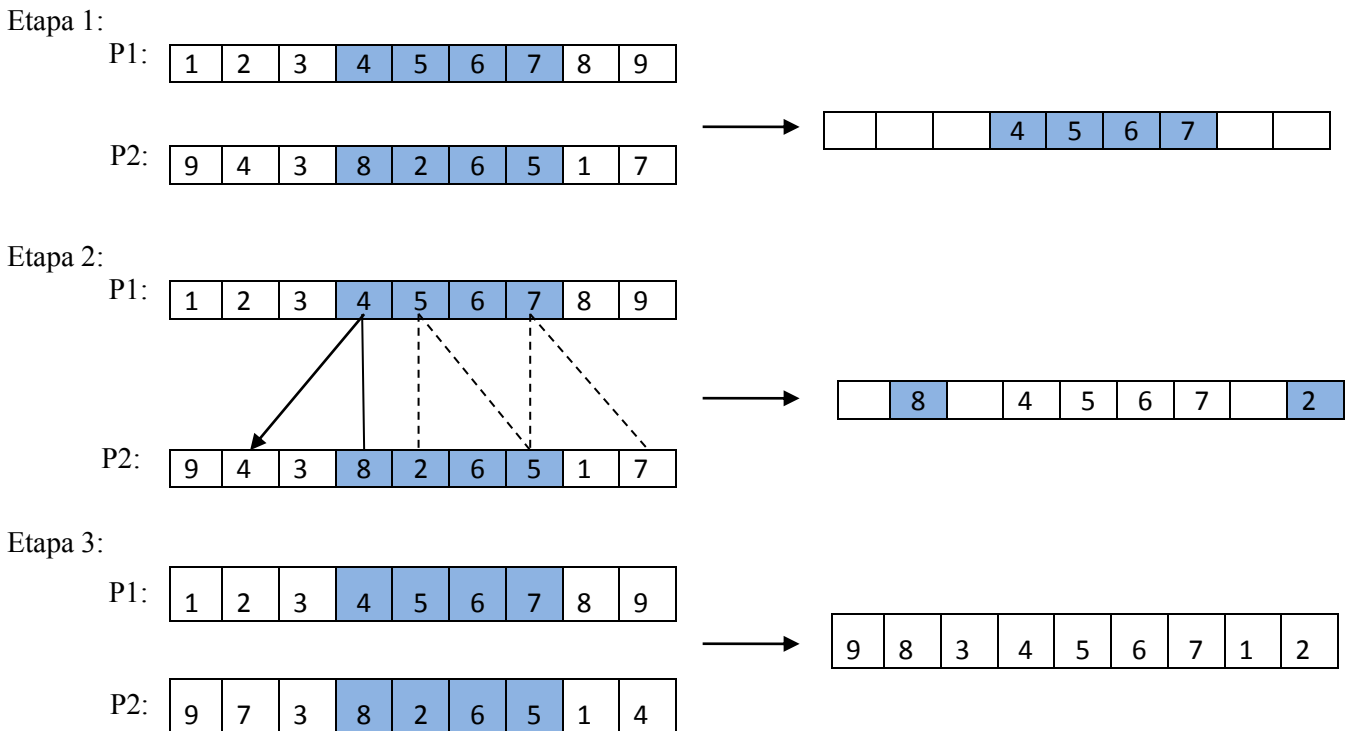
**Figura 2.12** Combinación alternada entre permutaciones P1 y P2

El *segundo método* de combinación es el método de combinación parcialmente coincidente. Se le conoce como mecanismo de cruce PMX en los algoritmos genéticos, y es el operador de combinación más ampliamente utilizado para problemas de permutación. Por conveniencia notacional, denominan a las dos permutaciones como subconjuntos P1 y P2 y la permutación combinada como P3, tal como se muestra en la Figura 2.13. Este método de combinación trabaja como sigue:

Etapa 1: Seleccionar un segmento de instalaciones al azar en P1 (segmento de combinación) y copiarlo a P3 en las mismas localizaciones.

Etapa 2: Empezando en el primer punto de la combinación (primera instalación en el segmento de combinación) buscamos las instalaciones en ese segmento de P2, las cuales aún no han sido copiadas a P3. Para cada una de estas instalaciones  $i$  en P2 buscar en P3 para ver cuál instalación  $j$  ya ha sido copiada en su lugar desde P1 y ubique la instalación  $i$  en P3 en la misma posición como la que ocupaba la instalación  $j$  en P2 (la instalación 8 se ubica en la posición 2 en P3, que es el lugar que ocupa en P2 la primera instalación del segmento de combinación en P1).

Etapa 3: Luego de haber tratado con las instalaciones del segmento de combinación, el resto de las localizaciones en P3 son completadas en orden, con las instalaciones desde P2 para obtener finalmente una permutación completa, como se muestra en la Figura 2.13.



**Figura 2.13** Combinación parcialmente coincidente entre P1 y P2

Los autores han realizado experimentos computacionales para comparar el desempeño de sus 4 algoritmos con los resultados más conocidos en la literatura, para tres series de casos de referencia. La experiencia computacional indica que los 4 algoritmos de búsqueda dispersa son superiores a otros algoritmos para el SRFLP conocidos en la bibliografía. En comparación con los resultados reportados en la literatura publicada, entre los 43 casos de referencia, ellos coinciden con las mejores soluciones conocidas en 17 instancias y obtienen mejores soluciones en las otras 26 instancias con tiempos de ejecución razonables.

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

El problema de distribución en planta tratado consiste en la asignación de M maquinas a N localizaciones en una planta manufacturera. Durante el proceso de producción, los materiales fluyen desde una máquina a otra hasta que el proceso se completa. Siendo el objetivo minimizar el costo del manejo de materiales en el sistema. Para determinar el costo del manejo de materiales de una posible distribución en planta, se requiere los volúmenes de producción, rutas de procesamiento y la tabla de costos que califica la distancia entre pares de máquinas localizadas. La siguiente notación es usada para desarrollar la función objetivo:

Fij cantidad de flujo de material entre las máquinas i y J (i, j= 1, 2, ..., M)

Cij costo unitario del manejo de material entre las localizaciones de máquinas i y j (i, j= 1, 2, ..., M)

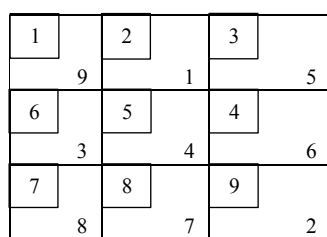
Dij distancia rectilínea entre las localizaciones de máquinas i y j.

C costo total del manejo de material.

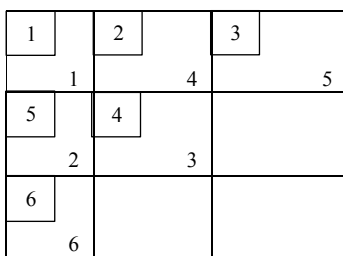
La función del costo total es definida como:

$$C = \sum_{i=1}^M \sum_{j=1}^M F_{ij} C_{ij} D_{ij} \quad (\text{p. 235})$$

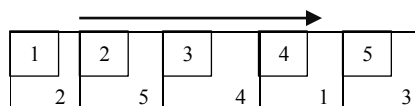
El trabajo considera varios patrones del flujo de materiales: distribución por taller de procesos, distribución de flujo en línea simple, distribución en multi línea y distribución en bucle cerrado. Ver Figura 2.14 a), b), c) y d), respectivamente.



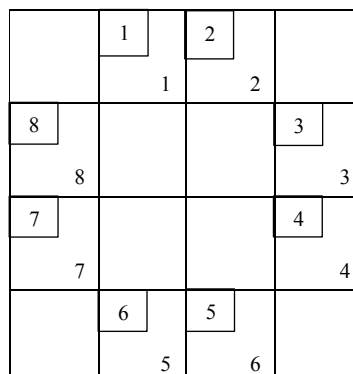
(a) Process shop layout



c) Multi-line layout



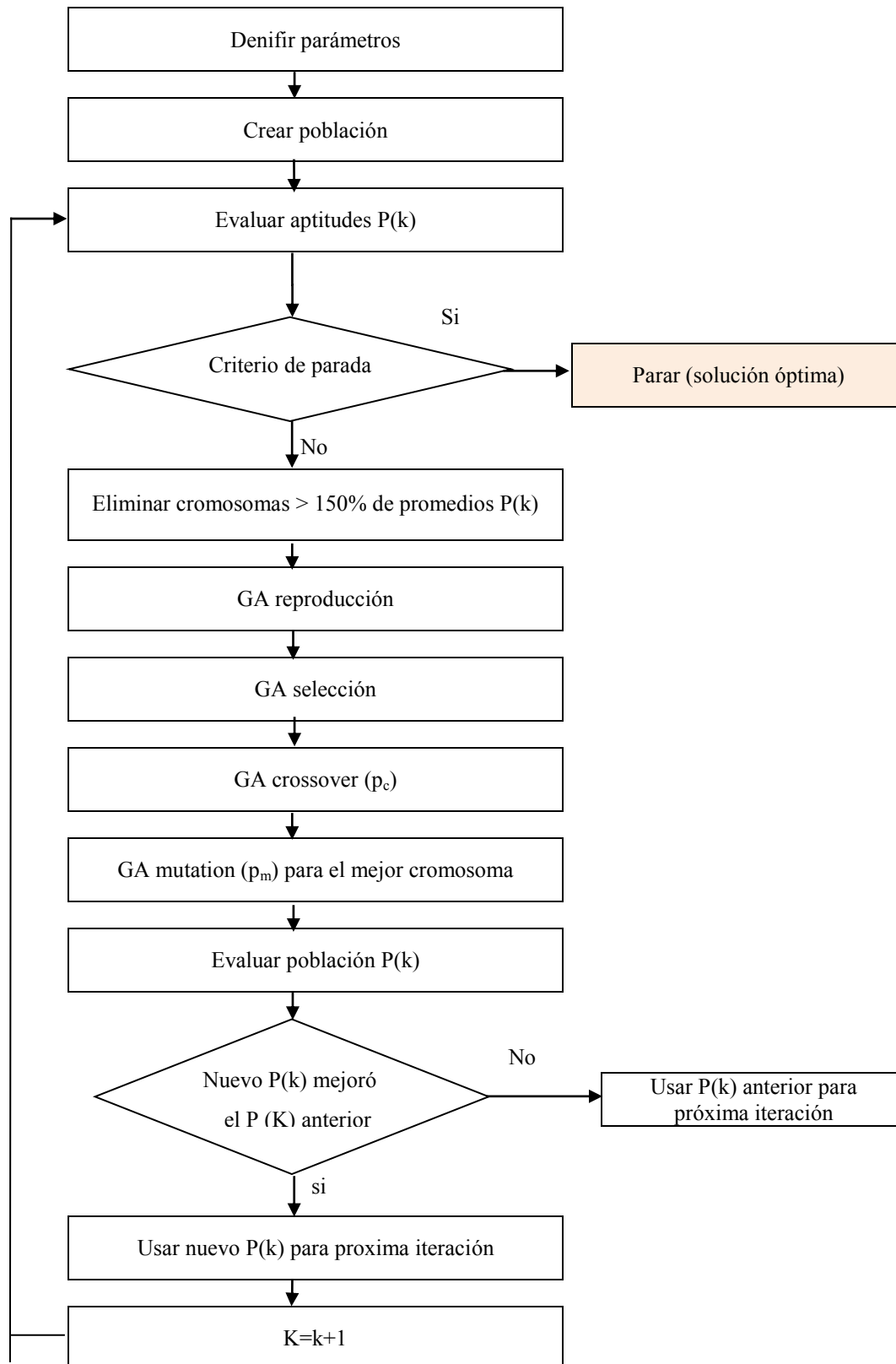
(b) Flow (Single) Line layout



(d) Closed-loop layout

**Figura 2.14** Tipos de distribución de planta

El Algoritmo Genético propuesto trata diferentes configuraciones de manufactura para resolver los FLP, cuyo esquema del proceso de búsqueda genética se presenta en la Figura 2.15.



**Figura 2.15** Proceso de búsqueda genética



Descripción detallada del diagrama de flujo del “Proceso de búsqueda genética”.

1. Generar aleatoriamente una población inicial de cromosomas con un tamaño de población  $P$ .
2. Evaluar cada cromosoma en la población, de acuerdo con la ecuación del costo del manejo de materiales.
3. Determinar el rendimiento promedio para toda la población.
4. Usar la estrategia de elitismo para fijar el mejor número potencial de cromosomas eliminando los peores números de cada generación y guardando los mejores números para las generaciones subsiguientes, el número total de cromosomas ( $P$ ) es constante para la economía y eficiencia computacional, el promedio de los cromosomas totales actúa como una guía para los cromosomas que serán eliminados y los cuales consiguen reproducirse en la próxima generación. Este proceso se aplica para eliminar miembros con valores de rendimiento  $P_{(k)}$  mayor que 1.5 veces el promedio de cromosomas y guardando el mejor número de cromosomas.
5. Aplicar la técnica de selección de Monte Carlo para seleccionar cromosomas padre desde la población actual. Este es usado para escoger aleatoriamente los padres para el casamiento y mutación.
6. Aplicar los operadores de casamiento y mutación para generar una nueva población basada en los valores probabilísticos de casamiento y mutación ( $p_c$  y  $p_m$ ). El resto de la población es completada desde las poblaciones anteriores, las cuales tienen los valores de mejores rendimientos.
7. Revisar el criterio de parada preestablecido. Si el criterio es encontrado, la búsqueda termina. En caso contrario procede a la próxima generación y va al punto 2.

La técnica del AG requiere de una cadena que represente el cromosoma de los tipos del layout, la que se presenta en la siguiente Figura 2.16.

9	1	5
3	4	6
8	7	2

(a) Process shop layout

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

(c) Flow-line layout

1	4	5
2	3	
6		

(e) Multi-line layout

	1	2	
8			3
7			4
	5	6	

(g) Closed-loop layout

9	1	5	6	4	3	8	7	2
---	---	---	---	---	---	---	---	---

(b) Chromosome of process shop layout

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

(d) Chromosome of flow-line layout

1	4	5	e	3	2	6	e	e
---	---	---	---	---	---	---	---	---

(f) Chromosome of multi-line layout

e	1	2	e	3	e	e	8	7	e	e	4	e	6	5	e
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(h) Chromosome of closed-loop layout

**Figura 2.16** Representación de los cromosomas de los tipos de distribución

### Operador Crossover:

El parámetro de casamiento  $P_c$  es la probabilidad de aplicar el Operador Crossover al cromosoma. El resto del cromosoma produce los cromosomas descendientes, idénticos a sus padres. Por otro lado, el cromosoma seleccionado para casamiento será cruzado para producir dos cromosomas descendientes usando dicho operador. En este trabajo para el proceso de casamiento se considera un par de cromosomas padres ( $P_1$ ,  $P_2$ ) mostrados abajo:

$P_1$	1	4	7	5	2	8	3	6	9
$P_2$	4	1	6	7	5	2	9	3	8

Primero se selecciona dos números aleatorios al ser alineados en la cadena de los pares. Supongamos que los dos números aleatorios en este ejemplo son 4 y 7. Los genes con bordes de doble línea dentro de la sección de corte, por ejemplo (5, 2, 8, 3) en  $P_1$  y (7, 5, 2, 9) en  $P_2$  son intercambiados, de manera que la porción del código genético  $P_1$  es transferido a  $P_2$  y viceversa. La estructura del cromosoma resultante queda así:

$P_1^-$	1	4	7	7	5	2	9	6	9
$P_2^-$	4	1	6	5	2	8	3	3	8

En esta etapa, varios genes ya existen en más de una posición en el cromosoma resultante (por ejemplo 7, 9 en  $P_1^-$  y 3, 8 en  $P_2^-$ ).

La efectividad del AG propuesto es evaluado con tres ejemplos numéricos tomados de la literatura.

Ejemplo I: El caso es tomado de Chan & Tansri (1994) y comparado con el trabajo de Mak *et al.*, (1998) que usan el mismo ejemplo para evaluar su trabajo, que consta de 9 instalaciones y se distribuye en un grid de 3 x 3. Obtiene un mejor resultado cuyo costo del manejo de materiales es \$ 4,818.

Ejemplo II: El tema es tomado de Kazerooni *et al.*, (1996) y los resultados son comparados con los trabajos de Chan & Tansri (1994) y Mak *et al.*, (1998). El caso consta de 24 departamentos que procesan 38 productos y se distribuyen en un grid de 5 x 6 localizaciones. Obtiene como resultado \$ 11,862 que es el mejor de los trabajos comparados.

Ejemplo III: Este caso es propio del autor, que consta de 12 instalaciones con 18 productos y se aplica a una distribución de flujo en línea con multi productos, es decir en un grid de 1 x 12. El costo del manejo de materiales obtenido es de \$ 11,440.

Los resultados muestran la efectividad del AG propuesto como una herramienta para resolver los FLP.

**2.2.9 Maghsud Solimanpur & Mehdi A. Kamran, Solving facilities location problem in the presence of alternative processing routes using a genetic algorithm, Computers & Industrial Engineering 59 (2010) 830-839**

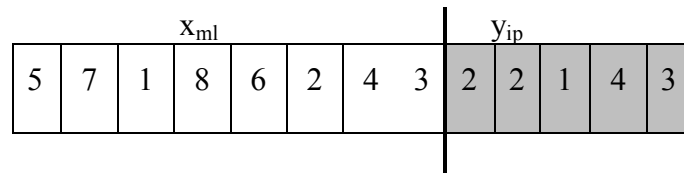
El problema de la localización de instalaciones trata con la optimización de la ubicación de instalaciones de manufactura, tales como máquinas, departamentos, etc., en una planta industrial. El problema afecta mayormente el performance del sistema de producción. Se asume en este artículo que hay múltiples productos a ser fabricados en varias máquinas. Rutas de procesamiento alternativo son consideradas para cada producto y el problema es determinar la ruta de procesamiento de cada producto y la localización de cada máquina, de manera que minimice la distancia total viajada por los materiales dentro del piso de fabricación.

El esquema del proceso de búsqueda genética propuesto es resumido en la Figura 2.17.

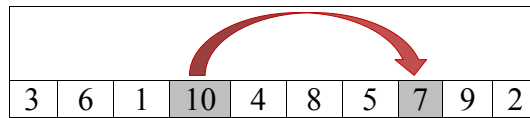
1. Generar aleatoriamente una población inicial de cromosomas con un tamaño de población.
2. Evaluar cada cromosoma en la población de acuerdo con la distancia viajada por los productos.
3. Seleccionar los padres usando la técnica de la ruleta y luego aplicar los operadores de mutación y casamiento para seleccionar los progenitores para producir nuevos descendientes y evaluarlos a esta nuevamente.
4. Seleccionar y producir la nueva generación desde los progenitores y descendientes usando la técnica de la ruleta y la política de elitismo.
5. Revisar el criterio de parada pre-especificado para el algoritmo. Si el algoritmo encuentra un número predefinido de iteraciones, la búsqueda del proceso termina, en caso contrario el algoritmo regresa al punto 3.

**Figura 2.17** Proceso de búsqueda del Algoritmo Genético

Para poder manejar las rutas de procesamiento alternativo los autores proponen una representación especial del cromosoma, la misma que lo divide en dos segmentos. La primera parte del cromosoma es hecho sobre  $M$  genes y muestra cómo  $M$  máquinas son asignadas en  $M=L$  localizaciones ( $x_{ml}$ ). El segundo segmento es construido sobre  $P$  genes que indica la ruta de procesamiento para cada producto ( $y_{ip}$ ), tal como se muestra en la Figura 2.18 a). Del mismo modo se presentan los operadores especiales de mutación y casamiento en las Figuras 2.18 b) y 2.18 c).



a) Primer y segundo segmento del cromosoma



b) Operador de mutación

c1) Casamiento en el primer segmento

	Crossover point	Second segment
Parent I	5 7 1   8 6 2 4 3	2 2 1 4 3
Parent II	4 8 7   1 5 3 2 6	1 2 1 3 1
Offspring I	5 7 1   4 8 3 2 6	2 2 1 4 3
Offspring II	4 8 7   5 1 6 2 3	1 2 1 3 1

(c2) Casamiento en el segundo segmento

	Crossover point
Parent I	5 7 1 8 6 2 4 3   2 2 1 4 3
Parent II	4 8 7 1 5 3 2 6   1 2 1 3 1
Offspring I	5 7 1 8 6 2 4 3   2 2 1 3 1
Offspring II	4 8 7 1 5 3 2 6   1 2 1 4 3

c) Operador de casamiento

**Figura 2.18** Cromosoma, operadores de mutación y casamiento

La efectividad del AG propuesto es evaluado con siete ejemplos numéricos propios creados aleatoriamente. En orden de comparar sus experimentos numéricos, los 7 casos los resuelve con el software Lingo 8.0 y se detalla en la Tabla 5 del artículo (p. 839). Para el ejemplo 1 se obtiene el mismo valor de 851,700.

Los resultados muestran que el AG propuesto es efectivo y eficiente en este tipo de problemas.

**2.2.10 Amir Sadrzadeh, A genetic algorithm with the heuristic procedure to solve the multi-line layout problem, *Computers & Industrial Engineering* 62 (2012) 1055-1064**

El artículo presenta una metaheurística basada en el AG para resolver el FLP de un sistema de manufactura, donde el patrón de flujo de materiales en distribuciones multi línea es considerado con multi productos. La técnica de codificación matricial ha sido usada para representar los cromosomas con objeto de minimizar el costo total del manejo de materiales. El algoritmo propuesto produce un arreglo especial denominado *tabla de descendientes* con los datos correspondientes a los valores de los flujos y costos, ordenados de mayor a menor. La tabla generada es usada para crear una representación esquemática de las instalaciones, las cuales se utilizan para generar heurísticamente la población inicial de individuos y manejar los operadores de casamiento y mutación, también heurísticamente.

Esta nueva propuesta de distribución de planta en multi línea se puede graficar en la siguiente figura para 8 instalaciones que se distribuyen en una cuadrícula de 12 (3 x 4) localizaciones. Los números en la figura se refieren a los departamentos y las celdas se refieren a las localizaciones (p. 1056):

	6	2	5
1	4	7	
	8	3	

Según el autor, los algoritmos genéticos convencionales presentados en los trabajos previos utilizan:

- Codificación popular del cromosoma (p. 1056):

e	6	2	5	e	7	4	1	e	8	3	e
---	---	---	---	---	---	---	---	---	---	---	---

- El criterio aleatorio para generar la población inicial y uso de los operadores de casamiento y mutación.

Para generar la población inicial en el procedimiento aleatorio, cada permutación de los departamentos puede seleccionarse como un cromosoma válido, pero sin considerar el factor costo.

En el procedimiento aleatorio de casamiento, el corte se realiza para cada pareja de padres con los cromosomas codificados en línea simple, dos enteros al azar son generados para indicar el inicio y el fin de la sección de corte.

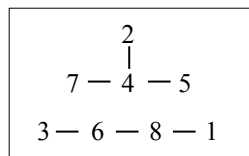
El operador aleatorio de mutación intercambia los genes de cada cromosoma al azar, que no garantiza una dirección favorable hacia la solución óptima.

En contraste con los esquemas tradicionales de los AG, propone un nuevo procedimiento heurístico que consta de once pasos cuyo diagrama de flujo se presenta en la Figura 2.19 y que se detalla a continuación:

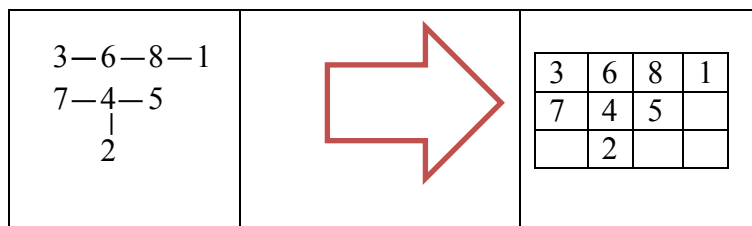
Paso 1. Producción de la tabla de descendientes, ver Tabla 1 del artículo (p. 1056):

Ponderación ( $f_{ij}c_{ij}$ )	Relaciones
<b>100</b>	4-7
<b>92</b>	3-6
<b>75</b>	4-5
<b>53</b>	2-4
<b>45</b>	6-8
<b>38</b>	1-8

Paso 2. Representación esquemática del layout (p. 1056):



Paso 3. Generación heurística de la población inicial: se generan P cromosomas (p. 1057)



Paso 4. Agrupamiento de las instalaciones.

Las instalaciones con relaciones de cercanía fuertes son ubicadas en el mismo grupo. Para el caso mostrado se agrupan en dos categorías: [2, 4, 5, 7] y [1, 3, 6, 8] constituyendo los grupos 1 y 2.

Paso 5. Selección I.

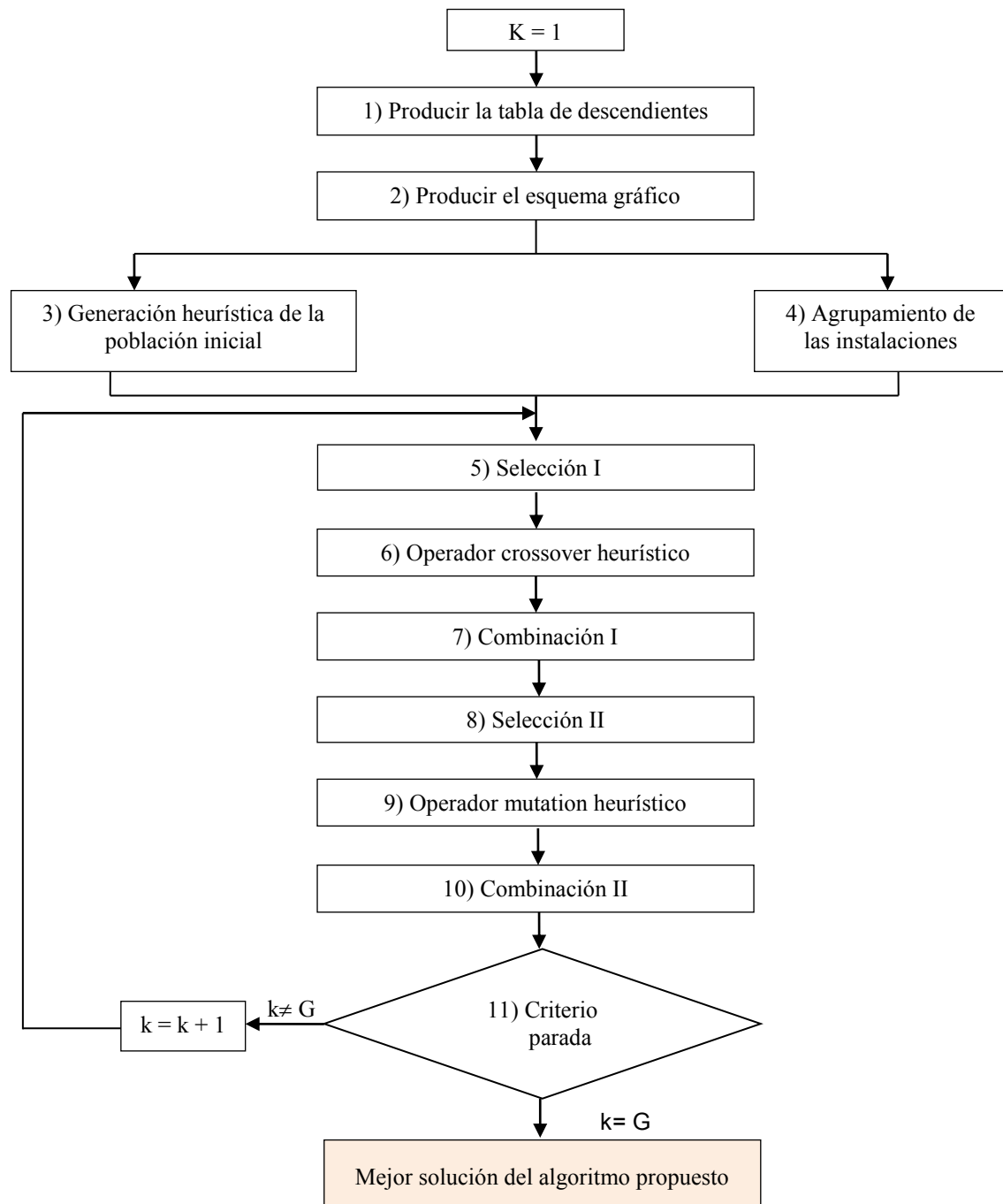
La probabilidad de casamiento  $P_c$  se multiplica por el tamaño de la población P. El resultado se denota como  $S_1$  e indica los cromosomas padre.

Paso 6. Operador heurístico de casamiento.

Sobre los  $S_1$  cromosomas seleccionados desde los P cromosomas de la población inicial se aplica el operador de casamiento. Primero un grupo de instalaciones es escogido aleatoriamente para cada pareja de padres y en seguida se define un área rectangular, de tal manera que cubra el número máximo de instalaciones del grupo y se escoge como la sección de corte.

Paso 7. Combinación I.

Los P cromosomas de la población inicial son combinados con los cromosomas descendientes creados por el operador heurístico de casamiento  $S_1$ . Los cromosomas que se repiten se eliminan.



**Figura 2.19** Diagrama de flujo del procedimiento heurístico

#### Paso 8. Selección II.

La probabilidad de mutación  $P_m$  se multiplica por el tamaño de la población resultante en la Combinación I. El resultado se denota como  $S_2$  e indica los cromosomas padre.

#### Paso 9. Operador heurístico de mutación.

El operador aleatorio de mutación intercambia los genes de cada cromosoma al azar y esto no garantiza una dirección favorable hacia la solución óptima. El algoritmo propuesto usa el operador de mutación heurística con la probabilidad de mutación de  $P_m$  para aliviar la observación anterior. La única diferencia entre estos dos procedimientos es el reemplazamiento de los dos genes. En el procedimiento aleatorio, para cada padre con el cromosoma codificado en línea simple, se generan al azar dos enteros que indicarían el reemplazamiento de los genes. Sin embargo, en el procedimiento heurístico, primero un grupo de instalaciones es seleccionado al azar para constituirse como padre, y luego dos instalaciones dentro del grupo escogido son seleccionados aleatoriamente como los genes de reemplazo (ver Fig. 7, p. 1058).

#### Paso 10. Combinación II.

Los cromosomas descendientes creados en la sección anterior son combinados con los cromosomas provenientes del Paso 7. Los cromosomas que se repiten se eliminan. Finalmente, desde la población resultante se escogen  $P$  cromosomas con los costos totales menores y se constituyen como la próxima generación.

#### Paso 11. Criterio de finalización.

En el proceso de movimiento desde una generación a la siguiente, los operadores de casamiento y mutación se repiten hasta encontrar un criterio de parada. El número máximo de generaciones ( $G$ ), la convergencia de las aptitudes y el tiempo de computación pueden seleccionarse como los criterios de parada. El criterio escogido por el autor es el número máximo de generaciones  $G$ .

La eficiencia del algoritmo propuesto ha sido evaluado usando dos ejemplos ilustrativos revisados en la literatura.

Como ejemplo I, el autor usa el mismo caso propuesto por Kazerooni *et al.* (1996) obteniendo un costo total de \$11,662, que es una mejor solución frente a otros algoritmos genéticos tradicionales que obtuvieron como El-Baz (2004) \$11,862 y Mak *et al.* (1998) \$12,982, tal como se puede observar en la Tabla 4 del artículo (p. 1060).

Luego de analizar el trabajo se puede observar que el procedimiento establecido para la representación esquemática del layout (Paso 2) es adecuado sólo para pocas instalaciones; esto, debido a la explosión combinatoria que se produce al tratar problemas con muchos departamentos.



A continuación, en la Tabla 2.4 se presenta un resumen de la revisión de los autores mencionados, y las diferentes estrategias que usan las 10 propuestas comentadas en la aplicación del Algoritmo Genético para la resolución de los Problemas de Distribución de Planta (FLP).

**Tabla 2.4** Estrategias de cada propuesta en la aplicación de Algoritmos Genéticos en FLP

Modelo	Título del artículo	Representación del layout	Descripción de la estrategia
PSD 2.2.1	Allocating Facilities with CRAFT	Representación discreta	Heurística de mejora por intercambio de pares de departamentos, desde una distribución inicial (Figura 2.4)
QAP 2.2.2	A greedy genetic algorithm for the quadratic assignment problem	La permutación de $n$ números en el set $N=\{1, 2, \dots, n\}$	Usa una heurística de construcción aleatoria para generar la población inicial Alternamiento entre diferentes poblaciones iniciales
QAP 2.2.3	Solving the Quadratic Assignment Problem by a Genetic Algorithm with New Replacement Strategy	La permutación de $n$ números en el set $N=\{1, 2, \dots, n\}$	Primero reemplaza el peor cada $T_c$ generaciones y luego sustituye a los similares a sus precursores una vez en cada generación
QAP 2.2.4	A new genetic representation for quadratic assignment problem	La permutación de $n$ números en el set $N=\{1, 2, \dots, n\}$	La permutación óptima debe corresponder a pares de instalaciones con flujos altos, ubicados en pares de localizaciones con distancias menores
UA-FLP 2.2.5	An artificial immune system based algorithm to solve unequal area facility layout problem	Estructura de bahías flexibles (FBS)	El CSA tiene una nueva codificación y un procedimiento novedoso para cubrir con departamentos ficticios, los espacios vacíos producidos en la distribución en planta
UA-FLP 2.2.6	A non dominated ranking Multi Objective Genetic Algorithm and electre method for unequal area facility layout problems	Estructura de árbol rebanado (STS)	Los objetivos cuantitativos y cualitativos se consideran con referencia al esquema de representación STS Mecanismo de diversificación y procedimiento de ranking (Figura 2.11)
SRFLP 2.2.7	Scatter search algorithms for the single row facility layout problem	La permutación de $n$ números en el set $N=\{1, 2, \dots, n\}$	Búsqueda dispersa para la nueva generación (Figura 2.12 y Figura 2.13)
nRFLP 2.2.8	A genetic algorithm for facility layout problems of different manufacturing environments	Representación discreta: considerando varios patrones de flujo de materiales (Figura 2.14)	Nuevo esquema del proceso de búsqueda genética (Figura 2.15)
nRFLP 2.2.9	Solving facilities location problem in the presence of alternative processing routes using a genetic algorithm	Representación discreta: especial que contempla dos partes	Presenta el cromosoma en dos segmentos: máquinas y rutas de procesamiento (Figura 2.18a)) Operadores genéticos: mutación y casamiento (Figura 2.18 b) y Figura 2.18 c))
nRFLP 2.2.10	A genetic algorithm with the heuristic procedure to solve the multi-line layout problem	Codificación matricial en multi línea (p. 41) desde la tabla de descendientes.	Nuevo procedimiento heurístico que consta de once pasos (Figura 2.19)

## **2.3. Herramientas para el diseño de distribución de planta**

### **2.3.1 Algoritmos tipo mejoría**

Los métodos de mejoría parten de una solución inicial del problema, a partir de la cual realizando modificaciones sistemáticas, obtienen diferentes soluciones mejoradas.

A este tipo de algoritmos pertenece el CRAFT (Buffa *et al.*, 1964) que viene a ser el método tipo mejoría más antiguo. Comienza determinando el costo total de una distribución inicial dada, es un método heurístico que trata de mejorar una solución intercambiando un par de departamentos de áreas iguales o adyacentes en el layout. Para cada intercambio CRAFT calcula y estima la reducción en el costo y escoge el intercambio con la reducción estimada más grande. El proceso de intercambio continúa hasta que ya no exista reducción estimada, haciendo los intercambios correspondientes. Hicks & Cowen (1976) criticaron el procedimiento de conmutación porque puede llevar a las instalaciones a tener formas irregulares. De este tipo de métodos, CRAFT (Computarized Relative Allocation of Facilities Technique) es quizá el más conocido, muchos otros métodos son variaciones realizadas sobre la base de éste. Puesto que el método CRAFT no garantiza una solución óptima, es necesario ejecutar varias veces con distribuciones iniciales diferentes.

WinQSB (Chang, 2000), este programa resuelve el problema de distribución funcional usando la misma metodología de CRAFT, la cual es una heurística para mejorar la distribución inicial por intercambio de departamentos. Incorpora nuevos métodos de intercambio de localización de departamentos: intercambio de dos vías (cambiar dos departamentos a la vez), intercambio de tres vías (cambiar tres departamentos a la vez), dos vías luego de intercambio de tres vías y tres vías luego de intercambio de dos vías.

La entrada incluye el flujo interdepartamental, contribución unitaria y una distribución inicial con dimensiones reducidas. La contribución unitaria representa el costo o beneficio de mover una unidad de flujo por unidad de distancia de un departamento a otro.

Las medidas de distancia que incluye son: distancia rectilínea, distancia euclidiana cuadrada y distancia euclidiana.

Pertenecen a este grupo la mayor parte de las técnicas basadas en búsqueda tabú, recocido simulado y algoritmos genéticos.

### 2.3.2 Algoritmos tipo constructor

De manera general, los métodos constructivos generan los layouts escogiendo una tras otra las instalaciones a distribuir, y colocándolas en determinadas posiciones del dominio de ubicación.

A este tipo de algoritmos pertenece CORELAP (COMputerized RELationship LAYout Planning), que fue desarrollado por Lee & Moore (1967) y se fundamenta en el procedimiento del Planeamiento Sistemático de la Distribución.

Para determinar una distribución requiere de las relaciones de cercanía de cada instalación. Bajo este criterio, la instalación con la calificación de cercanía más alta es seleccionada y asignada al centro del área de la planta, para orientar la distribución de los departamentos restantes. Las instalaciones subsiguientes son adicionadas al layout dependiendo de sus relaciones con las instalaciones ya localizadas.

También ALDEP es un algoritmo constructor que ha sido desarrollado por Seehof & Evans (1967). Este método (Automated Layout DDesign Program) trabaja en la siguiente forma: primero una instalación es seleccionada aleatoriamente y es ubicada en el rincón izquierdo superior del layout. La siguiente instalación escogida para ser localizada es una que tiene la calificación de cercanía mayor o igual a una relación de cercanía especificada por el usuario, con una selección aleatoria para la primera instalación.

Este método puede manejar distribuciones en edificios de varios niveles.

De manera general, la mayor parte de los métodos basados en técnicas de corte o teoría de grafos y algunos que hacen uso de la lógica difusa, pertenecen a este grupo.

En la Tabla 2.5 se presentan las diferencias básicas entre los dos tipos de algoritmos, sólo considerando CRAFT y CORELAP por ser los más antiguos y conocidos como técnicas de distribución computarizada.

**Tabla 2.5** Resumen comparativo de los métodos: CRAFT y CORELAP

características	mejoría: CRAFT	constructor: CORELAP
entrada	matriz de De – A matriz de costos área disponible distribución inicial	tabla de relaciones de actividad matriz REL de De – A área disponible -
objetivo	basado en la distancia	basado en las relaciones
representación	rejillas discretas	Continua
método	intercambio por parejas (sólo departamentos adyacentes)	incorporación paulatina
varios niveles	Uno	Uno
número de departamentos	máximo 40	máximo 55
hay espacio remanente	NO	SÍ

## Capítulo 3 Un algoritmo greedy para el diseño de distribución de planta multi línea

### 3.1 Formulación del problema

Supongamos que un sistema de producción tiene  $N$  departamentos de áreas conocidas que deben ser distribuidos en la planta industrial de tamaño  $W \times H$ , la misma que debe dividirse en  $q$  localizaciones, donde  $N \leq q$  y sin ninguna limitación para ubicar las instalaciones. Mientras, en tales sistemas algunos de los productos son manufacturados en diferentes volúmenes de producción y rutas de procesamiento. Sea  $c_{ij}$  el costo necesario para el manejo de una unidad de material desde la instalación  $i$  hasta la instalación  $j$ ,  $f_{ij}$  la cantidad del flujo de material desde la instalación  $i$  a  $j$ , y  $d_{ij}$  es la distancia entre las localizaciones de las instalaciones  $i$  y  $j$ . El Problema de Distribución de Planta (FLP) consiste en asignar las instalaciones en el área de la planta con el fin de optimizar un criterio dado, sujeto a la restricción de que las instalaciones no se pueden superponer.

Diferentes criterios cuantitativos han sido considerados en los FLP: costo de manejo de materiales, relaciones de cercanía, la forma de los departamentos y métricas de distancia. En este trabajo, el énfasis es dado para optimizar el costo total del manejo de materiales ( $C$ ), y su formulación para la función objetivo es planteada por Buffa *et al.* (1964) y últimamente por Wong & Komarudin (2010) de la siguiente manera:

$$C = \sum_{i=1}^N \sum_{j=1, i \neq j}^N f_{ij} c_{ij} d_{ij} \quad (3-1)$$

El Problema de Distribución de Planta puede modelarse como un Problema de Asignación Cuadrática (QAP) y si es así pertenece a la clase de los problemas NP-duro; por eso, varias técnicas basadas en heurísticas y metaheurísticas se han desarrollado para conseguir soluciones cercanas al óptimo (El-Baz, 2004 & Zadrzadeh, 2012). Continuando con esta tarea el presente trabajo define otro modelo de optimización combinatoria que vamos a plantear para desarrollar el algoritmo.

Consideremos el caso de una empresa hipotética que manufactura 4 productos y que se procesan entre 6 departamentos de operación: A, B, C, D, E y F. En la Tabla 3.1 se indica la relación de los productos con sus rutas de procesamiento para cada uno y también se cuenta con los flujos mensuales y el costo unitario por producto.

Es necesario precisar la naturaleza del flujo entre los departamentos y como se transporta el material. Vamos a suponer de que todo el material se transporta en un embalaje de tamaño estándar por medio de un montacargas (cada uno de estos embalajes constituye una carga o flujo).

**Tabla 3.1** Un ejemplo de 6 instalaciones con 4 productos

Productos	Rutas de proceso	Flujos $f_{ij}$	Costo unitario $c_{ij}$
a	A-C-D-B-F	30	1
b	A-E-B-E	90	1
c	A-D-C-B-E-F	120	1
d	A-B-E-D	21	2

La ruta de procesamiento define un camino de máquinas necesarias para elaborar un producto. Observe que el flujo de material para el producto **a** en la ruta de procesamiento es de 30, es decir, los flujos de materiales en las máquinas de A-C-D-B-F son 30 unidades en la línea de producción de dicho producto. Asimismo, se observa que el flujo total de materiales entre las máquinas B y E es de 321 (180 para el producto b, 120 para el producto c y 21 para el producto d).

El costo total (flujo de materiales \* costo unitario) para cada par de relaciones de las instalaciones (eslabones) se muestra en la Tabla 3.2.

**Tabla 3.2** Evaluación del costo para las relaciones del ejemplo

Relaciones	Costo ( $w_{ij} = f_{ij} \times c_{ij}$ )	
A – B	42	21*2
A – C	30	30*1
A – D	120	120*1
A – E	90	90*1
B – C	120	120*1
B – D	30	30*1
B – E	342	90*1+90*1+120*1+21*2
B – F	30	30*1
C – D	150	30*1 + 120*1
D – E	42	21*2
E – F	120	120*1

Consideramos el tipo de distancia Manhattan, y la distancia entre dos lugares adyacentes será la unidad. Dada la Figura 3.1 como una solución del ejemplo, entonces el costo total ( $C_1$ ) se calcula como sigue:

A	E	B
C	D	F

**Figura 3.1** Una solución del ejemplo

$$C_1 = 1668 = 42*2 + 30*1 + 120*2 + 90*1 + 120*3 + 30*2 + 342*1 + 30*1 + 150*1 + 42*1 + 120*2 \quad (3-2)$$

Dada la Figura 3.2 como otra solución del ejemplo, entonces el costo total ( $C_2$ ) será 1,770.

E	C	F
B	A	D

**Figura 3.2** Otra solución para el ejemplo

Observe que para los mismos datos de la Tabla 3.2, el costo total depende de la ubicación de las máquinas en el área de distribución, permaneciendo constante el peso o costo de las relaciones interdepartamentales. Por lo tanto, la función objetivo dada en (3-1) se puede formular de la siguiente manera:

$$C = \sum_{i=1}^N \sum_{j=1, i \neq j}^N w_{ij} \bar{d}(i, j) \quad (3-3)$$

donde  $w_{ij}$  es el costo total de la relación entre las máquinas  $i$  y  $j$ , es decir  $w_{ij} = f_{ij} * c_{ij}$ ;

$\bar{d}(i, j)$  es la distancia rectilínea entre las localizaciones de las máquinas  $i$  y  $j$ .

La matriz de distancias Manhattan para la cuadrícula de  $2 \times 3$  del ejemplo se encuentra en la Tabla 3.3 a) y las 6 localizaciones correspondientes se indica en la Tabla 3.3 b).

**Tabla 3.3** Matriz de distancias Manhattan y las localizaciones

	1	2	3	4	5	6
1	0	1	2	1	2	3
2	1	0	1	2	1	2
3	2	1	0	3	2	1
4	1	2	3	0	1	2
5	2	1	2	1	0	1
6	3	2	1	2	1	0

a) Matriz de distancias Manhattan

1		2		3	
4		5		6	

b) Localizaciones

Observe que la distancia Manhattan, entre los lugares 1 y 6 es 3, y que es el mismo que para los lugares 6 a 1, es decir, la matriz de distancia Manhattan es una matriz simétrica. Además, se puede observar que la matriz de distancia para otras métricas como euclidiana y Chebyshev tiene la propiedad de matriz simétrica.

Por lo tanto, podemos reformular la función objetivo dada en (3-3) de la siguiente manera:

$$C = \frac{1}{2} \sum_{i=1}^N \sum_{j=1, i \neq j}^N w_{ij} \bar{d}(i, j) \quad (3-4)$$

Sea  $x_i$  el lugar de ubicación de la máquina  $i$ . Observe que  $x_i \in \{1, 2, \dots, N\}$ . Entonces, el modelo de optimización para los FLP es dado como sigue (AlgoDist):

$$\text{Minimizar } f(x) = \frac{1}{2} \sum_{i=1}^N \sum_{j=i+1}^N w_{ij} \bar{d}(x_i, x_j) \quad (3-5)$$

Sujeto a:

$$x_i \in \{1, 2, \dots, N\} \quad \forall i = 1, \dots, N \quad (3-6)$$

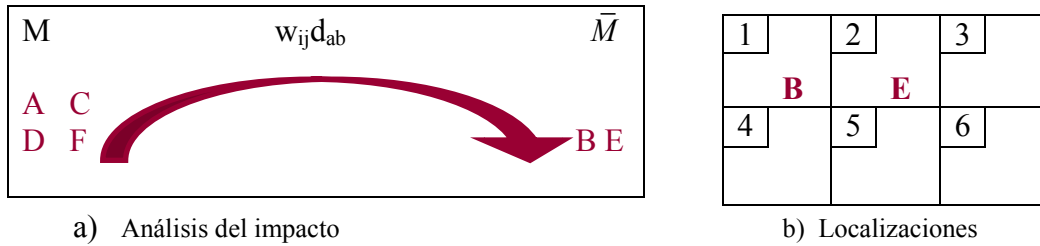
$$x_i \neq x_j \quad \forall i \neq j \in \{1, 2, \dots, N\} \quad (3-7)$$

La relación (3-5) formula la optimización del costo total de la manipulación de materiales. La restricción (3-6) asegura que cada máquina se asigna a cada ubicación. La restricción (3-7) garantiza que cada máquina se encuentra en diferentes lugares. Las restricciones (3-6) y (3-7) aseguran que cada máquina se asigna a un solo lugar y éste es único.

## 3.2 Algoritmo greedy

### 3.2.1 Criterio greedy

Sea  $\bar{M}$  y  $M$  el conjunto de máquinas asignadas y no asignadas, es decir, respectivamente, el conjunto de máquinas que pertenecen y que no pertenecen al conjunto de soluciones. Sea  $L$  y  $\bar{L}$  el conjunto de localizaciones libres y no libres, respectivamente, en el grid de localizaciones. Este planteamiento se esquematiza en la Figura 3.3.



**Figura 3.3** Proceso de distribución de las máquinas en el grid de localizaciones

Ahora vamos a analizar el impacto de localizar una máquina de  $M$  no asignada en una posición libre de  $L$  (3, 4, 5 o 6). Escogemos una máquina  $i$  ( $i \in M$ ) y  $p$  como la posición libre para la localización ( $p \in L$ ). Entonces, el impacto de una máquina  $i \in M$  localizada en  $p \in L$  sobre todas las máquinas en  $\bar{M}$  respecto al costo total de manejo de materiales es dado por la función  $g(i, p)$  y se define como sigue:

$$g(i, p) = \sum_{j \in \bar{M}} w_{ij} \bar{d}(p, x_j) \quad (3-8)$$

A continuación vamos a calcular la mejor máquina y la mejor posición para pasarlo. Es fácil ver que la mejor posición libre  $p_i$  para la máquina  $i \in M$  puede encontrarse de la manera siguiente:

$$p_i := \text{Arg Min}_{p \in L} \{g(i, p)\} \quad (3-9)$$

Es decir,  $p_i \in L$  es un lugar libre que presenta un costo total mínimo para  $i \in M$  respecto a las máquinas asignadas, y

$$g(i, p_i) = \text{Min}_{p \in L} \{g(i, p)\} \quad (3-10)$$

Resulta fácil ver que la búsqueda de  $p_i$  puede reducirse a los lugares libres y frontera de  $\bar{L}$ .

Por otra parte, la mejor máquina del conjunto  $M$  respecto al coste total, en relación con las máquinas en  $\bar{M}$  es tal que presenta el flujo máximo entre éste y todas las máquinas en  $\bar{M}$ , esto se debe a que tenemos que cubrir todos los flujos; si se deja la máquina con el flujo máximo para la última localización, entonces es posible aumentar considerablemente el valor de la función objetivo, porque la distancia entre esta máquina y máquinas asignadas es más probable que aumente considerablemente. En este sentido, la mejor máquina es evaluada como sigue:

$$b := \text{Arg Max}_{i \in L} \{g(i, p_i)\} \quad (3-11)$$

Es decir  $b \in M$  es la mejor máquina a localizar y,

$$g(b, p_b) = \text{Max}_{i \in M} \{g(i, p_i)\} \quad (3-12)$$

La relación (3-11) define el criterio goloso para evaluar la mejor máquina para asignar, y la relación (3-9) evalúa el mejor lugar para cada máquina.

### 3.2.2 La primera pareja de máquinas y su ubicación en el layout

#### Selección de la primera pareja

Cuando  $\bar{M}$  está vacío, es decir no hay máquinas asignadas, entonces las relaciones (3-9) y (3-11) no se pueden aplicar debido a que el valor de la función de  $g$  es cero para cualquier primera máquina para asignar. Un criterio para evaluar la primera pareja a asignar es el par de instalaciones con mayor flujo interdepartamental.

Por lo tanto, la primera pareja de máquinas a asignar es:

$$w_{ab} = \text{Arg Max}_{(i,j) \in A} \{w_{ij}\} \quad (3-13)$$

Observe que este criterio es concordante con el criterio goloso para asignar la máquina (3-11).



## El lugar para la primera pareja de máquinas

La primera pareja de máquinas se puede asignar en cualquier ubicación, pero su lugar puede influir en el valor final de la función objetivo. Una alternativa es probar todos los lugares para la primera pareja a asignarse. Esta alternativa nos permite obtener varias soluciones y vamos a utilizarla en el algoritmo propuesto.

### 3.2.3 Métrica de distancia Manhattan

En el criterio para medir la distancia entre departamentos en el problema de distribución en planta, primero se debe definir el punto final de la medida de la distancia, aquí se asume que las distancias son medidas entre los centroides o centros de gravedad de los departamentos; en seguida se debe especificar la ruta del viaje. Una posibilidad es que siga el camino de la línea recta a la que se denomina medida euclidiana. Una alternativa más realista es asumir que el flujo viaja por caminos paralelos a los ejes del área de la planta, esta es la medida rectilínea conocida también como distancia Manhattan.

Los centroides son especificados en términos de las coordenadas del sistema cartesiano como:

$x(i)$ = coordenada - x del centroide del departamento i

$y(i)$ = coordenada - y del centroide del departamento i

En la Figura 3.1 los departamentos del ejemplo están ordenados en una cuadrícula de 2 x 3, donde cada cuadro representa una superficie de 10 x 10 metros cuadrados. Tenemos los centroides como sigue:

Departamento A:  $x(A)= 5$ ,  $y(A)= 15$

Departamento C:  $x(C)= 5$ ,  $y(C)= 5$

Departamento D:  $x(D)= 15$ ,  $y(D)= 5$

etc.

En la tesis establecemos la métrica de medida rectilínea y la distancia entre dos departamentos queda definida como:

$$d_{ij} = |(x(i) - x(j))| + |y(i) - y(j)| \quad (3-14)$$

En la cuadrícula determinamos  $n$  puntos fijos como centroides en el área de distribución en planta y asignamos los departamentos a los puntos. Es decir la distancia entre los centros de gravedad no variará, pero la distancia entre departamentos dependerá de la asignación de los mismos a los centroides.

### 3.2.4 Algoritmo AlgoDist

Nomenclatura:

Índices

$i, j$  ítems de máquinas o instalaciones

Parámetros

$N$	número de departamentos (instalaciones o máquinas)
$n1$	número de filas de la rejilla de distribución
$n2$	número de columnas de la rejilla de distribución
$p$	posición libre en el área de localizaciones
$q$	número de localizaciones ( $n1 \times n2$ )

Sets

$M$	conjunto de máquinas no distribuidas
$\bar{M}$	conjunto de máquinas distribuidas
$L$	conjunto de localizaciones libres
$\bar{L}$	conjunto de localizaciones ocupadas
$A$	conjunto de relaciones interdepartamentales

Variables continuas

$W = (w_{ij})$  conjunto de pesos asociado a las relaciones interdepartamentales  $(i, j)$  en  $A$ .

$D = (d_{ab})$  distancia entre las localizaciones  $a$  y  $b$  en el área (rejilla) de distribución.

Ahora vamos a utilizar los conceptos desarrollados en las secciones 3.2.1, 3.2.2 y 3.2.3 para construir un algoritmo para resolver los Problemas de Distribución de Planta en filas múltiples.

En primer lugar, determinamos la primera pareja de máquina y asignamos a éste en un lugar diferente en cada iteración ( $r$ ), de manera que podamos obtener diferentes soluciones para los FLP, es decir,  $n$  soluciones al final de todas las iteraciones.

En segundo lugar, en cada iteración el algoritmo genera una solución usando el criterio codicioso de mejor lugar para cada máquina (3-9) y el criterio goloso de mejor máquina para asignar (3-11). El algoritmo propuesto se denota **AlgoDist** cuyo pseudocódigo se presenta en la Figura 3.4.

Algoritmo AlgoDist

1.  $n = N-1$   $f^* = +\infty$   $x^* = (0, \dots, 0)$
2. Evaluar  $w_{ab} = \text{Arg Max}_{(i,j) \in A} \{w_{ij}\}$
3. Para  $r=1, \dots, n$
4.      $x_a = r$
5.      $x_b = r+1$
6.      $\bar{M} = \{a, b\}$      $M = \{1, \dots, N\} - \{a, b\}$      $L = \{1, \dots, q\} - \{r, r+1\}$
7.     Mientras  $M \neq \emptyset$
8.         Evaluar  $g(i, p) = \sum_{j \in \bar{M}} w_{ij} \bar{d}(p, x_j)$
9.         Evaluar  $g(i, p_i) = \text{Min}_{p \in L} \{g(i, p)\}$
10.        Evaluar  $g(b, p_b) = \text{Max}_{i \in M} \{g(i, p_i)\}$
11.         $x_b = p_b$   $M \leftarrow M - \{b\}$   $\bar{M} = \bar{M} \cup \{b\}$   $L \leftarrow L - \{p_b\}$
12.     Fin\_Mientras
13.     if  $f(x) < f^*$  then  $x^* = x$ ,  $f^* = f(x)$
14.     end-if
15.   Fin\_Para
16.   Escribir  $(f^*, x^*)$

**Figura 3.4** Pseudocódigo del algoritmo propuesto **AlgoDist**

En el paso 1 inicializamos la mejor solución y el valor de la función de costo total.

La primera pareja de máquinas para asignar se determina en el paso 2.

Observe que el proceso para calcular  $n$  soluciones es determinado en los pasos 3-15.

También observamos que el proceso de construir una solución se da en los pasos 7-12.

La grabación de la mejor solución y su correspondiente valor de la función objetivo se actualizan en los pasos 13-14.

Observe que en cada iteración el algoritmo generará una solución, en  $n$  ( $N-1$ ) iteraciones habrá encontrado  $n$  soluciones, de las cuales la mejor solución es registrada en  $x^*$  y su valor en  $f^*$ .

Finalmente, en el paso 16 se escribe la mejor solución encontrada después de  $n$  iteraciones.

### 3.3 Ilustración del algoritmo

Consideremos los datos del ejemplo de seis instalaciones y cuatro productos mostrados en la Tabla 3.1. A continuación se construye la tabla de las cargas (costo) interdepartamentales  $w_{ij}$  la misma que se presenta en la Tabla 3.4.

**Tabla 3.4** Datos de entrada del ejemplo

Relaciones	Costo ( $w_{ij} = f_{ij} \times c_{ij}$ )
A – B	42
A – C	30
A – D	120
A – E	90
B – C	120
B – D	30
B – E	342
B – F	30
C – D	150
D – E	42
E – F	120

Sea el conjunto de aristas A que resulta de las relaciones interdepartamentales de la Tabla 3.4, como datos de cálculo:

(i,j)	AB	AC	AD	AE	BC	BD	BE	BF	CD	DE	EF
$w_{ij}$	42	30	120	90	120	30	<b>342</b>	30	150	42	120

$w_{ab} = w_{BE} = 342$  máximo valor

Definir una rejilla de  $n_1 \times n_2$ :  $2 \times 3 = 6$  localizaciones.

1	2	3
4	5	6

**r= 1**

Considerando  $x_B = 1$  y  $x_E = 2$ , asignar las máquinas BE en las localizaciones 1 y 2 respectivamente.

<b>B</b>	<b>E</b>	3
4	5	6

M	$\bar{M}$	L	g			
A C D F	B E	3 4 5 6	A	$g(A,3) = AB$	$42 \times 2$	AE $90 \times 1$ 174
				$g(A,4) = AB$	$42 \times 1$	AE $90 \times 2$ 222
			C	$g(C,3) = BC$	$120 \times 2$	240
				$g(C,4) = BC$	$120 \times 1$	120
			D	$g(D,4) = BD$	$30 \times 1$	DE $42 \times 2$ 114
				$g(D,5) = BD$	$30 \times 2$	DE $42 \times 1$ 102
			F	$g(F,4) = BF$	$30 \times 1$	EF $120 \times 2$ 270
				$g(F,5) = BF$	$30 \times 2$	EF $120 \times 1$ 180

**F** ⇒

<b>B</b>	<b>E</b>	3
4	<b>F</b>	6

$g(F,5) = \max \{g(A,3), g(C,4), g(D,5), g(F,5)\}$

174      120      102      180

Colocamos la máquina F en la posición 5:

M	$\bar{M}$	L	g			
A C D	B E F	3 4 6	A	g(A, 3)= AB 42 x 2	AE 90 x 1	174
				g(A, 4)= AB 42 x 1	AE 90 x 2	222
			C	g(C, 3)= BC 120 x 2	240	
				g(C, 4)= BC 120 x 1	120	
			D	g(D, 3)= BD 30 x 2	DE 42 x 1	102
				g(D, 4)= BD 30 x 1	DE 42 x 2	114

A ⇒

	B	E	A
4		F	6

$$g(A,3) = \max \begin{matrix} \{g(A,3), g(C,4), g(D,3)\} \\ 174 \quad 120 \quad 102 \end{matrix}$$

Colocamos la máquina A en la posición 3

M	$\bar{M}$	L	g			
C D	B E F A	4 6	C	g(C,4)= AC 30 x 3	BC 120 x 1	210
				g(C,6)= AC 30 x 1	BC 120 x 3	390
			D	g(D,4)= AD 120 x 3	BD 30 x 1	DE 42 x 2 474
				g(D,6)= AD 120 x 1	BD 30 x 3	DE 42 x 2 294

C ⇒

	B	E	A
4		F	D

$$g(D,6) = \max \begin{matrix} \{g(C,4), g(D,6)\} \\ 210 \quad 294 \end{matrix}$$

Colocamos la máquina D en la posición 6 y la máquina C en la posición 4:

Evaluación de la función objetivo ( $f^*$ ):

$$=84(AB)+90(AC)+120(AD)+90(AE)+120(BC)+90(BD)+342(BE)+60(BF)+300(CD)+84(DE)+120(EF) \\ =1,500$$

**r= 2**

Localización de las máquinas BE en las ubicaciones 2 y 3, respectivamente.

1		B	E
4	5		6

M	$\bar{M}$	L	g			
A C D F	B E	1 4 5 6	A	g(A,1)= AB 42 x 1	AE 90 x 2	222
				g(A,6)= AB 42 x 2	AE 90 x 1	174
			C	g(C,1)= BC 120 x 1	120	
				g(C,6)= BC 120 x 2	240	
			D	g(D,1)= BD 30 x 1	DE 42 x 2	111
				g(D,6)= BD 30 x 2	DE 42 x 1	81
			F	g(F,1)= BF 30 x 1	EF 120 x 2	270
				g(F,6)= BF 30 x 2	EF 120 x 1	180

F ⇒

1		B	E
4	5		F

$$g(F,6) = \max \begin{matrix} \{g(A,6), g(C,1), g(D,6), g(F,6)\} \\ 174 \quad 120 \quad 81 \quad 180 \end{matrix}$$

Colocamos la máquina F en la posición 6:

M	$\bar{M}$	L	g
A C D	B E F	1 4 5	A g(A, 1) = AB 42 x 1    AE 90 x 2    222 g(A, 4) = AB 42 x 2    AE 90 x 3    354 C g(C, 1) = BC 120 x 1                    120 D g(D, 1) = BD 30 x 1           DE 42 x 2    114 g(D, 5) = BD 30 x 1           DE 24 x 2    114

$A \Rightarrow$

A	B	E
4	5	F

$$g(A, 1) = \max \begin{matrix} \{g(A,1), g(C,1), g(D,1)\} \\ 222 \quad 120 \quad 114 \end{matrix}$$

Colocamos A en la posición 1:

M	$\bar{M}$	L	g
C D	B E F A	4 5	C g(C,4)= AC 30 x 1    BC 120 x 2 = 270 g(C,5)= AC 30 x 2    BC 120 x 1 = 180 D g(D,4)= AD 120 x 1    BD 30 x 2    DE 42 x 3 = 306 g(D,5)= AD 120 x 2    BD 30 x 1    DE 42 x 2 = 354

$C \Rightarrow$

A	B	E
D	5	F

$$g(D,4) = \max \begin{matrix} \{g(C,5), g(D,4)\} \\ 180 \quad 306 \end{matrix}$$

Colocamos la máquina D en la posición 4 y la máquina C en la posición 5:

Evaluación de la función objetivo ( $f^*$ ):

$$= 42(AB) + 60(AC) + 120(AD) + 180(AE) + 120(BC) + 60(BD) + 342(BE) + 60(BF) + 150(CD) + 126(DE) + 120(EF) = 1,380$$

Se presenta el resumen en la Tabla 3.5 de las 5 iteraciones desde  $r=1$  hasta  $r=5$ .

**Tabla 3.5** Resumen de los cálculos de la ilustración del algoritmo

r	$x^*$	$f^*$	layout
1	3 1 4 6 2 5	1,500	B E A C F D
2	1 2 5 4 3 6	1,380	A B E D C F
3	1 3 6 2 4 5	2,124	A D B E F C
4	6 4 1 3 5 2	1,500	C F D B E A
5	2 5 4 1 6 3	1,380	D A F C B E

### 3.4 Análisis de complejidad

De la misma manera que nos preocupamos por la corrección de los algoritmos, también nos preocupa su eficiencia (Martí *et al.*, 2013), es decir, su bondad a la hora de consumir recursos tales como el tiempo de procesamiento, la memoria auxiliar necesaria al ejecutarlos o el número de evaluaciones de la función objetivo.

Análisis del algoritmo **AlgoDist**:

Para determinar la complejidad del algoritmo, podemos agrupar en 4 grandes bloques: etapas 1, 2, 3-15 y 16. Sean  $T_1$ ,  $T_2$ ,  $T_3$  y  $T_4$  la complejidad de los bloques, respectivamente; entonces, la complejidad del algoritmo se define como;

$$T_1 + T_2 + T_3 + T_4 \quad (3-15)$$

Aplicando la primera regla sobre complejidad, que dice: si un algoritmo está constituido por bloques independientes, entonces la complejidad del algoritmo estará dada por la mayor complejidad asintótica de los bloques.

En este caso la complejidad de AlgoDist estará definida por  $T_3$  que corresponde a las etapas del 3 al 15.

Ahora vamos a calcular la complejidad de  $T_3$ . Observe que  $T_3$  es  $n(N-1)$  veces la complejidad de las etapas 7-12. Denotemos como  $T_4$  la complejidad de los pasos 7-12; entonces,  $T_3$  se da de la siguiente manera:

$$T_3 = O(nT_4) \quad (3-16)$$

Observe también que  $T_4$  es  $n$  veces la complejidad de los pasos 8-11. Por otro lado, los pasos 8-11 están dominados por los pasos 8 y 9. Es fácil ver que la complejidad de los pasos, 8 y 9 es

$$O(|M||L|).$$

Por otra parte, observamos que parte del esfuerzo de los pasos 10 y 11 se ejecutan en los pasos 8 y 9, respectivamente; por esta razón, la complejidad de los pasos 8, 9 y 10 son  $O(|M||L|)$  y  $O(|M|)$ , respectivamente. Por lo tanto,  $T_3$  se da como sigue:

$$T_3 = O(n|M||L|) \quad (3-17)$$

Desde las etapas 6 y 11 observamos que  $|\bar{M}| + |M| = N$  y  $|L| = |M|$ . Entonces, la relación (3-17) se reduce a:

$$T_3 = O(n^3) \quad (3-18)$$

Finalmente, las relaciones (3-16), (3-17) y (3-18) definen la complejidad del algoritmo:

$$O(\text{AlgoDist}) = O(n^3)$$

## Capítulo 4 Experimentos numéricos

### 4.1 Hardware y software empleado

El algoritmo AlgoDist ha sido implementado en el lenguaje de programación Code::Blocks 12.11 y los experimentos se han desarrollado en una computadora personal Toshiba con procesador Intel(R) Core(TM) i7 CPU@2.30 Ghz, 8GB de RAM con el sistema operativo Windows 7. Una evaluación comparativa del algoritmo propuesto ha sido proporcionada por el uso de siete ejemplos numéricos. Para todos los casos de prueba se considera la métrica de distancia Manhattan, con la distancia de la unidad entre dos instalaciones adyacentes.

### 4.2 Casos de prueba

#### 4.2.1 Caso 1: Curso de Distribución de Planta FII-UNMSM (Salas 2014-2)

##### 4.2.1.1 Definición

Una empresa X debe establecer una nueva disposición de sus seis departamentos de producción: A, B, C, D, E y F. Se fabrican 6 diferentes artículos con la secuencia y volumen que se indica en la Tabla 4.1. Los productos para su manipulación se embalan así: a, b, c y e en cajas/50 pz y d, f en cajas/100 pz.

**Tabla 4.1** Lista de los productos, rutas de proceso y volúmenes de producción del Caso 1

Producto	Rutas	Pz/año	Pz/caja	Cajas/año
a	A-C-D-B-F	3,000	50	60
b	A-E-B-E-F	9,000	50	180
c	A-D-C-B-E-F	12,000	50	240
d	A-B-E-F	4,200	100	42
e	A-D-B-C-F	6,000	50	120
f	A-B-E-B-F	3,600	100	36

Los costos del manejo de materiales por producto se asume que son constantes.

##### 4.2.1.2 Datos de entrada

Para simplicidad del programa las denominaciones de los departamentos de producción se codifican así: A=1, B=2, C=3, D=4, E=5 y F=6. A partir de la información de la Tabla 4.1 y asumiendo que los costos unitarios por ruta de procesamiento son constantes, se construye la Tabla 4.2 que consiste en la matriz de datos de entrada para el programa.



**Tabla 4.2** Matriz de entrada del Caso 1

a	b	c	d	e	f
1,3,60	1,5,180	1,4,240	1,2,42	1,4,120	1,2,36
3,4,60	5,2,180	4,3,240	2,5,42	4,2,120	2,5,36
4,2,60	2,5,180	3,2,240	5,6,42	2,3,120	5,2,36
2,6,60	5,6,180	2,5,240		3,6,120	2,6,36
		5,6,240			

La distribución de planta se configura en una cuadrícula de  $2 \times 3 = 6$  localizaciones.

En la primera fila de la matriz aparecen los productos, que son 6, las columnas comprenden las rutas de procesamiento de cada uno con sus volúmenes de producción, respectivamente. El programa toma los datos de los productos como un vector columna, así de sencillo.

#### 4.2.1.3 Resultados

A continuación presentamos en la Tabla 4.3 a) los resultados de las 5 iteraciones del algoritmo.

En la Tabla 4.3 b) se presentan las evaluaciones de la función objetivo para  $r = 1, 2, 3, 4$  y 5.

**Tabla 4.3** Iteraciones del algoritmo, función objetivo y la mejor distribución del Caso 1

```

E:\AlgoDist\AlgoDist\bin\Debug\AlgoDist.exe
0 70 60 360 180 0
70 0 360 180 714 96
360 360 0 360 0 120
360 180 360 0 0 0
180 714 0 0 0 462
0 96 120 0 462 0

0 1 2 1 2 3
1 0 1 2 1 2
2 1 0 3 2 1
1 2 3 0 1 2
2 1 2 1 0 1
3 2 1 2 1 0

a1:2
a2:5
maxpro:714

r:1
f:=3822
x:=
6 1 4 5 2 3

r:2
f:=3426
x:=
1 2 5 4 3 6

r:3
f:=4854
x:=
3 2 5 4 1

r:4
f:=3864
x:=
4 1 3 5 2

r:5
f:=3426
x:=
5 2 1 6 3

X:=
1 2 5 4 3 6
f:=3426
Process returned 0 (0x0) execution time : 0.030 s
Press any key to continue.

```

a) Resultados de las 5 iteraciones del algoritmo

r	x*	f*
1	6 1 4 5 2 3	3,822
2	1 2 5 4 3 6	3,426
3	6 3 2 5 4 1	4,854
4	6 4 1 3 5 2	3,864
5	A B C D E F 4 5 2 1 6 3	3,426

b) Valores de la función objetivo

1	2	3
D	C	F
4	5	6
A	B	E

c) Distribución de la mejor solución

Representación esquemática de la mejor solución para  $r=5$ :

$$S = \begin{matrix} & A & B & C & D & E & F & \leftarrow \text{instalaciones} \\ \begin{matrix} 4 & 5 & 2 & 1 & 6 & 3 \end{matrix} & & & & & & & \leftarrow \text{localizaciones} \end{matrix}$$

La distribución de planta que corresponde a esta representación se presenta en la Tabla 4.3 c), indicada por las letras en la parte inferior de cada cuadro.

El tiempo de procesamiento para la obtención de la mejor solución es 0.030 segundos.

#### 4.2.2 Caso 2: Chase *et al.* (2009), ejercicio 11, p. 249

##### 4.2.2.1 Definición

En la Tabla 4.4 se presenta el flujo de materiales por los ocho departamentos. La tabla muestra los flujos de ingreso y de salida de los distintos departamentos, pero suponga que la dirección del flujo no es importante. Además, suponga que el costo de trasladar el material sólo depende de la distancia que se traslade.

**Tabla 4.4** Matriz de flujos interdepartamentales

Departamentos								
	1	2	3	4	5	6	7	8
1	—	20						
2	15	—	25				4	
3		5	—	40	5			
4			5	—	10			
5				20	—	30		
6				—		—	20	
7				3			—	10
8							5	—

Crear un esquema de distribución en el cual los departamentos estén ordenados en una cuadrícula de  $2 \times 4$ , donde cada cuadro representa una superficie de  $10 \times 10$  metros cuadrados.

Evaluar la distribución midiendo la distancia multiplicada por el flujo. Suponga que la distancia se mide en forma rectilínea (en este caso, los departamentos que están directamente adyacentes están separados por 10 metros y los que están en diagonal están separados por 20 metros).

##### 4.2.2.2 Datos de entrada

A partir de la información de la Tabla 4.4 y teniendo en cuenta los parámetros del problema, se construye la matriz de datos de entrada para el programa, tal como se ilustra en la Tabla 4.5 a).

**Tabla 4.5** Matriz de entrada y cuadrícula de 2 x 4

1,2,35	2,3,30	2,7,4	3,4,45	3, 5,5
4,5,30	5,6,30	6,7,20	7,4,3	7,8,15

a) Datos de entrada

1	2	3	4
5	6	7	8

b) Cuadrícula de 8 localizaciones

Definir una cuadrícula de  $2 \times 4 = 8$  localizaciones, tal como se muestra en la Tabla 4.5 b) indicada por los números en las cajas de la parte superior de cada cuadro.

#### 4.2.2.3 Resultados

En la Tabla 4.6 a) se presentan los resultados de las 7 iteraciones del algoritmo.

A continuación, en la Tabla 4.6 b) se presenta la salida del programa para  $r=1, 2, 3, 4, 5, 6$  y  $7$ .

**Tabla 4.6:** Iteraciones del algoritmo, función objetivo y la mejor distribución del Caso 2

```

E:\AlgoDist\AlgoDist\bin\Debug\Algo.exe
0 1 2 3 4 5 6 7 8 9
1 0 1 2 3 4 5 6 7 8
2 1 0 1 2 3 4 5 6 7
3 2 1 0 1 2 3 4 5 6
4 3 2 1 0 1 2 3 4 5
5 4 3 2 1 0 1 2 3 4
6 5 4 3 2 1 0 1 2 3
7 6 5 4 3 2 1 0 1 2
8 7 6 5 4 3 2 1 0 1
9 8 7 6 5 4 3 2 1 0

a1:3
a2:4
maxpeso:45

r:1
f:=236
x:=
0 5 1 2 3 4 8 7
-----
r:2
f:=230
x:=
0 1 2 3 4 8 7 6
-----
r:3
f:=228
x:=
1 2 3 4 8 7 6 5
-----
r:4
f:=395
x:=
2 3 4 5 1 6 7 8
-----
r:5
f:=302
x:=
3 1 5 6 2 4 8 7
-----
r:6
f:=260
x:=
4 2 6 7 3 4 8 5
-----
r:7
f:=294
x:=
1 3 7 8 4 2 6 5
-----
r:8
f:=228
x:=
2 3 4 8 7 6 5
-----
Process returned 0 (0x0) execution time : 0.040 s
Press any key to continue.

```

a) Resultados de las 7 iteraciones del algoritmo

<b>r</b>	<b>x*</b>	<b>f*</b>
1	6 5 1 2 3 4 8 7	236
2	5 1 2 3 4 8 7 6	230
<b>3</b>	<b>1 2 3 4 8 7 6 5</b>	<b>228</b>
4	2 3 4 5 1 6 7 8	395
5	3 1 5 6 2 4 8 7	302
6	1 2 6 7 3 4 8 5	260
7	1 3 7 8 4 2 6 5	294

b) Valores de la función objetivo

1	2	3	4
1	2	3	4
5	6	7	8
8	7	6	5

### c) Distribución de la mejor solución

Representación esquemática de la mejor solución para  $r=3$ :

$$S = \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \leftarrow & \text{instalaciones} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 8 \\ 7 \\ 6 \\ 5 \end{matrix} & \begin{bmatrix} 1 & 2 & 3 & 4 & 8 & 7 & 6 & 5 \end{bmatrix} & & & & & & & & \leftarrow & \text{localizaciones} \end{matrix}$$

La distribución de planta que corresponde a esta representación se presenta en la Tabla 4.6 c) indicada por los números en la parte inferior de cada cuadro.

El tiempo de procesamiento para la obtención de la mejor solución es 0.040 segundos.

#### 4.2.3 Caso 3: El-Baz (2004), Mak *et al.* (1998) y Chan & Tansri (1994)

##### 4.2.3.1 Definición

Se ha tomado el ejemplo I del artículo correspondiente al primer autor. El flujo de materiales entre las nueve instalaciones y el costo respectivo del manejo de materiales están indicados en la Tabla 4.7 a) y 4.7 b), respectivamente.

**Tabla 4.7** Datos generales del Caso 3

	1	2	3	4	5	6	7	8	9
1	0	100	3	0	6	35	190	14	12
2		0	6	8	109	78	1	1	104
3			0	0	0	17	100	1	31
4				0	100	1	247	178	1
5					0	1	10	1	79
6						0	0	1	0
7							0	0	0
8								0	12

a) Flujo de materiales

	1	2	3	4	5	6	7	8	9
1	0	1	2	3	3	4	2	6	7
2		0	12	4	7	5	8	6	5
3			0	5	9	1	1	1	1
4				0	1	1	1	4	6
5					0	1	1	1	1
6						0	1	4	6
7							0	7	1
8								0	1

b) Costo del manejo de materiales

##### 4.2.3.2 Datos de entrada

La matriz de entrada se genera por la multiplicación del flujo por el costo del manejo de materiales entre las máquinas y se muestra en la Tabla 4.8.

**Tabla 4.8** Datos de entrada del Caso 3

1,2,100	1,3,6	1,5,18	1,6,140	1,7,380	1,8,84	1,9,84	2,3,72
2,4,32	2,5,763	2,6,390	2,7,8	2,8,6	2,9,520	3,6,17	3,7,100
3,8,1	3,9,31	4,5,100	4,6,1	4,7,247	4,8,712	4,9,6	5,6,1
5,7,10	5,8,1	5,9,79	6,8,4	8,9,12			

La distribución de planta se configura en una cuadrícula de  $3 \times 3 = 9$  localizaciones.

#### 4.2.3.3 Resultados

En la Tabla 4.9 a) se presentan los resultados de las 8 iteraciones del algoritmo.

Se presentan en la Tabla 4.9 b) los cálculos de la función objetivo para  $r=1$  hasta 8.

**Tabla 4.9** Iteraciones del algoritmo, la función objetivo y la mejor distribución del Caso 3

```

E:\AlgoDist\AlgoDist\bin\Debug\AlgoDist.exe
a1:2
a2:2
a3:5
maxpeso:763
r=1
f*:5331
x*:
1 7 9 2 3 6 8 4
r=2
f*:4992
x*:
2 6 8 3 5 7 9 1
r=3
f*:6720
x*:
3 9 7 4 6 8 1 2
r=4
f*:4862
x*:
4 2 6 5 7 9 3 1
r=5
f*:5447
x*:
5 7 9 6 4 3 8 2
r=6
f*:6552
x*:
6 3 1 7 9 4 2 5
r=7
f*:5567
x*:
7 9 3 8 5 2 6 4
r=8
f*:4938
x*:
8 6 2 9 7 1 3 5
f*:4862
Process returned 0 (0x0)   execution time : 0.040 s
Press any key to continue.

```

a) Resultados de las 8 iteraciones

r	x*	f*
1	5 1 7 9 2 3 6 8 4	5,331
2	4 2 6 8 3 5 7 9 1	4992
3	5 3 9 7 4 6 8 1 2	6720
4	<b>8 4 2 6 5 7 9 3 1</b>	<b>4862</b>
5	1 5 7 9 6 4 3 8 2	5447
6	8 6 3 1 7 9 4 2 5	6552
7	1 7 9 3 8 5 2 6 4	5567
8	4 8 6 2 9 7 1 3 5	4938

b) Valores de la función objetivo

1	2	3
9	3	8
4	5	6
2	5	4
7	8	9
6	1	7

c) Distribución de la mejor solución

Representación esquemática de la mejor solución para  $r=4$ :

$$S = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & \leftarrow \text{instalaciones} \\ [8 & 4 & 2 & 6 & 5 & 7 & 9 & 3 & 1 & \leftarrow \text{localizaciones} \end{matrix}$$

La distribución de planta que corresponde a esta representación se presenta en la Tabla 4.9 c) indicada por los números en la parte inferior de cada cuadro.

El tiempo de procesamiento para la obtención de la mejor solución es 0.040 segundos.

## 4.2.4 Caso 4: Buffa *et al.* (1964), oficina con 10 departamentos

### 4.2.4.1 Definición

Los autores consideran el problema de arreglar varios departamentos en una planta en un solo piso y dimensiones fijas. Para ilustrar consideran una oficina con 10 departamentos, cada instalación tiene un área de 100 pies<sup>2</sup>. Las diez instalaciones son colocadas en un área de 50 pies de longitud (x) por 20 pies de ancho (y), tal como se puede observar en la Figura 2.4.

### 4.2.4.2 Datos de entrada

A partir de la Tabla 2.1 (Sección 2.2.1) se elaboran los datos de entrada, tal como se muestra en la Tabla 4.10 a). La distribución se configura en una cuadrícula de 2 x 5= 10 localizaciones (Tabla 4.10 b)).

**Tabla 4.10** Matriz de entrada y la cuadrícula de distribución del Caso 4

1,5,5	1,9,1	1,10,9	2,4,3	2,5,4	2,7,1	3,4,1	3,7,9	3,10,2
4,7,6	4,9,1	5,6,8	5,7,3	6,7,2	6,8,5	7,8,6	8,9,8	

a) Datos de entrada del Caso 4

1		2		3		4		5	
6		7		8		9		10	

b) Cuadrícula de 10 localizaciones del Caso 4

### 4.2.4.3 Resultados

En la Tabla 4.11 a) se presentan los resultados de las 9 iteraciones del algoritmo.

**Tabla 4.11** Iteraciones del algoritmo y la función objetivo del Caso 4

The screenshots show the following data for iterations 11 through 19:

Iteration	Layout (10x10 grid)
11	8 5 9 6 7 3 4 10 2
12	7 10 4 1 6 8 9 5 3
13	5 8 10 2 1 7 6 9 4
14	1 9 10 3 2 8 7 6 5
15	1 8 2 4 3 9 10 7 6
16	3 9 8 1 2 4 5 10 7
17	10 9 5 2 1 3 4 6 8
18	6 10 1 3 2 4 5 7 9
19	1 8 7 4 3 5 2 6 10
20	4 1 9 10 3 2 8 7 6 5

a) Resultados de las 9 iteraciones

<b>r</b>	<b>x*</b>	<b>f*</b>
1	1 8 5 9 6 7 3 4 10 2	128
2	2 7 10 4 1 6 8 9 5 3	132
3	3 5 8 10 2 1 7 6 9 4	119
<b>4</b>	<b>4 1 9 10 3 2 8 7 6 5</b>	<b>108</b>
5	5 1 8 2 4 3 9 10 7 6	167
6	6 3 9 8 1 2 4 5 10 7	108
7	7 10 9 5 2 1 3 4 6 8	144
8	8 6 10 1 3 2 4 5 7 9	147
9	9 1 8 7 4 3 5 2 6 10	150

b) Valores de la función objetiva

La Tabla 4.11 b) presenta las evaluaciones de la función objetivo para  $r=1$  hasta 9, donde se observa que la solución óptima corresponde para  $r=4$ .

El tiempo de procesamiento para la obtención de la mejor solución es 0.50 segundos.

#### 4.2.5 Caso 5: Red Crystal Glass Products Company (Gaither *et al.*, 2000)

##### 4.2.5.1 Definición

RCGP Company produce seis productos que se transportan entre 10 departamentos de operación dentro de su actual planta de producción: 1 Soplar y moldear, 2 Tratamiento térmico, 3 Formación del cuello, 4 Empacar, 5 Esmerilar, 6 Pintar, 7 Perforar, 8 Retrabajar, 9 Recubrimiento y 10 Embarque y recepción. También se muestra la relación de los seis productos, su volumen de producción mensual y su recorrido entre departamentos (ver Tabla 4.12).

**Tabla 4.12** Lista de productos y los volúmenes de producción del Caso 5

Producto	Secuencia de proceso por departamentos	Cantidad de productos procesados mensualmente	Producto	Secuencia de proceso por departamentos	Cantidad de productos procesados mensualmente
a	1-5-4-10	1,000	d	1-7-8-10	1,000
b	2-6-3-9	2,000	e	2-5-6-9	2,000
c	2-10-1-9	3,000	f	1-7-4-10	4,000

Los costos del manejo de materiales entre los departamentos se asume que son constantes.

#### 4.2.5.2 Datos de entrada

A partir de la Tabla 4.12 se elaboran los datos de entrada para el programa (ver Tabla 4.13 a)).

**Tabla 4.13** Matriz de entrada y cuadrícula de 2 x 5 del Caso 5

1,5,1000	5,4,1000	4,10,1000	2,6,2000	6,3,2000	3,9,2000	2,10,3000	10,1,3000	1,9,3000
1,7,1000	7,8,1000	8,10,1000	2,5,2000	5,6,2000	6,9,2000	1,7,4000	7,4,4000	4,10,4000

a) Datos de entrada del Caso 5

Configurar la distribución de planta en una cuadrícula de  $2 \times 5 = 10$  localizaciones (ver Tabla 4.13 b)).

1		2		3		4		5	
6		7		8		9		10	

b) Cuadrícula de 10 localizaciones del Caso 5

#### 4.2.5.3 Resultados

En la Tabla 4.14 a) se presentan los resultados de las 9 iteraciones del algoritmo.

**Tabla 4.14** Iteraciones del algoritmo y la función objetivo del Caso 5

Iteration (n)	Objective Function (f)	Grid Values (x)
11	67000	1 9 10 3 6 7 2 5 8 4
12	65000	2 10 8 4 1 6 3 9 7 5
13	61000	3 9 6 5 2 7 4 1 8 10
14	56000	4 8 6 10 3 1 5 7 2 9
15	79000	5 7 10 1 3 8 6 9 4 2
16	56000	4 8 6 10 3 1 5 7 2 9
17	63000	7 6 5 3 1 4 8 10 9 2
18	58000	8 1 10 4 2 6 9 5 7 3
19	56000	9 2 6 5 3 1 10 7 8 4
20	56000	4 8 6 10 3 1 5 7 2 9

a) Resultados de las 9 iteraciones del algoritmo



<b>r</b>	<b>x*</b>	<b>f*</b>
1	1 9 10 3 6 7 2 5 8 4	67,000
2	2 10 8 4 1 6 3 9 7 5	65,000
3	3 9 6 5 2 7 4 1 8 10	61,000
<b>4</b>	<b>4 8 6 10 3 1 5 7 2 9</b>	<b>56,000</b>
5	5 7 10 1 3 8 6 9 4 2	79,000
6	6 3 10 2 8 4 7 5 9 1	60,000
7	7 6 5 3 1 4 8 10 9 2	63,000
8	8 1 10 4 2 6 9 5 7 3	58,000
9	9 2 6 5 3 1 10 7 8 4	56,000

b) Valores de la función objetivo

Los valores de la función objetivo se presentan en la Tabla 4.14 b) para  $r=1$  hasta 9, obteniéndose como la mejor distribución para  $r=4$  con un costo de \$ 56,000.

El tiempo de procesamiento para la obtención de la mejor solución es 0.050 segundos.

#### 4.2.6 Caso 6: El-Baz (2004)

##### 4.2.6.1 Definición

Se ha tomado el Ejemplo III del artículo mencionado que consta de 12 departamentos. La lista de artículos y sus volúmenes de producción correspondiente a 18 artículos se indica en la Tabla 4.15.

Los costos del manejo de materiales entre los departamentos se asume que es constante.

**Tabla 4.15** Datos generales del Caso 6

<b>Item</b>	<b>Producción</b>	<b>Rutas de proceso</b>	<b>Item</b>	<b>Producción</b>	<b>Rutas de proceso</b>
P01	100	M1-M4-M2-M6	P10	180	M3-M12-M8-M10
P02	120	M3-M5-M12-M10	P11	80	M2-M6-M2-M4-M6
P03	50	M2-M4-M12-M6	P12	60	M11-M9-M10-M8
P04	45	M5-M8-M10	P13	70	M1-M4-M5-M7
P05	60	M3-M5-M12-M6	P14	150	M2-M4-M6-M2-M6
P06	80	M4-M2-M4-M6	P15	120	M3-M7-M9-M10
P07	90	M1-M5-M9	P16	120	M3-M10-M12-M9-M12
P08	120	M3-M7-M10-M4-M8	P17	100	M5-M10-M8-M9-M12
P09	140	M1-M4-M6	P18	90	M2-M8-M9-M10

##### 4.2.6.2 Datos de entrada

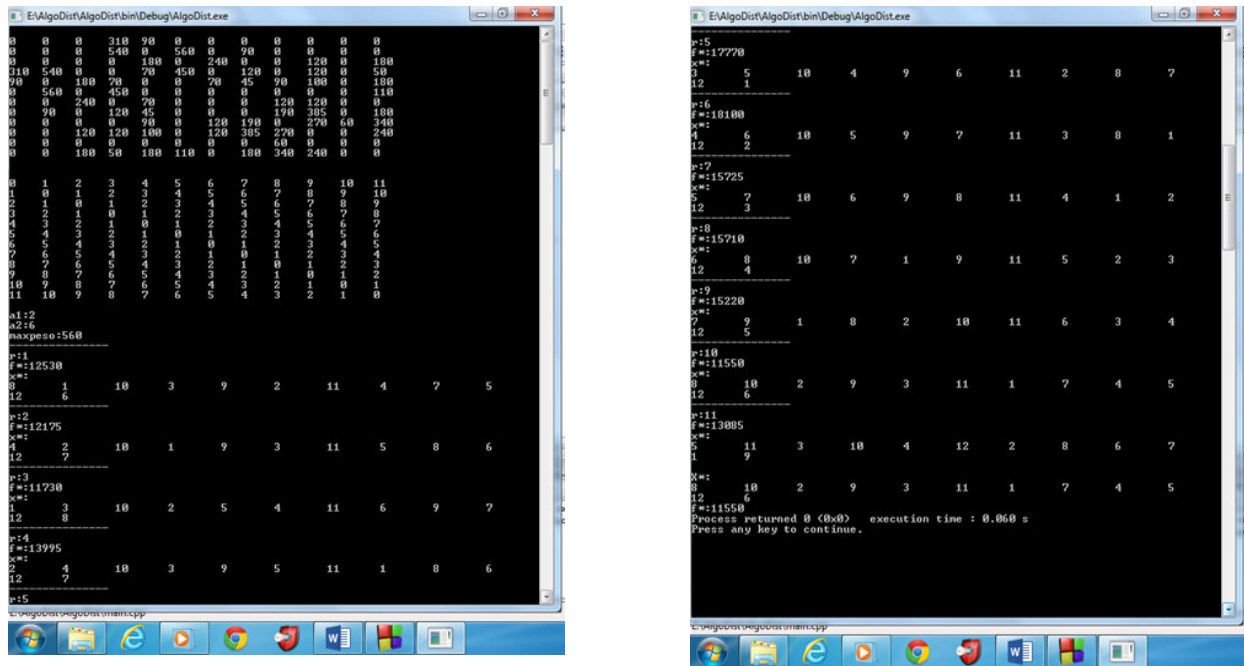
Los datos de entrada se generan directamente desde la Tabla 4.15. La configuración es una distribución de flujo en línea con multi productos en una cuadrícula de  $1 \times 12 = 12$  localizaciones.

##### 4.2.6.3 Resultados

En la Tabla 4.16 a) se presentan los resultados de las 11 iteraciones del algoritmo.

Se presentan en la Tabla 4.16 b) las evaluaciones de la función objetivo para  $r=1$  hasta 11.

**Tabla 4.16** Iteraciones del algoritmo y la función objetivo del Caso 6



a) Resultados de las 11 iteraciones

r	x*	f*
1	8 1 10 3 9 2 11 4 7 5 12 6	12,530
2	4 2 10 1 9 3 11 5 8 6 12 7	12,175
3	1 3 10 2 5 4 11 6 9 7 12 8	11,730
4	2 4 10 3 9 5 11 1 8 6 12 7	13,995
5	3 5 10 4 9 6 11 2 8 7 12 1	17,770
6	4 6 10 5 9 7 11 3 8 1 12 2	18,100
7	5 7 10 6 9 8 11 4 1 2 12 3	15,725
8	6 8 10 7 1 9 11 5 2 3 12 4	15,710
9	7 9 1 8 2 10 11 6 3 4 12 5	15,220
10	8 10 2 9 3 11 1 7 4 5 12 6	11,550
11	5 11 3 10 4 12 2 8 6 7 1 9	13,085

b) Valores de la función objetivo

La mejor solución corresponde para  $r=10$ , cuya distribución de máquinas de flujo en línea sería:

Distribución de planta de flujo en línea:

M 7	M 3	M 5	M 9	M 10	M 12	M 8	M 1	M 4	M 2	M 6	M 11
-----	-----	-----	-----	------	------	-----	-----	-----	-----	-----	------

Instalaciones:

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

Localizaciones:

8	10	2	9	3	11	1	7	4	5	12	6
---	----	---	---	---	----	---	---	---	---	----	---

El tiempo de procesamiento para la obtención de la mejor solución es 0.060 segundos.

#### 4.2.7 Caso 7: Amir Sadrzadeh (2012)

##### 4.2.7.1 Definición

Este ejemplo se ha tomado del artículo cuyo autor se menciona arriba y consta de 24 departamentos. La lista de artículos y sus volúmenes de producción correspondiente a 38 productos se indica en la Tabla 4.17. Los costos del manejo de materiales entre los departamentos se asume que son constantes.

**Tabla 4.17** Lista de artículos y volumen de producción del Caso 7

Rutas	Producción	Producto
22-1-13-21	130	01
3-20-24	150	02
14-7-23-24	125	03
15-6-18-8-12	145	04
15-6-18-8-12-5	65	05
9-17-10	78	06
9-17-10	95	07
4-16	160	08
22-1-13-21	85	09
2-11-19-5-21	105	10
3-20	130	11
3-20	140	12
2-11-19	150	13
2-11-19-5	185	14
3-20	78	15
22-1-13-21	95	16
1-13-22	160	17
15-6-18-8-12	85	18
4-16	105	19
10-17-12	130	20
4-16	105	21
2-5-11-19	130	22
20-12	150	23
7-14-23	185	24
15-6-18-8-10	145	25
15-6-18-8-12	65	26
9-17	95	27
6-18-8-12	160	28
3-20-17	85	29
14-7-23-24-16	105	30
22-1-13-21-2	130	31
3-20	150	32
11-19-5	125	33
20-12-21	145	34
16-11-14	65	35
4-16	78	36
4-16	95	37
1-13-19	160	38

##### 4.2.7.2 Datos de entrada

La matriz de datos de entrada para el programa se construye directamente desde la Tabla 4.17.

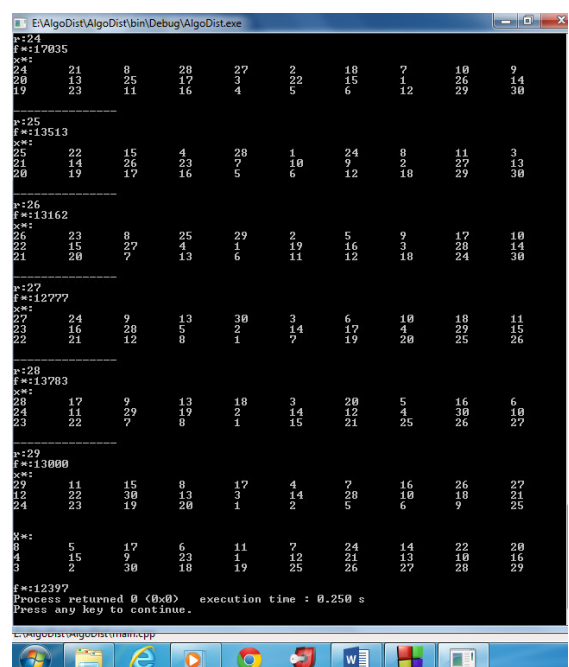
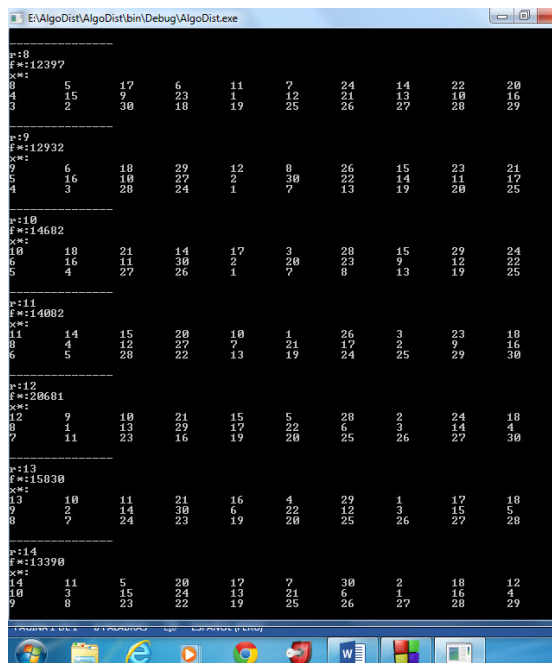
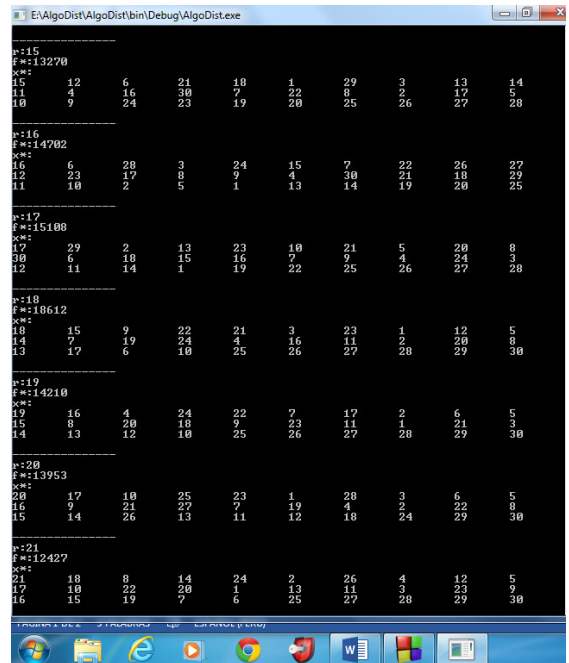
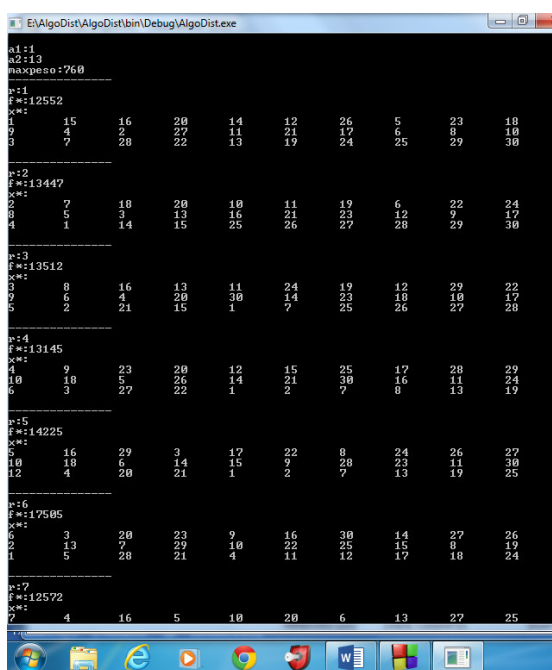
Las instalaciones serán distribuidas en una cuadrícula de 5 x 6= 30 localizaciones.

#### 4.2.7.3 Resultados

En la Tabla 4.18 a) se presentan los resultados de las 29 iteraciones del algoritmo.

Se presenta en la Tabla 4.18 b) las evaluaciones de la función objetivo para  $r=1$  hasta 29.

**Tabla 4.18** Iteraciones del algoritmo y la función objetivo del Caso 7



a) Resultados de las 29 iteraciones del algoritmo

<b>r</b>	<b>x*</b>	<b>f*</b>
1	1 15 16 20 14 12 26 5 23 18 9 4 2 27 11 21 17 6 8 10 3 7 28 22 13 19 24 25 29 30	12,552
2	2 7 18 20 10 11 19 6 22 24 8 5 3 13 16 21 23 12 9 17 4 1 14 15 25 26 27 28 29 30	13,447
3	3 8 16 13 11 24 19 12 29 22 9 6 4 20 30 14 23 18 10 17 5 2 21 15 1 7 25 26 27 28	13,512
4	4 9 23 20 12 15 25 17 28 29 10 18 5 26 14 21 30 16 11 24 6 3 27 22 1 2 7 8 13 19	13,145
5	5 16 29 3 17 22 8 24 26 27 10 18 6 14 15 9 28 23 11 30 12 4 20 21 1 2 7 13 19 25	14,225
6	6 3 20 23 9 16 30 14 27 26 2 13 7 29 10 22 25 15 8 19 1 5 28 21 4 11 12 17 18 24	17,505
7	7 4 16 5 10 20 6 13 27 25 3 14 8 12 21 11 26 19 9 15 2 1 18 17 22 23 24 28 30	12,572
8	<b>8 5 17 6 11 7 24 14 22 20 4 15 9 23 1 12 21 13 10 16 3 2 30 18 19 25 26 27 28 29</b>	<b>12,397</b>
9	9 6 18 29 12 8 26 15 23 21 5 16 10 27 2 30 22 14 11 17 4 3 28 24 1 7 13 19 20 25	12,932
10	10 18 21 14 17 3 28 15 29 24 6 16 11 30 2 20 23 9 12 22 5 4 27 26 1 7 8 13 19 25	14,682
11	11 14 15 20 10 1 26 3 23 18 8 4 12 27 7 21 17 2 9 16 6 5 28 22 13 19 24 25 29 30	14,082
12	12 9 10 21 15 5 28 2 24 18 8 1 13 29 17 22 6 3 14 4 7 11 23 16 19 20 25 26 27 30	20,681
13	13 10 11 21 16 4 29 1 17 18 9 2 14 30 6 22 12 3 15 5 8 7 24 23 19 20 25 26 27 28	15,830
14	14 11 5 20 17 7 30 2 18 12 10 3 15 24 13 21 6 1 16 4 9 8 23 22 19 25 26 27 28 29	13,390
15	15 12 6 21 18 1 29 3 13 14 11 4 16 30 7 22 8 2 17 5 10 9 24 23 19 20 25 26 27 28	13,270
16	6 28 3 24 15 7 22 26 27 12 23 17 8 9 4 30 21 18 29 11 10 2 5 1 13 14 19 20 25	14,702
17	17 29 2 13 23 10 21 5 20 8 30 6 18 15 16 7 9 4 24 3 12 11 14 1 19 22 25 26 27 28	15,108
18	18 15 9 22 21 3 23 1 12 5 14 7 19 24 4 16 11 2 20 8 13 17 6 10 25 26 27 28 29 30	18,612
19	16 4 24 22 7 17 2 6 5 15 8 20 18 9 23 11 1 21 3 14 13 12 10 25 26 27 28 29 30	14,210
20	20 17 10 25 23 1 28 3 6 5 16 9 21 27 7 19 4 2 22 8 15 14 26 13 11 12 18 24 29 30	13,953
21	18 8 14 24 2 26 4 12 5 17 10 22 20 1 13 11 3 23 9 16 15 19 7 6 25 27 28 29 30	12,427
22	22 12 5 8 30 21 1 28 7 14 18 29 23 2 15 9 10 27 24 11 17 16 3 6 13 14 19 20 25 26	15,178
23	23 22 10 14 29 4 19 6 1 2 28 12 24 13 3 15 8 5 30 11 18 17 7 9 16 20 21 25 26 27	14,430
24	24 21 8 28 27 2 18 7 10 9 20 13 25 17 3 22 15 1 26 14 19 23 11 16 4 5 6 12 29 30	17,035
25	25 22 15 4 28 1 24 8 11 3 21 14 26 23 7 10 9 2 27 13 20 19 17 16 5 6 12 18 29 30	13,513
26	26 23 8 25 29 2 5 9 17 10 22 15 27 4 1 19 16 3 28 14 21 20 7 13 6 11 12 18 24 30	13,162
27	27 24 9 13 30 3 6 10 18 11 23 16 28 5 2 14 17 4 29 15 22 21 12 8 1 7 19 20 25 26	12,777
28	28 17 9 13 18 3 20 5 16 6 24 11 29 19 2 14 12 4 30 10 23 22 7 8 1 15 21 25 26 27	13,783
29	29 11 15 8 17 4 7 16 26 27 12 22 30 13 3 14 28 10 18 21 24 23 19 20 1 2 5 6 9 25	13,000

b) Valores de la función objetivo

En la Tabla 4.18 b) se observa que se ha obtenido como la mejor solución para  $r=8$ , con un costo de \$ 12,397. La distribución de planta que corresponde a esta representación se presenta en la Tabla 4.19 b); la numeración de las localizaciones es el establecido en el método AlgoDist, es decir de izquierda a derecha y fila por fila.

A manera de comparación se ilustra la mejor solución obtenida hasta el momento por Sadrzadeh (2012), cuyo costo es \$ 11,662 (ver Tabla 4.19 a)).

En la Tabla 4.19 c) se ilustra la mejor distribución obtenida para el mismo caso por Mak *et al.*, (1998) con un costo de \$ 12,982.

**Tabla 4.19** Distribución de planta de las mejores soluciones del Caso 7

1	2	3	4	5	6
	4	23	7	14	
7	8	9	10	11	12
	16	24	1	22	
13	14	15	16	17	18
	3	20	13	19	11
19	20	21	22	23	24
9	17	12	21	5	2
25	26	27	28	29	30
	10	8	18	6	15

a) Mejor solución obtenida por Sadrzadeh (2012)

1	2	3	4	5	6
	15	22	21	11	2
7	8	9	10	11	12
	6	1	13	19	5
13	14	15	16	17	18
	18	8	12	20	3
19	20	21	22	23	24
	10	17	9	14	7
25	26	27	28	29	30
					23

b) Mejor solución obtenida por el algoritmo **AlgoDist**

1	2	3	4	5	6
	3	24	16	4	22
7	8	9	10	11	12
9	20	23	5	13	1
13	14	15	16	17	18
17	12	7	19	21	
19	20	21	22	23	24
10	8	14	11	2	
25	26	27	28	29	30
	18	6	15		

c) Mejor solución obtenida por Mak *et al.* (1998)

El tiempo de procesamiento para la obtención de la mejor solución es 0.250 segundos.

### 4.3 Evaluación comparativa

La Tabla 4.20 representa una comparación entre el algoritmo AlgoDist propuesto y las soluciones obtenidas por otros autores: Chan and Tansri (1994), Mak *et al.* (1998), Gaither *et al.* (2000), El-Baz (2004), Chase *et al.* (2009), Zadrzadeh (2012) y Salas (2014-2).

**Tabla 4.20** Comparación entre AlgoDist propuesto y otros métodos

Casos	Referencia	Modelo	Tamaño (N)	# Productos	Función objetivo	# Corridas Experimentos (*)	Otros
Caso 1 4.2.1	Salas B. J. A. (2014-2)	PSD	6	6	3,426	-	
	AlgoDist (2015)	Greedy	6	6	3,426	1	
Caso 2 4.2.2	Chase <i>et al.</i> (2009)	PSD	8	-	328	-	
	AlgoDist (2015)	Greedy	8	-	228	1	
	El-Baz (2004) Ejemplo I	AG	9	-	4,818	30	P=20, G=40
Caso 3 4.2.3	AlgoDist (2015)	Greedy	9	-	4,862	1	
	Chang & Tansri (1994)	AG	9	-	4,938	10 (*)	P=20, G=40
	AlgoDist (2015)	Greedy	9	-	4,862	1	
	Mak <i>et al.</i> (1998)	AG	9	-	5,225	10 (*)	P=20, G=40
	AlgoDist (2015)	Greedy	9	-	4,862	1	G=40
Caso 4 4.2.4	Buffa <i>et al.</i> (1964)	PSD	10	-	210	-	
	WinQSB	CRAFT	10	-	109	1	
	AlgoDist (2015)	Greedy	10	-	108	1	
Caso 5 4.2.5	Gaither <i>et al.</i> (2000)	PSD	10	6	53,000	-	
	AlgoDist (2015)	Greedy	10	6	56,000	1	
Caso 6 4.2.6	El-Baz (2004) Ejemplo III	GA	12	18	11,440		
	AlgoDist (2015)	Greedy	12	18	11,550		
Caso 7 4.2.7	Zadrzadeh (2012) Ejemplo I	AG	24	38	11,662	30	
	AlgoDist (2015)	Greedy	24	38	12,397	1	
	Mak <i>et al.</i> (1998)	AG	24	38	12,982	30	
	AlgoDist (2015)	Greedy	24	38	12,397	1	

Para el Caso 1, en el curso mencionado el docente utiliza el concepto de Disposición Ideal (DI), si la distribución que se consigue por el procedimiento PSD supera el 80% es una buena distribución. En este caso se ha obtenido 84.94% (2,910/3,426). El mejor resultado del método propuesto es exactamente igual al valor obtenido por el autor, lo que quiere decir que el algoritmo produce un buen resultado con ese enfoque.

Como el Caso 2, se ha escogido el Ejercicio 11 del texto de Chase *et al.* (2009) para comparar los cálculos que hacen los estudiantes de Ingeniería Industrial que usan el libro en mención como texto guía para los ejemplos y ejercicios en el curso de Distribución de Planta. El resultado del algoritmo 228 es mucho mejor que el valor obtenido en clase 328 para una distribución al azar.

Para el Caso 3, comparando los resultados de los trabajos de tres autores el valor obtenido por el algoritmo se ubica en el segundo lugar con 1% de diferencia al mejor valor del momento 4,818. El resultado de El-Baz está primero, pero con 30 corridas frente a sólo una corrida del AlgoDist. El Algoritmo propuesto ha superado los trabajos de Chan & Tansri (1994) y de Mak *et al.* (1998).

El Caso 4 consiste en el ejemplo de una oficina de 10 departamentos, la que se considera como un modelo precursor para la distribución de planta con CRAFT. Los autores proponen una distribución inicial cualquiera, cuya evaluación de la función objetivo da 210, esta solución es mejorada a 109 a través del procedimiento heurístico que proponen. El mismo caso, con el programa WinQSB, se obtiene en 5 iteraciones el valor de 109. Con el método propuesto AlgoDist, en 9 iteraciones se obtiene el mejor resultado de 108.

Para el Caso 5, el resultado del algoritmo propuesto es 56,000 y está próximo en 6% a la mejor solución obtenida de 53,000 por el autor en su última edición del libro.

Para el Caso 6, el valor obtenido por el método propuesto está a menos del 1% de diferencia al mejor resultado 11,440 que tiene el autor en mención. La ventaja del algoritmo propuesto es que trata con gran flexibilidad los casos del SRFLP al definir la cuadrícula de distribución de planta de una fila por varias columnas.

Para el Caso 7, el resultado del método propuesto es 12,397 que se ubica en el segundo puesto, con 6% de error frente al mejor resultado del momento de Zadrzadeh (2012), pero con una sola corrida frente a 30 corridas del otro autor y tiempo de procesamiento pequeño. En cambio, el resultado obtenido por el método AlgoDist es mejor si se compara con el valor obtenido por Mak *et al.* (1998).



## Capítulo 5 Conclusiones y trabajos futuros

El trabajo plantea los Problemas de Distribución de Planta como un problema de diseño y propone un algoritmo goloso para resolver la localización de las instalaciones industriales con multi productos y rutas de procesamiento, para minimizar el costo total de manejo de materiales, bajo el enfoque de una heurística de construcción.

El algoritmo propuesto es mucho más rápido de implementar que otras técnicas desarrolladas. En los experimentos numéricos presentados en el Capítulo 4 se muestra que el método propuesto tiene solución eficiente de buena calidad en comparación con los Algoritmos Genéticos de Mak *et al.*, (1998) y Chan & Tansri (1994) con operadores genéticos PMX, OM y CX. Por lo tanto, el algoritmo propuesto es un buen enfoque para resolver los Problemas de Distribución de Planta.

El AlgoDist también se puede utilizar para calcular la población inicial de otros algoritmos.

El enfoque dado como multi línea al área de distribución de la planta, se puede flexibilizar considerando la cuadrícula como una fila simple y así poder aplicar el algoritmo a Sistemas de Manufactura Flexible FMS, tal como se puede observar en los resultados que se presentan en la Tabla 4.16, Caso 6.

En la mayoría de las formulaciones el término  $c_{ij}$  se define como el costo necesario para el manejo de una unidad de material desde la instalación  $i$  hasta la instalación  $j$ , aquí no interesa la naturaleza del producto que está pasando. En el presente trabajo el  $c_{ij}$  tiene la misma definición pero está directamente relacionado con un producto en particular. Es decir  $c_{13}$  puede ser 2 para el producto  $a$  y 2.5 para el producto  $c$  y ésta se refleja en el factor  $w_{ij}$  mostrado en la Tabla 3.2.

Como trabajo futuro se puede implementar el AlgoDist en un software para resolver los ejercicios de los textos de Distribución de Planta en Ingeniería Industrial, para los casos de diseño de sistemas de manufactura multi productos con rutas de procesamiento.

También se recomienda realizar un estudio comparativo con los algoritmos propuestos para los Sistemas de Manufactura Flexible FMS.

## Referencias bibliográficas

- Adam, E. & Ebert, R. (1991). Administración de la Producción y las Operaciones, Prentice-Hall Hispanoamericana.
- Ahuja, R.K., Orlin, J.B. & Tiwari, A. (2000). A greedy genetic algorithm for the quadratic assignment problem, *Computers and Operations Research*, 27, pp. 917-934.
- Aiello, G., La Scalia, G. & Enea, M. (2013). A non dominated ranking Multi Objective Genetic Algorithm and electre method for unequal area facility layout problems, *Expert Systems with Applications*, 40, 4812-4819.
- Anstreicher, K.M., Brixu, N.W., Lindereth, J. & Goux, P. (2002) "Solving large quadratic assignment problems on computational grids", *Mathematical Programming*, 91, pp. 563-588.
- Armour, G.C. & Buffa, E.S. (1963). A heuristic algorithm and simulation approach to relative location of facilities, *Management Science*, 9, 294-309.
- Balakrishnan, J. Cheng, C.H. & Wong, K.F. (2003). FACOPT: A User Friendly FACility Layout OPTimization System, *Computers & Operations Research*, vol. 30, pp. 1625-1641.
- Buffa, E.S. (March-April, 1955). Sequence Analysis for Functional Layouts, *Journal of Industrial Engineering*, vol. 6, no. 2, pp. 12-13.
- Buffa, E.S., Armour, G.C. & Vollman, T.E. (March-April, 1964). Allocating Facilities with CRAFT, *Harvard Business Review*, vol. 42, no. 2, pp. 136-159.
- Chang, Yih-Long (2000). WinQSB.
- Chase, R., Aquilano N. & Jacobs, R. (2009). Administración de la Producción y Operaciones para una ventaja competitiva, 10ª. edición, McGraw-Hill.
- Datta, D., Amaral, A. & Figueira, J. (2011). Single row facility layout problem using a permutation-based genetic algorithm, *European Journal of Operational Research*, 213:388-394.
- Díaz, B., Jarufe, B. & Noriega, M.T. (2008). Disposición de planta, 2ª. edición, Fondo Editorial, Universidad de Lima (Perú).

- Diego-Más, J., Santamarina-Siurana, C., Alcaide-Marzal, J. & Cloquell-Ballester, V. (2007). Solving facility layout problems with strict geometric constraints using a two-phase genetic algorithm Francis, *International Journal of Production Research*.
- El-Baz, M.A. (2004). A genetic algorithm for facility layout problems of different manufacturing environments, *Computers & Industrial Engineering*, 47, 233-246.
- Francis, Richard L. & White, John A. (1974). Facilities Layout and Location, Prentice-Hall, Englewood Cliffs, NJ.
- Gaither, N. & Frazier, G. (2000). Administración de Producción y Operaciones, International Thomson Editores.
- García, H.L. (2011). Genetic Approaches for the Unequal Area Facility Layout Problem, Universidad de Córdoba, Tesis Doctoral, España.
- García-Hernández, L., Pierreval, H., Salas-Morera, L. & Arauzo-Axofra, A. (2013). Handling qualitative aspects in Unequal Area Facility Layout Problem: An Interactive Genetic Algorithm, *Applied Soft Computing*, 13, 1718-1727.
- Gilmore, P.C. (1962). Optimal and suboptimal algorithms for the quadratic assignments problem, *SIAM Journal on applied mathematics*, 10, pp. 305-313.
- Glover, F. (1998). A template for scatter search and path relinking, *Lecture notes in computer science*, 1363 13-54.
- Goldberg, D.E. (1999). Genetic Algorithms in Search, Optimization and Machine Learning, *Addison-Wesley*, Boston, MA, USA.
- Hasan, M. & Osman (1995). Local Search Algorithms for the Maximal Planar Layout Problem. *International Transactions in Operational Research*, vol. 2, pp. 89-106.
- Hicks, P.E. & Cowen, T.E. (1976). CRAFT-M for layout Rearrangement, *Industrial Engineering*, vol. 8, pp. 30-35.
- Johnson, R.V. (1982). SPACECRAFT for multi-floor layout planning, *Management Science*, vol. 28, no. 4, pp. 407-417.

- Konak, A., Kulturel-Konak, S., Norman, B.A. & Smith, A.E. (2006). A new mixed integer programming formulation for facility layout design using flexible bays, *Operations Research Letters*, 34, 660-672.
- Koopmans, T.C. & Beckmann, M. (Jan., 1957). Assignment Problems and the Location of Economic Activities, *Econometrica*, vol. 25, no. 1, 53-76.
- Krajewski, L.J., Ritzman, L.P. & Malhotra, M.K. (2008). ADMINISTRACIÓN DE OPERACIONES Procesos y cadena de valor, Ed. PEARSON Educación, México.
- Kothari, R. & Ghosh, D. (2012). Scatter search algorithms for the single row facility layout problem, P & QM Area, IIM Ahmedabad, Vastrapur, Ahmedabad 380015, Gujarat, India.
- Kratka, J., Tosić, D., Filipović, V. & Dugosija, D. (2011). A new genetic representation for quadratic assignment problem, *Yugoslav Journal of Operations Research*, 21, no. 2, 225-238.
- Lee, R. & Moore, J.M. (1967). CORELAP-computerized relationship layout planning. *The Journal of Industrial Engineering*, vol. 18, pp. 195-200.
- Loiola, E.M., Abreu, N.M.M, Boaventura-Netto, P.O., Hahn, P. & Querido, T. “Analytical survey for the quadratic assignment problem”, documento electrónico disponible en: [http://www.seas.upenn.edu/qaplib/QAP\\_Survey\\_04.pdf#search='boaventura%20netto](http://www.seas.upenn.edu/qaplib/QAP_Survey_04.pdf#search='boaventura%20netto), fecha de último acceso: 17/11/2014.
- Martí, O.N., Segura, C. & Verdejo, A. (2013). Algoritmos correctos y eficientes, Ed. IBERGACETA PUBLICACIONES, S.L., Madrid.
- Mckendall, A.R. & Jaramillo, J.R. (2006). A Tabu Search Heuristic for the Dynamic Space Allocation Problem, *Computers & Operations Research*, vol. 33, pp. 768-789.
- Meller, R.D. & Gau, K.Y. (1996). The facility Layout Problem: recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, vol. 15, no. 5, pp. 351-366.
- Meller, R.D. Chen, W. & Sherali, H.D. (2007). Applying the Sequence-pair Representation to Optimal Facility Layout Problems, *Operations Research Letters*, vol. 35, no. 5, pp. 651-659.
- Montreuil, B. (1990). A Modelling Framework for Integrating Layout Design and Flow Network Design. *Proceedings of the Material Handling Research Colloquium*, pp. 43-58.
- Muther, R. (1961). Systematic Layout Planning, Industrial Education Institute, Boston.

- Nawaz, R.K.S., Glette, K., Nizam, K.K., Hovin, M. & Torresen, J. (2013). Adaptive variable neighborhood search for solving multi-objective facility layout problems with unequal area facilities, *Swarm and Evolutionary Computation*, 8, 1-12.
- Noori, H. & Radford R. (1997). Administración de Operaciones y Producción, Calidad total y respuesta sensible rápida, McGraw-Hill.
- Nugent, C.E. Vollman, T.E. & Ruml, J. (1968). An experimental comparison of techniques for the assignment of facilities to locations, *Operations Research*, 16, pp. 150-173.
- Osman, I.H. (1995). “An introduction to Meta-Heuristics”, *Operational Research, Tutorial Papers*, pp. 92-122.
- Pierreval, H., Cauxx, C., Paris, J.L. & Viguiet, F. (2003). Evolutionary approaches to the design and organization of manufacturing systems, *Computers & Industrial Engineering*, vol. 44, pp. 39-364.
- Sadrzadeh, A. (2012). A genetic algorithm with the heuristic procedure to solve the multi-line layout problem, *Computers & Industrial Engineering*, 62, 1055-1064.
- Salas, B.J.A. (2014-2). Curso: Distribución de Planta, apuntes de clase, Facultad de Ingeniería Industrial de la UNMSM, Lima.
- Seehof, J.M. & Evans, W.O. (1967). Automated layout design program, *The Journal of Industrial Engineering*, vol. 18, pp. 690-695.
- Shayan, E. & Chittilapilly, A. (2004). Genetic algorithm for facilities layout problems based on slicing tree structure, *International Journal of Production Research*, 42, pp. 4055-4067.
- Solimanpur, M. & Kamran, M.A. (2010). Solving facilities location problem in the presence of alternative processing routes using a genetic algorithm, *Computers & Industrial Engineering* 59, 830-839.
- Taha, H.A. (2012). Investigación de operaciones, 9ª. edición, Pearson Educación, México.
- Tam, K.Y. (1992). A simulated Annealing for Allocation Space to Manufacturing Cells, *European Journal of Operational Research*, vol. 63, pp. 332-346.
- Tam, K.Y. (1992). Genetic Algorithms, Function Optimization, and Facility Layout Design, *International Journal of Production Research*, vol. 30, pp. 63-87.

- Tate, D.M. & Smith, A.E. (1995). Unequal Area Facility Layout Using Genetic Search, *IEE Transactions*, vol. 27, pp. 465-472.
- Tompkins, J.A., White, J.A., Bozer, Y.A., Frazelle, E.H., Tanchoco, J.M. & Trevino, J. (1996). Facilities planning, John Wiley, NY.
- Tong, X. (1991). SECOT: A sequential construction technique for facility design, Pittsburgh, PA, University of Pittsburgh, Doctoral Dissertation.
- Ulutas, B.H., & Kulturel-Konak, S. (2012). An artificial immune system based algorithm to solve unequal area facility layout problem, *Expert Systems with Applications*, 39, 5384-5395.
- Wong, K.Y. & Komarudin (2010). Solving facility layout problems using flexible bay structure representation and ant system algorithm, *Expert System with Applications*, 37, 5523-5527.
- Wu, Y. & Ji, P. (2008). Solving the Quadratic Assignment Problems by a Genetic Algorithm with New Replacement Strategy, *International Journal of Information and Mathematical Sciences*, 4:3.
- Xu, G. & Papageorgiou, L.G. (2009). Process plant layout using an improvement-type algorithm, *Chemical Engineering Research and Design*, 87, 780-788.
- Yang, T., Peters, B.A. & Tu, M. (2005). Layout design for flexible manufacturing systems considering single-loop directional flow patterns, *European Journal of Operational Research*, vol. 164, pp. 440-455.
- Yongzhong, Wu & Ping, Ji (2008). Solving the Quadratic Assignment Problems by a Genetic Algorithm with a New Replacement Strategy, *International Journal of Information and Mathematical Sciences* 4:3.
- Zanakis, S.H. & Evans, J.R. (1981). "Heuristic optimization: Why, when and how to use it", *Interfaces*, 5.