

Resumen

El objetivo de este proyecto es la resolución, mediante métodos heurísticos, de una variante novedosa y poco tratada del Problema de Equilibrado de Líneas de Montaje o *Assembly Line Balancing Problem* (ALBP).

Dentro del ALBP se trata la tipología de problema *Lexicographic Bottleneck Assembly Line Balancing Problem* (LB-ALBP), cuyo objetivo es: dado un número determinado de estaciones, minimizar el tiempo de la estación más cargada (tiempo de ciclo), posteriormente el tiempo de la segunda estación más cargada, el tiempo de la tercera estación más cargada, y así sucesivamente hasta minimizar, de forma jerárquica, el tiempo de todas las estaciones.

El hecho de considerar los tiempos de la segunda estación más cargada, la tercera, etc. es importante, ya que no es lo mismo obtener una distribución de la carga de trabajo entre seis estaciones igual a 10, 10, 10, 4, 3, 3 que una distribución igual a 10, 6, 6, 6, 6, 6. A pesar de que el tiempo de ciclo en ambos casos es 10, la segunda distribución es más fiable y está mejor equilibrada. Además, el hecho de mejorar el equilibrio evita los problemas de calidad derivados de la existencia de estaciones con cargas de trabajo excesivas.

Para solucionar el LB-ALBP, inicialmente se desarrolla una heurística “principal” que consiste en ejecutar 14 heurísticas sencillas ya existentes en la literatura que resuelven el problema ALBP y quedarse con la mejor solución. A continuación, se fija la estación más cargada (es decir, las tareas asignadas a dicha estación), y se divide el problema en dos sub-problemas que se resuelven de forma independiente ejecutando otra vez las 14 heurísticas sencillas. Esta operativa se aplica sucesivamente hasta que son fijadas todas las estaciones.

En este proyecto se desarrollan diez heurísticas para la resolución del LB-ALBP, que son el resultado de la combinación de dos variantes de la heurística “principal” con seis métodos de optimización local de soluciones. Para comprobar la calidad de los procedimientos desarrollados se resuelve un total de 274 ejemplares para cada heurística. Dicha base de ejemplares se obtiene a partir de los 16 problemas que se proponen en la página web del Dr. Scholl (uno de los investigadores más importantes en el ALBP), variando el número de estaciones para cada problema.





Sumario

RESUMEN	1
SUMARIO	3
1. GLOSARIO	5
2. INTRODUCCIÓN	7
2.1. Objetivos	8
2.2. Alcance.....	9
3. ESTADO DEL ARTE DEL PROBLEMA DE EQUILIBRADO DE LÍNEAS DE MONTAJE	11
3.1. Líneas de montaje.....	11
3.1.1. Introducción histórica.....	11
3.1.2. Conceptos básicos	11
3.1.3. Tipos de líneas de montaje.....	12
3.2. Tipos de problemas de líneas.....	16
3.2.1. Problemas de equilibrado de líneas de montaje (ALBP)	16
3.2.2. Problemas de diseño de líneas de montaje (ALDP).....	19
3.3. Algoritmos de resolución de problemas de equilibrado de líneas de montaje.....	20
3.3.1. Algoritmos para la resolución de SALBP's	20
3.3.2. Algoritmos para la resolución de GALBP's.....	24
3.3.3. Algoritmos para la resolución de ALDP's.....	25
3.3.4. Principales funciones objetivo para el problema de equilibrado de líneas de montaje.....	25
4. DESCRIPCIÓN DEL <i>LEXICOGRAPHIC BOTTLENECK ASSEMBLY LINE BALANCING PROBLEM</i> (LB-ALBP)	31
4.1. Definición y características principales del LB-ALBP	31
4.2. Estado del arte del LB-ALBP	32
4.2.1. Medida de la calidad de una solución.....	33
4.2.2. Algoritmos para la resolución del LB-ALBP.....	34
5. RESOLUCIÓN DEL LB-ALBP	37
5.1. Procedimientos heurísticos para la resolución del LB-ALBP	37
5.1.1. Heurística "principal" para la resolución del LB-ALBP	37
5.1.1.1. Definición de la heurística "principal" para la resolución del LB-ALBP.....	37



5.1.1.2. Dos opciones de desarrollo de la heurística “principal”	38
5.1.1.3. Heurísticas base	39
5.1.1.3.1 Heurísticas de una sola pasada	40
5.1.1.3.2 Heurística de Hoffmann	42
5.1.2. Procedimientos de optimización local de soluciones	43
5.1.2.1. Método 1	45
5.1.2.2. Método 2	47
5.1.2.3. Método 3	47
5.1.2.4. Método 4	49
5.1.2.5. Método 5	50
5.1.2.6. Método 6	50
5.1.3. Heurísticas para la resolución del LB-ALBP	51
5.2. Experiencia computacional: ejemplares	52
5.3. Experiencia computacional: análisis de resultados	54
5.3.1. Comparación: Heurísticas diseñadas – Heurísticas base	54
5.3.2. Comparación: heurísticas diseñadas	56
5.3.2.1. Análisis comparativo de las heurísticas diseñadas	56
5.3.2.2. Criterios de utilización de las heurísticas diseñadas	60
5.3.2.2.1 Número de tareas	60
5.3.2.2.2 Número de estaciones	62
5.3.2.2.3 Criterio de <i>Order Strength</i> (OS)	64
5.3.3. Comparación: soluciones óptimas	66
CONCLUSIONES	69
COSTES DEL ESTUDIO EFECTUADO	71
IMPACTO AMBIENTAL DEL ESTUDIO	73
REFERENCIAS BIBLIOGRÁFICAS	75



1. Glosario

- i:** índice de las tareas.
- j:** índice de las estaciones de trabajo.
- T:** tiempo de ciclo del sistema.
- t_i :** tiempo de proceso de la tarea i .
- t_j :** carga de trabajo de la estación j .
- m:** número de estaciones de la una línea de montaje.
- n:** número de tareas a desarrollar para la fabricación de una unidad de producto en una línea de montaje.
- $c_{i,j}$:** coste de asignar la tarea i a la estación j .
- δ :** parámetro que sirve para comparar dos soluciones de un mismo problema.
- PP_i :** peso posicional de la tarea i .
- NP_i :** número de tareas predecesoras de la tarea i .
- NS_i :** número de tareas sucesoras de la tarea i .
- NSI_i :** número de tareas sucesoras inmediatas de la tarea i .
- LB_i :** cota inferior de la tarea i .
- UB_i :** cota superior de la tarea i .
- ST_{max} :** tiempo de trabajo de la estación más cargada del sistema.
- ST_{min} :** tiempo de trabajo de la estación menos cargada del sistema.
- OS:** *order strength*, parámetro que relaciona el número de relaciones de precedencia de un proceso productivo, con su número de tareas n .





2. Introducción

En una línea de montaje (o de producción, indistintamente) un conjunto de piezas deben ser montadas o ensambladas para producir un producto intermedio o final. Para realizar este proceso se dispone de un número limitado de estaciones, en las cuales se tiene que desarrollar un conjunto determinado de tareas. Cada tarea tiene una duración determinada, las tareas tienen relaciones de precedencia entre ellas y suponen unidades indivisibles, es decir, una tarea no se puede llevar a cabo en dos estaciones distintas. Las precedencias entre tareas se pueden representar mediante un grafo (gráfico 2.1), donde los vértices simbolizan las tareas y los arcos las relaciones de precedencia entre una tarea y otra.

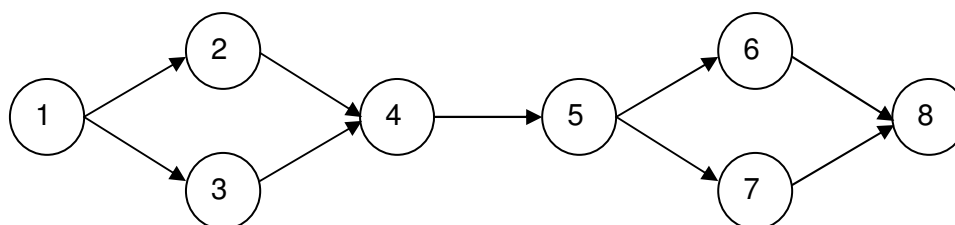


Gráfico 2.1 Ejemplo de grafo

El *Assembly Line Balancing Problem*, o problema de equilibrado de líneas de montaje, consiste básicamente en determinar una asignación de las tareas a las estaciones respetando las restricciones de precedencia (además puede haber otras restricciones como, por ejemplo, de incompatibilidad entre tareas). Dicha asignación se realiza teniendo en cuenta una función objetivo para el problema, como puede ser minimizar el número de estaciones o minimizar el tiempo de ciclo. En el caso que se trata en este proyecto, el *Lexicographic Bottleneck Assembly Line Balancing Problem* (LB-ALBP), se desea minimizar el tiempo de la estación más cargada, posteriormente, el de la segunda estación más cargada, el de la tercera más cargada, y así sucesivamente.

En el gráfico 2.2 se muestra el tiempo total asignado a cada estación en dos soluciones distintas (llamadas A y B) para un mismo problema. Aunque para ambas soluciones el tiempo de ciclo es el mismo (13 unidades de tiempo), se puede afirmar que la Solución B es más fiable que la Solución A, ya que se consigue reducir el tiempo de trabajo de la tercera, cuarta y quinta estación más cargada. De esta forma, es menos probable que una incidencia que retrase la producción dentro de la cadena de montaje, ya que se pasa de tener seis estaciones que son cuello de botella a solo tener dos.



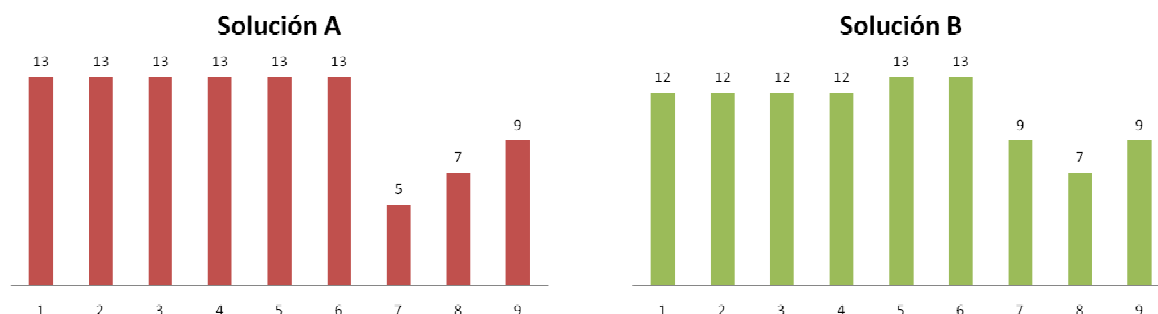


Gráfico 2.2 Ejemplo de dos soluciones factibles para un mismo problema

El equilibrado de líneas de montaje es un problema actual e importante dentro de la industria, ya que la productividad en la fabricación de un producto determinado puede variar sustancialmente según cómo se asignen las tareas a las estaciones de trabajo disponibles.

En la literatura existen procedimientos heurísticos y exactos que permiten resolver el ALBP. En este proyecto se estudian las heurísticas que solucionan el SALBP-2 (problema ALBP en el que dado el número de estaciones, se desea minimizar el tiempo de ciclo), ya que con éstas se puede resolver el problema LB-ALBP también de forma heurística, y/o diseñar nuevos procedimientos heurísticos basados en ellos. Todos estos procedimientos se han programado y ejecutado con el programa Microsoft Visual Basic 2005 Express Edition.

2.1. Objetivos

El objetivo de este proyecto es resolver el LB-ALBP mediante procedimientos heurísticos. En este problema se desea minimizar el tiempo de la estación más cargada, posteriormente el de la segunda más cargada, el de la tercera más cargada, y así sucesivamente.

Para la resolución del problema, inicialmente se parte de una heurística “principal” que consiste en ejecutar 14 heurísticas existentes en la literatura para el SALBP-2 y quedarse con la mejor solución. A continuación, se fija la estación más cargada, es decir, las tareas asignadas a dicha estación, y se divide el problema principal en dos sub-problemas que nuevamente se resuelven de forma independiente con las 14 heurísticas de la literatura. Esta operativa se repite reiteradamente hasta fijar todas las estaciones. A partir de la combinación de dos variantes de la heurística “principal” descrita, con seis métodos de optimización local diseñados (que se aplican a las heurísticas de la literatura cada vez que se ejecutan), se desarrollan un total de 10 heurísticas nuevas.



A continuación se comparan los resultados obtenidos con los de las heurísticas ya existentes en la literatura, para comprobar en qué grado se logra una mejora. Obviamente, nunca se podrá empeorar, ya que la heurística desarrollada en este proyecto parte de la mejor solución de la ejecución de las heurísticas ya existentes en la literatura.

Además, se comparan los resultados obtenidos con las 10 heurísticas entre sí, para poder decidir qué heurística es la mejor valorando la calidad de la solución y el tiempo de ejecución empleado.

Con el fin de evaluar la calidad de las heurísticas diseñadas, se comparan los resultados obtenidos en 16 problemas test, a partir de los cuales se desarrolla una base de datos de 274 ejemplares variando el número de estaciones. Estos ejemplares son los problemas test que se suelen usar en la comunidad internacional para probar nuevos procedimientos de resolución del problema de equilibrado de líneas de montaje.

2.2. Alcance

En este proyecto se resuelve la variante LB-ALBP del problema de equilibrado de líneas de montaje, teniendo en cuenta las siguientes hipótesis, propias del SALBP (problema más sencillo del ALBP):

- Producción en masa de un único producto homogéneo.
- Cada tarea es indivisible.
- La línea es simple.
- La línea es serial.
- Las tareas únicamente tienen entre ellas restricciones de precedencia.
- Los datos son conocidos (deterministas) y se dispone de un número de estaciones o tiempo de ciclo fijado (en nuestro caso, número de estaciones).
- La tasa de entrada de piezas a la línea es fija.
- Todas las estaciones están equipadas de forma ecuánime tanto en máquinas como en trabajadores.

Tal y como se ha especificado anteriormente, para comprobar la eficiencia y eficacia de las heurísticas diseñadas, éstas se aplican a una base de 274 ejemplares test. Para cada



ejemplar se dispone de la totalidad de sus valores de definición: tareas, duración de cada tarea y relaciones de precedencia inmediatas.

Con estos ejemplares se consigue un rango de prueba muy elevado tanto en estaciones (desde 3 hasta 51), como en número de tareas (desde 29 hasta 148), precedencias (desde 32 hasta 181) y media de duraciones individuales de las tareas (desde 6 hasta 1355 unidades de tiempo). Este rango se puede considerar representativo de los problemas que se presentan en la industria.



3. Estado del arte del Problema de Equilibrado de Líneas de Montaje

El ALBP es un problema que ha sido estudiado ampliamente por diferentes autores a lo largo de la historia, y en la actualidad aún se realizan muchas investigaciones como consecuencia de la importancia que tiene dentro de la industria.

Para exponer el estado del arte del ALBP, primeramente se presenta una introducción histórica a las líneas de montaje, los conceptos básicos que las caracterizan y las posibles clasificaciones en las que se pueden enmarcar. Posteriormente, se introducen los distintos tipos de problemas de líneas y los algoritmos que existen en la literatura para su resolución. Finalmente se resumen las principales funciones objetivo desarrolladas por los investigadores hasta la actualidad.

3.1. Líneas de montaje

3.1.1. Introducción histórica

El origen del concepto de línea de montaje se puede situar en el siglo XVI en Venecia, donde los barcos eran producidos en masa usando partes pre-manufacturadas. Ya en el año 1799, en la era industrial, Eli Whitney inventó el sistema de manufactura americano, usando los conceptos de división de trabajo y tolerancia en ingeniería, para crear ensamblados de partes de una forma repetitiva. Posteriormente, en 1913, Henry Ford instaló la primera línea de montaje móvil con dos objetivos claros: disminuir costes y permitir la producción en masa.

3.1.2. Conceptos básicos

Este apartado está basado principalmente en Scholl (1999, p.721-736) y en los trabajos de Pinto et al. (1983) y Becker y Scholl (2006, p.694-715).

A continuación se presentan los conceptos más ampliamente utilizados en los problemas de líneas de montaje:

- *Tarea*: es una unidad de trabajo indivisible que tiene asociado un tiempo de proceso. El trabajo total necesario para fabricar un producto en una línea se divide en un conjunto de tareas.



- *Relaciones de precedencia:* están definidas por las restricciones sobre el orden en el cual las operaciones se pueden ejecutar en una línea de montaje. Así, una tarea no puede procesarse hasta que todas sus predecesoras se hayan realizado. Comúnmente, dichas relaciones de precedencia se pueden representar en forma de diagrama de precedencias.
- *Estación:* es la parte de la línea de montaje donde se ejecutan las tareas. Pueden ser de naturaleza variada: desde cualquier tipo de maquinaria hasta mecanismos especializados. Los operadores asignados a las estaciones pueden ser humanos o robotizados.
- *Tiempo de ciclo:* es el tiempo disponible en cada estación para completar las tareas asignadas para una unidad de producto. Usualmente se considera igual al mayor tiempo asignado a todas las estaciones que componen la línea.
- *Carga de trabajo:* es el conjunto de tareas asignadas a una estación.
- *Tiempo de cada estación:* es la suma de los tiempos de todas las tareas asignadas a una estación.
- *Tiempo ocioso o libre:* es la diferencia entre el tiempo de ciclo y el tiempo de estación.

3.1.3. Tipos de líneas de montaje

En la literatura las líneas de montaje se clasifican de diversas formas, dependiendo de diferentes conceptos como pueden ser arquitectura, tipo de producto, duración de las tareas, tipo de operador, disciplina de entrada de piezas y flujo de piezas.

En este punto se resumen las principales clasificaciones propuestas por Becker y Scholl (2006, p.694-715), Scholl (1999, p.721-736), Plans (1999) y Ghosh y Gagnon (1989, p.637-670).

A. Tipos de líneas de montaje según su arquitectura

- a. *Línea serial:* se tienen estaciones simples colocadas en serie, por donde las tareas pasan consecutivamente de una estación a la siguiente. El transporte se suele realizar con algún mecanismo como puede ser una cinta transportadora.
- b. *Línea con estaciones en paralelo:* consiste en una única línea donde las estaciones en paralelo están permitidas. Las estaciones en paralelo



consisten en dos o más estaciones idénticas que realizan en paralelo las mismas tareas. Con este tipo de arquitectura se consigue solucionar el problema que se crea con aquellas tareas cuya duración es mayor al tiempo de ciclo ya que, de esta forma, la duración de dicha tarea se reduce proporcionalmente al número de estaciones.

- c. Líneas paralelas: son varias líneas colocadas en paralelo. Este tipo de arquitectura resulta especialmente útil en el caso de modelos múltiples, en el que se puede asignar una línea para cada modelo o familia de modelos.
- d. Líneas de dos lados: son dos líneas seriales en paralelo, donde en vez de tener una estación simple, se tienen dos estaciones opuestas en cada lado de línea (estación derecha e izquierda) que procesan simultáneamente una pieza común. Se pueden usar en cualquier tipo de producto, pero son más comunes para productos grandes, donde se requiere desarrollar tareas a ambos lados, como en los coches. De esta manera, algunas tareas pueden ser asignadas a un lado de línea (colocar la tapa del depósito de gasolina), otras pueden ser asignadas de manera indiferente a cualquiera de los dos lados (colocar maletero) y otras deben ser asignadas a ambos lados simultáneamente (colocar ruedas).
- e. Líneas circulares o cerradas: en este caso se tiene una cinta circular o cerrada que sirve a todas las estaciones. En esta, las piezas circulan mientras son tomadas por los operadores, que pueden ser humanos o robots, y una vez han terminado de procesar las piezas, las liberan de nuevo en la cinta, excepto la estación donde se realiza la última tarea, que deposita la pieza fuera de la cinta.
- f. Líneas en forma de U: las líneas en forma de U son consecuencia de la introducción del concepto de producción *just in time*. Esta disposición de las estaciones hace más flexible el sistema de producción, ya que las estaciones pueden trabajar en dos segmentos de la línea de montaje si ambos están uno en frente del otro. Esta configuración puede dar un mejor balance en la carga de las estaciones debido a que el número de posibles asignaciones tareas-estación es mayor.

B. Tipos de líneas de montaje según su flujo de piezas

- a. Líneas sincrónicas: todas las estaciones tienen un tiempo de ciclo común, consecuentemente, las piezas pasan de una estación a la siguiente al



mismo tiempo, por lo que no es necesaria la instalación de pulmones o *buffers* entre estaciones.

- b. Líneas asincrónicas: las estaciones están relacionadas a través de pulmones o *buffers* que se colocan entre ellas para almacenar las piezas que se han procesado en las estaciones precedentes. En este caso, las estaciones tienen velocidades de proceso distintas. Estas líneas conllevan un problema: tomar la decisión acerca de dónde colocar los *buffers* y el tamaño que estos deben tener.
- c. Líneas de alimentación: son sistemas de montaje compuestos en los que se tiene una o más líneas de alimentación o *feeder lines*, donde se realizan sub-ensamblados; es decir, se tiene una línea principal donde se van montando las piezas, y líneas suplementarias que pre-procesan ciertas partes para, posteriormente, alimentar con productos intermedios la línea de montaje principal. El problema añadido que conlleva este tipo de línea es el de sincronizar la tasa de producción de las líneas de alimentación.

C. Tipos de líneas de montaje según el tipo de productos que trabajan

- a. Líneas simples: procesan un único tipo de producto, y las estaciones realizan repetitivamente las mismas tareas.
- b. Líneas mixtas: son líneas en las que se pueden producir variantes de un producto principal. Las operaciones a realizar son muy semejantes para cada una de las variantes. La producción en estas líneas no necesita de tiempos de preparación debido a la igualdad de las operaciones.
- c. Líneas multi-modelo: en esta tipología varían los procesos de producción de un modelo a otro, consecuentemente, se produce por lotes de productos y se tienen en cuenta los tiempos de preparación de cada estación de la línea para cambiar de un producto a otro.

En Ghosh y Gagnon (1989, p.637-670) se agrupan los apartados b y c en un solo apartado (*multi/mixed model*). Esto es ampliamente aceptado hasta el año 1996, cuando Scholl (1996, p.217-244) establece la clasificación anteriormente mostrada.

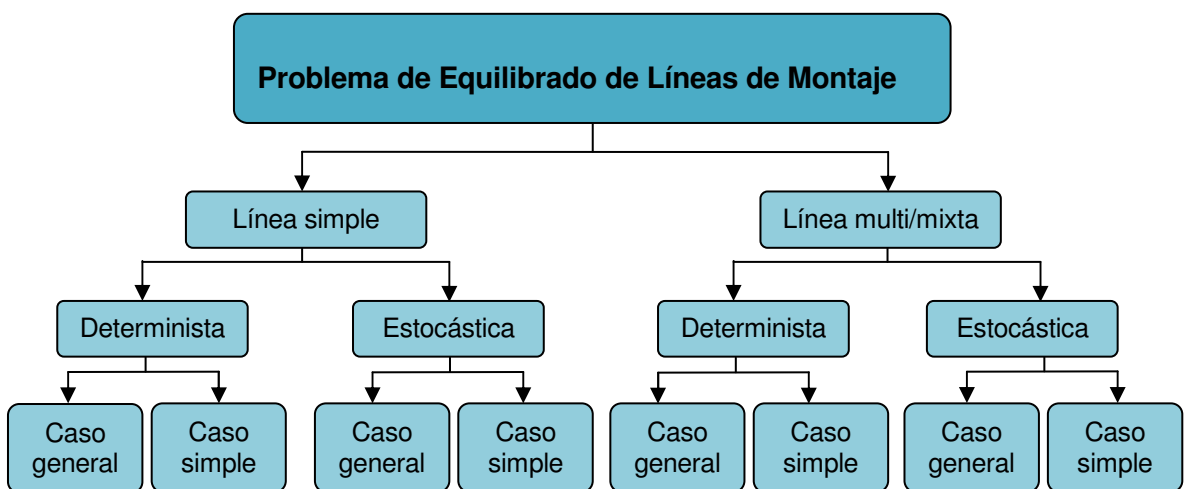
D. Tipos de líneas dependiendo de las duraciones de las tareas

- a. Líneas deterministas: todos los tiempos de proceso de las tareas son conocidos con certeza.



- b. Líneas estocásticas: los tiempos de proceso de las tareas son aleatorios o probabilísticos, debido a que la variabilidad de las duraciones es significativa.
- c. Líneas dependientes: el tiempo de proceso de las tareas depende de alguna variable como pueden ser la estación a la que han sido asignadas, la secuencia o el operador.

En Ghosh y Gagnon (1989, p.637-670), se presenta de forma esquemática una clasificación de las líneas de montaje según los conceptos definidos en los apartados C y D (esquema 3.1).



Esquema 3.1 Clasificación de Ghosh y Gagnon (1989, p.637-640)

E. Tipos de líneas según la disciplina de entrada de las piezas a la línea

- a. Líneas de entrada fija: las piezas llegan a intervalos regulares. Si se da el caso de que la línea es sincrónica este intervalo es el tiempo de ciclo.
- b. Líneas de entrada variable: las piezas llegan a intervalos variables.

F. Tipos de líneas según el tipo de operador de las operaciones

- a. Líneas manuales: los operadores son humanos y los procesos pueden ser automatizados o no.
- b. Líneas robotizadas: los operadores son robots y los procesos son totalmente automatizados.



3.2. Tipos de problemas de líneas

Una vez los problemas de líneas han sido clasificados, a continuación se plantea el tipo de problema a resolver. Éste queda perfectamente definido con el objetivo que se quiere alcanzar en los resultados de la línea y con las restricciones que se deben respetar.

Los problemas más comunes que se encuentran en la literatura se clasifican en dos grupos: los de equilibrado de líneas (ALBP) y los de diseño de líneas (ALDP: *Assembly Line Design Problem*). Además, se pueden encontrar otros tipos de problemas menos comunes como el DLBP (*Disassembly Line Balancing Problem*), donde el objetivo es el equilibrado de líneas de desmontaje, es decir, se desea determinar la asignación de las tareas de desmontaje.

A continuación se explican con mayor profundidad los dos tipos de problema más comunes.

3.2.1. Problemas de equilibrado de líneas de montaje (ALBP)

El objetivo de este tipo de problema es asignar las tareas que son necesarias para el montaje de un producto a las distintas estaciones que se tienen disponibles, que forman la línea de montaje. A este objetivo se le pueden añadir distintas restricciones, como pueden ser las relaciones de precedencia o incompatibilidad, que permiten diferenciar los distintos subtipos de ALBP que existen.

Según el planteamiento clásico del ALBP, se busca encontrar el número mínimo de estaciones, o el tiempo de ciclo mínimo, necesario para procesar las tareas, cumpliendo las relaciones de precedencia.

Por ejemplo, según Baybars (1986, p.909-932), una línea está equilibrada si en plena utilización de sus recursos, los tiempos libres u ociosos en cada estación sumados son mínimos. Baybars establece además que si se consigue que todas las estaciones tengan el mismo tiempo de ejecución, el equilibrio es perfecto. En caso contrario, la tasa de producción de la línea viene determinada por un cuello de botella, que es la estación más lenta.

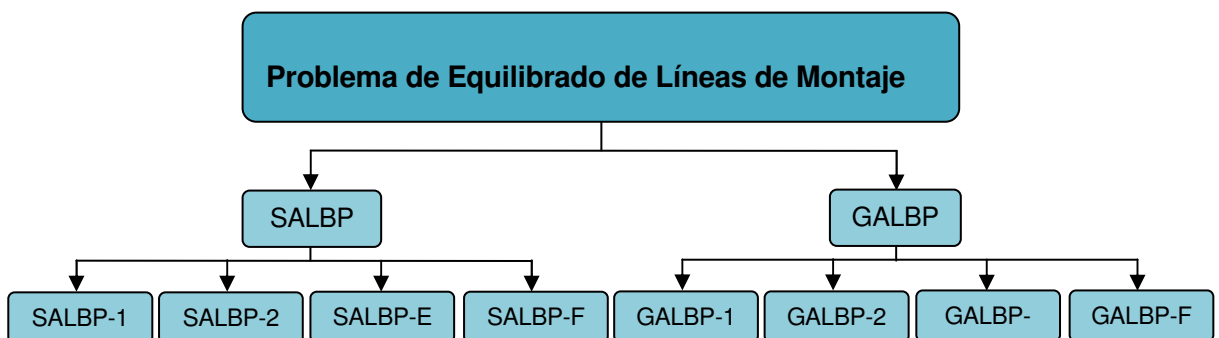
En la literatura se pueden encontrar diversas clasificaciones, las más conocidas y aceptadas históricamente, son las siguientes:

- Baybars (1986, p.909-932) diferencia dos tipos de problemas: el simple (SALBP: *Simply Assembly Line Balancing Problem*) y el general (GALBP: *General Assembly Line Balancing Problem*).



- Ghosh y Gagnon (1989, p.637-670) diferencian cuatro tipos de problemas: el simple determinista, el simple estocástico, el multi/mixto determinista y el multi/mixto estocástico.

A continuación se explican las características de los problemas más importantes. Antes, no obstante, se representa un esquema (Esquema 3.2) que engloba las propuestas de clasificaciones de Baybars (1986, p.909-932), Ghosh y Gagnon (1989, p.637-670), Scholl (1999), Rekiek [et al.] (2002, p.163-174), Graves (1983, p.522-545), Becker y Scholl (2006), Scholl y Becker (2006), Pinto (1983) y Plans (1999).



Esquema 3.2 Tipos de problemas de equilibrado de líneas de montaje

A. SALBP: *Simply Assembly Line Balancing Problem*

Este tipo de problema abarca los problemas de equilibrado más simples, cuyas características son:

- Producción en masa de un único producto homogéneo.
- Cada tarea es indivisible.
- La línea es simple.
- La línea es serial.
- Las tareas solo tienen entre ellas restricciones de precedencia.
- Los datos son conocidos (deterministas) y se dispone de un número de estaciones o tiempo de ciclo fijado (en nuestro caso, número de estaciones).
- La tasa de entrada de piezas a la línea es fija.
- Todas las estaciones están equipadas de forma ecuánime tanto en máquinas como en trabajadores.



A partir de estas características comunes, se pueden distinguir cuatro sub-variantes del SALBP:

- a. SALBP-1: se dispone de un tiempo de ciclo asignado (o tasa de producción) y se busca minimizar el número de estaciones necesarias para realizar el proceso. Este tipo de problema es común cuando la demanda externa puede ser estimada y se quiere instalar un nuevo sistema de montaje.
- b. SALBP-2: en este caso se parte de un número de estaciones fijo y se busca minimizar el tiempo de ciclo de la línea de montaje. Este problema se suele presentar cuando la línea de montaje ya existe.

Una variedad del SALBP-2 es el LB-ALBP, donde, además de minimizar el tiempo de ciclo del sistema, se desea minimizar el tiempo de la segunda estación más cargada, el de la tercera,... y así sucesivamente.

- c. SALBP-E: se busca maximizar la eficiencia de la línea, o, lo que es lo mismo, minimizar el producto entre número de estaciones y el tiempo de ciclo.
- d. SALBP-F: se busca encontrar una solución factible para una combinación cualquiera de tiempo de ciclo y número de estaciones.

B. GALBP: *General Assembly Line Balancing Problem*

Este problema engloba a todos los tipos de ALBP que no son el SALBP. Son ejemplo del GALBP, problemas como: modelos mixtos, estaciones en paralelo, rutas de procedimiento alternativas, tiempos de proceso variables,... Estos problemas se caracterizan por representar problemas más reales que los SALBP.

Dentro de la variedad de los problemas GALBP, destacan los siguientes tipos:

- a. UALBP (*U-line Assembly Line Balancing Problem*): en vez de considerar líneas seriales, se trabaja con líneas de tipo U. Becker y Scholl (2006, p.694-715) establecen que en una línea de tipo U, las estaciones pueden ser colocadas de tal manera que se pueden manejar a la vez dos piezas en diferentes posiciones de la línea. En este caso también se pueden distinguir diferentes variantes: UALBP-1 (minimizar el número de estaciones), UALBP-2 (minimizar el tiempo de ciclo) y UALBP-E (maximizar la eficiencia de la línea).



- b. MALBP (Mixed-Model Assembly Line Balancing Problem): consiste en el equilibrado de líneas con modelos mixtos. Se caracterizan por tener varios modelos de un mismo producto, con un conjunto de tareas comunes para todos los modelos, y sin tener en cuenta tiempos de preparación.
- c. RALBP (Robotic Assembly Line Balancing Problem): consiste en el equilibrado de líneas robotizadas. En este tipo de problema se desea optimizar la ejecución de las tareas en la línea considerando la asignación de las tareas a cada estación y la destinación de los robots a las estaciones.
- d. MOALBP (Multi-Objective Assembly Line Balancing Problem): consiste en el equilibrado con objetivos múltiples. Dichos objetivos pueden ser, por ejemplo: minimizar el número de estaciones, el coste o el número de pulmones y/o maximizar la eficiencia. Rekiek (2002, p.163-174) establece que la mayoría de problemas de equilibrado de líneas de montaje presentan objetivos múltiples.

Después de esta primera explicación, se debe considerar que tanto el SALBP como el GALBP, se pueden subdividir, a su vez, teniendo en cuenta dos nuevos conceptos:

- El tipo de producto que se procesa en la línea: modelo simple y modelo mixto (o múltiple).
- La variabilidad de las duraciones de las tareas: determinista o estocástico.

3.2.2. Problemas de diseño de líneas de montaje (ALDP)

En el ALDP se consideran las posibles características de diseño de la línea, antes que la asignación de las diferentes tareas a las estaciones. Los problemas de diseño de líneas de montaje comprenden otros campos, como pueden ser:

- Especificación de las dimensiones de los espacios de almacenaje y los pulmones o *buffers*.
- Sistemas de transporte.
- Selección del equipo de manufactura para cada operación.

Aun así, siempre se tiene en cuenta igualmente la asignación de tareas y el equilibrado.

Rekiek [et al.] (2002, p163-174) establece que existen diferentes factores que han generado nuevos problemas de diseño. Estos factores son, entre otros, la actualización de



los sistemas de producción caracterizados por una vida útil corta, un alto grado de automatización, la aparición de nuevos equipos de trabajo para llevar a cabo la manufactura, nuevas tecnologías, etc.

3.3. Algoritmos de resolución de problemas de equilibrado de líneas de montaje

Los problemas de optimización combinatoria, como el ALBP, pueden resolverse tanto por métodos exactos (basados en programación matemática, procedimientos de programación dinámica y *branch&bound*, que encuentran solución óptima aunque, habitualmente, para problemas de dimensiones reducidas) como por métodos heurísticos (algoritmos que suelen dar, en un tiempo breve y de forma simple, una buena solución, aunque sin garantizar siempre la óptima).

Tal y como establece Plans (1999), los métodos exactos garantizan una solución óptima (si existe), pero tienen el inconveniente de que, normalmente, únicamente pueden utilizarse con ejemplares con un reducido número de tareas, ya que cuando las dimensiones del ejemplar son elevadas el problema se hace intratable por problemas de tiempo de cálculo y recursos. Este hecho ha provocado la proliferación de métodos heurísticos eficientes para problemas de dimensiones reales.

3.3.1. Algoritmos para la resolución de SALBP's

En la literatura pueden encontrarse diversos procedimientos que resuelven el problema SALBP-1, pero menos que solucionan el SALBP-2. No obstante, diferentes autores proponen que para resolver el SALBP-2 pueden usarse procedimientos que solucionan el SALBP-1 variando el tiempo de ciclo; por lo tanto, la solución es aquella que, con un menor tiempo de ciclo, proporcione una solución factible para el número de estaciones que se tiene.

Una de las mejores aportaciones que existen para conocer el estado del arte del SALBP, es el artículo de Scholl y Becker (2006, p.666-693). En éste se concluye que los métodos disponibles son buenos para resolver problemas de tamaño medio, pero que hacen falta mejoras para que se puedan usar en la resolución de ejemplares de mayores dimensiones. También se concluye que la mayoría de los métodos heurísticos están basados en reglas de prioridad y procedimientos enumerativos, como el algoritmo de Hoffman (1963, p.551-562).



Otro trabajo importante para conocer las últimas investigaciones realizadas hasta la actualidad sobre el problema de equilibrado de líneas de montaje, es el artículo de Boysen, Fliedner y Scholl (2007, p.674-693).

Además de los estudios anteriormente citados, también se pueden encontrar diferentes trabajos que se clasifican, a continuación, en las siguientes tres categorías: programación matemática, trabajos de *branch&bound* y programación dinámica, y heurísticos.

A. Algoritmos exactos. Programación matemática

- a. White (1961, p.274-276): es una de las primeras modelizaciones del problema con Programación Lineal Binaria y constituye un trabajo usado posteriormente como base para muchas otras investigaciones. En este modelo se obliga a no asignar tareas en las últimas estaciones, si antes no se han llenado las anteriores disponibles
- b. Thangavelu y Shetty (1971, p.61-68): estos autores proponen un cambio en los coeficientes de la función objetivo del modelo anterior con la intención de mejorar el funcionamiento del modelo de White (1961, p.272-276).
- c. Valero (1991): resuelve un modelo de programación lineal teniendo en cuenta los recursos consumidos por las tareas, las incompatibilidades entre parejas y considerando diferentes tipos de estaciones de trabajo.
- d. Plans (1999): es un trabajo completo y detallado de síntesis en el que se trata de manera extensa diferentes modelos expuestos por diversos autores.
- e. Pastor y Ferrer (2008): proponen un programa matemático mejorado para la resolución del SALBP.

B. Algoritmos exactos. Branch&bound y programación dinámica

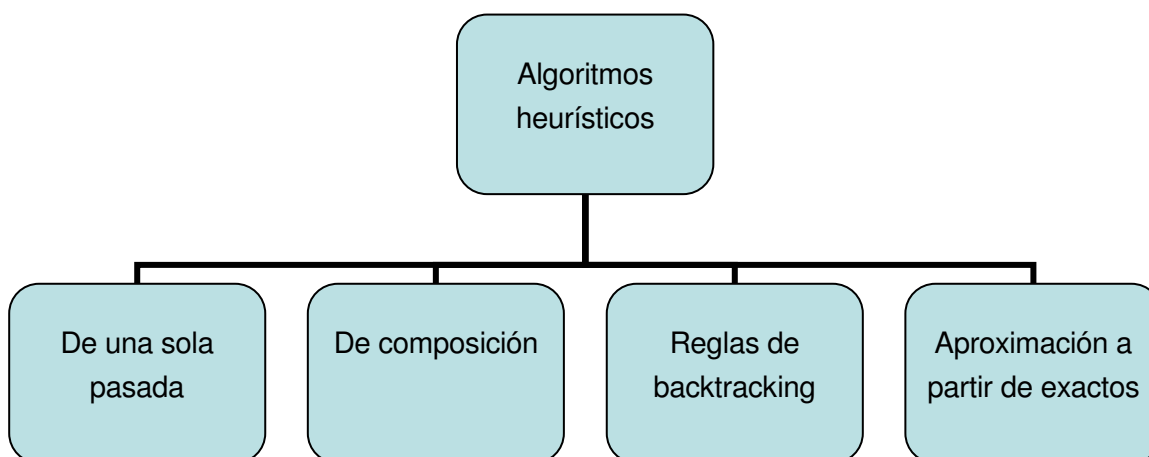
- a. Held, Kart y Shreshian (1963, p.442-459): es la obra de referencia de la programación dinámica aplicada al problema de equilibrado de líneas de montaje.
- b. Talbot y Patterson (1984, p.85-99): proponen un procedimiento *branch&bound* que tiene un funcionamiento correcto para un número de tareas inferior a 50.
- c. Johnson (1988, p.240-253): propone el algoritmo FABLE para la resolución de los problemas de equilibrado de líneas.



- d. Hoffman (1992, p.39-47): propone el algoritmo EUREKA para la resolución de los problemas de equilibrado de líneas.
- e. Klein y Scholl (1997, p.319-334): proponen el algoritmo SALOME, que se basa en un *branch&bound* bidireccional. Existen dos variantes del mismo, el SALOME-1 y el SALOME-2 que sirven para resolver el SALBP-1 y el SALBP-2 respectivamente.

C. Algoritmos heurísticos

Talbot, Patterson y Gehrlein (1986, p.439-454) está considerada como una obra de referencia que trata algoritmos heurísticos que sirven para resolver el SALBP. En esta, se hace una clasificación de los algoritmos heurísticos desarrollados hasta el momento del estudio, cuya representación se puede encontrar en el Esquema 3.3.



Esquema 3.3 Clasificación de algoritmos heurísticos según Talbot, Patterson y Gehrlein

A continuación se comenta esta clasificación:

- a. Heurísticas de una sola pasada: son las que usan reglas de decisión simples, entre las que destacan: Moodie y Young (1965, p.23-29), Tongue (1960, p.21-39) y Helgeson y Birnie (1961, p.394-398).
- b. Heurísticas de composición: se basan en la composición de reglas de decisión. Entre estas heurísticas destaca el algoritmo COMSOAL de Arcus (1966, p.259-278).
- c. Reglas de backtracking (retroceso): entre éstas destaca el algoritmo MALB de Dar-El (1973, p.343-356) y la heurística de Hoffman (1963, p.551-562).



- d. Aproximación partiendo de algoritmos exactos: estas heurísticas parten de un procedimiento exacto al cual se le limita el tiempo de ejecución. En este grupo destacan las de Talbot y Patterson (1984) y el algoritmo de Held, Kart y Shareshian (1963, p.442-459).

Talbot, Patterson y Gherlein (1986, p.430-454) concluyen, además, que los algoritmos con backtracking son los mejores, ya que consiguen obtener soluciones óptimas, o muy próximas al óptimo, de problemas considerablemente grandes con un bajo tiempo de proceso.

Ponnambalalm, Aravindan y Naidu (1999, p.577-586) realizan una evaluación comparativa de distintas heurísticas que resuelven el SALBP-1. En concreto, en este trabajo se analizan heurísticas de una sola pasada y la heurística de Hoffman. Finalmente, se comparan los resultados obtenidos según el número de estaciones usado, la eficiencia de la línea, la homogeneidad de la carga de trabajo en las estaciones y el tiempo de ejecución.

También destaca por su aportación de síntesis el artículo realizado por Dar-El (1975, p.302-310). En este, se lleva a cabo una comparación detallada de diez algoritmos heurísticos de una sola pasada.

A su vez, es importante remarcar el trabajo de Hackman, Magazine y Wee (1989, p.916-923). Consiste en un algoritmo *branch&bound* al cual se le aplica una heurística para reducir la dimensión del árbol de búsqueda. Este método resulta útil tanto para la resolución del SALBP-1, como para la resolución del SALBP-2.

En relación con la resolución de problemas tipo SALBP-2, Klein y Scholl (1996, p.367-385) concluyen que la mayoría de investigadores usan la repetición de los métodos usados para SALBP-1.

Desde otra vertiente, también se han propuesto metaheurísticas basadas en, entre otros:

- Algoritmos genéticos: Rubinovitz y Levitin (1995, p.343-354) y Gonçalves y Almeida (2002, p.629-642)
- Recocido simulado: Suresh y Sahu (1994, p.1801-1810).
- Búsqueda tabú: Pastor [et al.] (2002, p. 1317-1323).
- Colonias de hormigas: Baykasoglu [et al.] (2003).



3.3.2. Algoritmos para la resolución de GALBP's

Una de las mejores obras para obtener información actualizada sobre los algoritmos de resolución de GALBP's es el trabajo de Becker y Scholl (2006, p.694-715).

Otra aportación a destacar es la de Erel y Sarin (1998, p.414-434), donde se encuentra un resumen completo de los procedimientos que se han usado para resolver problemas de equilibrado de líneas de montaje.

Los trabajos que resuelven GALBP's son muy variados y con una metodología distinta dependiendo de las características del problema concreto.

En cuanto a procedimientos heurísticos y/o metaheurísticos se pueden nombrar, entre otros: Gen (1996, p.631-634), Voss (1994, p.183-194), Pastor [et al.] (2002, p.1317-1323), Vilarinho y Simaria (2002, p.1405-1420), Erel, Sabuncuoglu y Aksu (2001, 3003-3015), Arcus (1966, p.259-277), Buxey (1974, p.1010-1021), Bautista y Pereira (2003), Ponnambalam (2003, p.669-690), Ruvimobitz y Leivitin (1995, p.343-354), Kim, Kim y Kim (1996, p.397-409), Park, Park y Kim (1997, p.321-332), Bhattatchajee y Sahu (1988, p.67-77), Spina (2003, p.573-583), Erel y Gokcen (1999, p.194-204) y Phonganant [et al.] (2001).

También se ha resuelto GALBP's mediante métodos exactos, como en: Urban (1998, p.738-741), Johnson (1983, p.1309-1324), Sarin, Erel y Dar-El (1999, p.525-535), Scholl y Klein (1999, p.721-736), Nicosia, Pacciarelli y Pacifici (2002, p.99-113), Miltenburg (1998, p.1-23) y Miltenburg y Wijngaard (1994).

Independientemente de la técnica de resolución utilizada, a continuación se destacan los siguientes trabajos por las diferentes características que tienen en cuenta de las líneas de montaje: Agraval (1985, p.403-421), donde se considera la cercanía física de las estaciones una vez realizada la distribución de tareas; Miyazaki y Ohta (1987, p.681-690), donde además de minimizar el número de estaciones, se minimiza la suma de las duraciones de las tareas de la estación más cargada de entre las soluciones iguales; Sarker y Harris (1988), donde se analizan las aplicaciones del *Just In Time*; Dar-El y Mazer (1989, p.2105-2116), en el que se introduce la problemática del stock intermedio; McClain y Moodie (1991), donde se expone la importancia de los stocks intermedios del inicio de la línea frente a los situados en las últimas estaciones; y Bautista y Pereira (2006, p.2016-2032), donde se tienen en cuenta restricciones de tiempo y espacio en la línea de producción.



3.3.3. Algoritmos para la resolución de ALDP's

Rekiek [et al.] (2002, p.163-174) es la obra de referencia para la resolución del ALDP. En ésta, se realiza un amplio estudio del estado del arte sobre los diferentes métodos existentes para el diseño de líneas de montaje. Finalmente, se concluye que la herramienta principal para la resolución de este tipo de problemas está basada en procedimientos *branch&bound*.

Otras aportaciones importantes son: He y Kusiak (1997), Rubinovitz y Bukchin (1993, p.497-500), Bukchin y Tzur (2000, p.585-598) y Pinnoi y Wilhem (1998).

Existe un número reducido de artículos que tratan problemáticas muy concretas, como son: líneas en U en Ajenblit y Wainwright (2003); estaciones en paralelo Bard (1989, p.721-736); líneas mixtas en Ponnambalam, Aravindan y Subba Rao (2003, p.669-690); problemas con estaciones diferentes en Nicosia, Pacciarelli y Pacifici (2002, p.99-113); líneas paralelas en Suer (1998, p.467-470).

En la literatura se plantean también problemas que destacan por sus restricciones, entre las que cabe remarcar: restricciones de capacidad en Moon [et al.] (2002, p.605-621); tiempos de proceso estocásticos en Sarin, Erel y Dar-El (1999, p.525-535); restricciones de incompatibilidad en Park, Park y Kim (1997, p.321-332); tiempos de proceso dependientes del tipo de operario en Corominas, Pastor y Plans (2008, p.1126-1132); datos de entrada no conocidos con seguridad o no disponibles, como tiempos de proceso difusos en Gen, Tsujimura y Li (1996, p.631-634); tiempos de proceso dependientes de la secuencia de productos en Spina, Galantucci y Dassisti (2003, p.573-583); líneas de montaje multi-productos en Pastor [et al.] (2002, p.1317-1323); alternativas de proceso en Pinto, Dannenbring y Khumawala (1983); líneas de montaje multi-objetivos en Kim, Kim y Kim (1996, p.397-409); alternativas de equipos de manufactura en Bukchin y Tzur (2000, p.585-598); problemas que ubican la carga de las estaciones en una cierta ventana de tiempo en Pastor y Corominas (2000, p.23-45).

3.3.4. Principales funciones objetivo para el problema de equilibrado de líneas de montaje

La fuente información con la que se ha desarrollado este apartado es el trabajo de Plans (1999).

La función objetivo para los SALBP-1 consiste en minimizar el número de estaciones de trabajo dado un tiempo de ciclo:



$$[MIN]z = \sum_{j=1}^{m \max} y_j$$

donde :

$$y_j \in \{0,1\} \forall j; \quad 1 \text{ si existe la estación } j$$

$m \max$: número máximo de estaciones de trabajo

En referencia a los SALBP-2, se desea minimizar el valor del tiempo de ciclo para un número de estaciones fijado:

$$[MIN]z = T$$

donde T es el tiempo de ciclo

Si existen estaciones en paralelo, esta característica se modeliza con una variación en la función objetivo del SALBP-1:

$$[MIN]z = \sum_{j=1}^{m \max} (y_j + q_j)$$

donde :

$$y_j \in \{0,1\} \forall j; \quad 1 \text{ si existe la estación } j$$

$m \max$: número máximo de estaciones de trabajo

$$q_j \in \{0,1\} \forall j; \quad 1 \text{ si a } j \text{ se le añade una estación en paralelo}$$

Tal y como se ha introducido anteriormente, la función objetivo de White (1961, p.274-276), se caracteriza por ser una de las primeras efectuadas para la programación lineal binaria:

$$[MIN]z = \sum_{i \in F} \sum_{j=m \min+1}^{m \max} c_j \cdot x_{i,j}$$

donde :

$$c_{j+1} \geq M \cdot c_j \quad j = m \min+1, \dots, m \max-1$$

$$x_{i,j} \in \{0,1\} \forall i, j$$

F es el conjunto de elementos sin sucesores, i se refiere a las tareas y j a las estaciones, $m \min$ y $m \max$ son el número mínimo/máximo de estaciones de trabajo, c_j es el coste asociado de asignar una tarea a la estación j , $x_{i,j}$ vale uno si se asigna la tarea i a la estación j y M es un valor suficientemente grande para garantizar que se asignan las tareas a la primera estación posible.

Posteriormente, Thangavelu Shetty (1971, p.61-68) proponen un cambio en el coeficiente c_j de White para mejorar las soluciones:



$$c_{i,j} = t_i \cdot \left[\sum_{k \in F} t_{k+1} \right]^{j=m \min+1} \quad j = m \min+1, \dots, m \max \quad i \in F$$

$$c_{i,j} = 0 \quad \text{contrariamente}$$

donde t_i es el tiempo de proceso de la tarea i . Consecuentemente la nueva función objetivo queda de la siguiente manera, donde el coeficiente $c_{i,j}$ representa el coste de asignar la tarea i a la estación j :

$$[MIN]z = \sum_{i \in F} \sum_{j=m \min+1}^{m \max} c_{i,j} \cdot x_{i,j}$$

Otra posible función objetivo es la característica del SALBP-E, en donde se minimiza el producto del número de estaciones por el tiempo de ciclo:

$$[MIN]z = m \cdot T$$

donde m es el número de estaciones de trabajo.

Siguiendo con la explicación que da Plans (1999), históricamente también se ha deseado minimizar el tiempo de ocio de la línea o el tiempo de ocio relativo al número de estaciones:

$$[MIN]z = m \cdot T - \sum_{i=1}^n t_i$$

Una aportación a la minimización del tiempo de ocio de la línea (I) se encuentra en Bennet y Bird (1976, p.195-201). Siendo w un valor muy grande, T el tiempo de ciclo del sistema y t_j la carga de la estación j :

$$[MIN]z = w \cdot \sum_{j=1}^m I_j + \sum_{j=1}^m I_j^2$$

donde :

$$I_j = T - t_j$$

Según Plans (1999), también se ha buscado la minimización del *flow-time*, que es el tiempo que tarda el producto desde que es lanzado a la línea hasta que sale terminado:

$$[MIN]z = F$$



Así como la discrepancia (diferencia entre la suma de las duraciones de las tareas asignadas a las estaciones y un tiempo de ciclo ideal T), del coste de la línea (minimización de estaciones) y del coste unitario (minimización del coste por unidad de producto).

En referencia a minimizar las discrepancias, se puede destacar la aportación de Pastor (1994), que minimiza las discrepancias con relación a un tiempo de ciclo ideal:

$$[MIN]z = \sum_{j=1}^{m \max} (\partial_j^- + \partial_j^+)$$

donde ∂_j^- es la diferencia por exceso entre la suma de duraciones de las tareas asignadas a la estación j y el tiempo de ciclo ideal T , y ∂_j^+ es la diferencia por defecto .

En Plans (1999) también aparece una función objetivo que hace referencia a las líneas de montaje, la de minimización de operarios inexpertos:

$$[MIN]z = \sum_{j=1}^{m \max} r_j$$

donde r_j vale 1 si el operario asignado a la estación j es inexperto.

En el modelo de Miyazaki y Ohta (1987, p.681-690) se pretende minimizar el número de estaciones y el tiempo de ciclo a la misma vez, resultando la siguiente función objetivo:

$$[MIN]z = \sum_{j=1}^{m \max} y_j + \frac{T}{T_{\max}}$$

donde y_j son las variables de existencia de las estaciones y T es el tiempo de ciclo a minimizar. De esta forma, entre dos asignaciones posibles de tareas a una estación, se selecciona la que implica un tiempo de ciclo menor.

Otra función objetivo se puede encontrar en Scholl (1999):

$$[MIN]z = \sum_{j=1}^m (\bar{t} - t_j)^2$$

donde m es el número de estaciones, t_j la carga de la estación j y \bar{t} el tiempo medio de carga de las estaciones. Esta función busca la homogeneización de la distribución de la carga de trabajo.



Moodie y Young (1965, p.23-29) también buscan homogeneizar la carga de trabajo entre las estaciones con su función objetivo *smoothness index* o índice de homogeneidad:

$$[MIN]z = \sqrt{\sum_{j=1}^m (T - t_j)^2}$$

donde T es el tiempo máximo de todas las estaciones de trabajo (tiempo de ciclo) y t_j la carga de la estación j .





4. Descripción del *Lexicographic Bottleneck Assembly Line Balancing Problem* (LB-ALBP)

En este apartado se describe en primer lugar el objetivo y las características del *Lexicographic Bottleneck Assembly Line Balancing Problem* (LB-ALBP). Posteriormente, se analizan las principales investigaciones que se han realizado hasta la actualidad en relación al LB-ALBP.

4.1. Definición y características principales del LB-ALBP

El problema de equilibrado de líneas de montaje ha presentado diferentes funciones objetivo a lo largo de la historia. Cada una de estas variantes ha derivado en unas modelizaciones y procesos de resolución concretos.

En este proyecto se trata el problema LB-ALBP, que se puede considerar como una derivación del problema SALBP-2. El objetivo del LB-ALBP es minimizar el tiempo de trabajo de la estación más cargada (tiempo de ciclo), posteriormente minimizar el tiempo de la segunda estación más cargada, el de la tercera estación más cargada,... y así sucesivamente. Es decir, se minimiza la carga de trabajo de todas las estaciones de forma jerárquica

En la literatura existen diversos procedimientos que intentan optimizar la primera parte de la función objetivo del LB-ALBP (minimizar el tiempo de ciclo), pero no tienen en cuenta la minimización jerárquica del tiempo del resto de estaciones.

La importancia del LB-ALBP se puede contemplar con el análisis del siguiente gráfico (Gráfico 4.1), donde se representa la distribución de la carga de trabajo entre todas las estaciones para dos soluciones de un mismo problema.

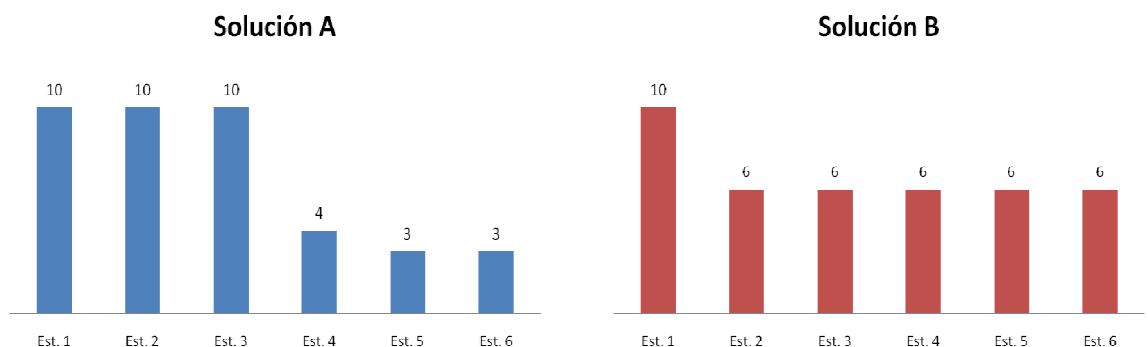


Gráfico 4.1 Dos posibles distribuciones de carga de trabajo



- Solución A: 10, 10, 10, 4, 3, 3.
- Solución B: 10, 6, 6, 6, 6, 6.

Comparando las dos soluciones se observa que el tiempo de ciclo del sistema es el mismo para las dos (10), pero la solución B es más fiable que la solución A.

Con el fin de comparar la fiabilidad de las dos soluciones se debe observar que en A existen 3 estaciones que representan un cuello de botella, mientras que en B únicamente existe 1. Cualquier problema en estas estaciones críticas, provocaría que una unidad no fuera completada o hubiera retrasos, con lo que la productividad se vería reducida. Por ejemplo, sea ω la probabilidad de que una estación cuya carga de trabajo es igual a T sea mayor que T (siendo T el tiempo de ciclo). Para la solución A, la probabilidad de que alguna estación tenga una carga de trabajo mayor a T es igual a $3\omega - 3\omega^2 + \omega^3 (1 - (1 - \omega^3))$. En cambio, para la solución B, la probabilidad de que una estación tenga una carga de trabajo mayor que T es igual a ω . Consecuentemente, se considera más fiable la solución B que la A.

Otra ventaja que aporta la aplicación del LB-ALBP es que, a pesar de no ser su objetivo, tiende a una homogeneización de la distribución de la carga de trabajo entre las distintas estaciones que forman el sistema, tal y como también consiguen las funciones de Bird (1976, p.195-201), Pastor (1994), Scholl (1999) y Moodie y Young (1965, p.23-29), todas ellas analizadas en el apartado 3.3.4. Además, Boysen, Fließner y Scholl (2006) afirman que la homogeneización de la carga evita los problemas de calidad derivados de la existencia de estaciones con cargas de trabajo excesivas.

4.2. Estado del arte del LB-ALBP

Las principales investigaciones sobre el LB-ALBP han sido realizadas por el Dr. Rafael Pastor en la *Universitat Politècnica de Catalunya*. En el trabajo de Pastor (2004) se describe por primera vez el LB-ALBP, y posteriormente, en el artículo Pastor (2007), se propone una función para la medida de la calidad de las soluciones, dos modelos de programación lineal mixta para la resolución exacta del problema, y tres procedimientos heurísticos por aproximación. Todas estas aportaciones se exponen con mayor detalle en este punto. Destaca también la aportación de Sentinella y Solsona (2006), que realizan un primer estudio para la resolución del LB-ALBP.



4.2.1. Medida de la calidad de una solución

Con el fin de poder comparar una solución Sol_1 y una solución Sol_2 del LB-ALBP, Pastor (2007) define el parámetro δ . El uso de este parámetro requiere en primer lugar, ordenar de mayor a menor todas las estaciones de cada solución según las cargas de trabajo. De esta forma, en la posición 1 se tendrá la estación más cargada, y en la posición m (siendo m el número de estaciones del problema a resolver) la estación menos cargada. El parámetro δ viene definido por la siguiente ecuación:

$$\delta = \frac{\Delta_1 \cdot \beta^m + \Delta_2 \cdot \beta^{m-1} + \dots + \Delta_m \cdot \beta}{T_{\delta^*} \cdot \beta^{m-1}}$$

Ecuación 4.1

donde el término Δ_j define la diferencia positiva, negativa o nula entre las j -ésimas estaciones más cargadas de las dos soluciones que se están comparando ($Sol_1 - Sol_2$); T_{δ^*} es el tiempo de ciclo de la mejor de las dos soluciones comparadas (solución de referencia); y β es un parámetro cuyo valor debe garantizar el objetivo jerarquizado, es decir, que se penalice más la diferencia de carga de trabajo entre las estaciones más cargadas que entre las estaciones menos cargadas. Siguiendo el trabajo de Pastor (2007), en este proyecto el valor de β también se fija en 100.

δ es un parámetro de comparación que sirve para cuantificar cuánto peor es una solución comparada con otra. Consecuentemente, cuanto mayor sea el valor de δ , peor será la solución que se está comparando con la de referencia.

Para comprender mejor el funcionamiento de esta función de medida de calidad se plantea el siguiente ejemplo. Considérense las siguientes distribuciones de carga para un mismo problema:

- Sol_1 : 50-48-46-44
- Sol_2 : 50-47-46-45

se puede decir que Sol_2 es mejor que Sol_1 porque, a pesar de que el tiempo de ciclo es el mismo (50), Sol_2 tiene una menor carga de trabajo en la segunda estación más cargada (47 respecto a 48). Consecuentemente, se dice que Sol_2 asume el papel de referencia, δ^* , y se comparará Sol_1 con ella. El valor δ asociado a este ejemplo es:

$$\delta = \frac{(50 - 50) \cdot 100^4 + (48 - 47) \cdot 100^3 + (46 - 46) \cdot 100^2 + (44 - 45) \cdot 100}{50 \cdot 100^3} = 0.02$$



4.2.2. Algoritmos para la resolución del LB-ALBP

En el trabajo de Pastor (2007), que amplía el trabajo de Sentinella y Solsona (2005), se diseñan dos modelos matemáticos para la resolución exacta del LB-ALBP. Posteriormente, en el mismo artículo, y partiendo de estos dos modelos exactos, se desarrollan tres heurísticas por aproximación. A continuación se introduce brevemente cada uno de estos algoritmos.

A. Algoritmos exactos (programación lineal mixta)

- a. Modelo Jerarquizado Global (MJG): este modelo soluciona el LB-ALBP minimizando una suma de funciones con pesos suficientemente diferenciados como para garantizar el objetivo jerárquico del problema. La función objetivo de Modelo Jerarquizado Global es:

$$[MIN]z = \sum_{k=1}^m \alpha_k \cdot T_k$$

donde el parámetro α_k garantiza la jerarquización de la solución, es decir, se tendrá un valor α_k mucho mayor para la estación más cargada que para la segunda más cargada, y así sucesivamente; y T_k es el tiempo de la estación k .

- b. Modelo Jerarquizado Sucesivo (MJS): consiste en resolver $m-1$ veces un programa lineal mixto, siendo m el número de estaciones del problema a resolver. El funcionamiento consiste en: aplicar el modelo para minimizar el tiempo de la estación más cargada (submodelo 1), posteriormente, volver a aplicar el método para minimizar el tiempo de la segunda estación más cargada (submodelo 2), y así sucesivamente hasta minimizar el tiempo la penúltima estación más cargada (submodelo $m-1$).

B. Algoritmos heurísticos

- a. Heurística derivada del MJG: consiste en ejecutar el MJG con un tiempo de ejecución límite de 18.000 segundos, y conservar la mejor solución obtenida.
- b. Heurística 1 derivada del MJS: consiste en asignar el tiempo de ejecución disponible al submodelo en curso una vez todos los submodelos previos han sido solucionados de forma óptima. En el trabajo de Pastor (2007) se ejecuta el primer submodelo con un tiempo de ejecución límite de 18.000



segundos; si al finalizar el primer submodelo hay tiempo disponible, se soluciona el segundo submodelo con este tiempo; y así sucesivamente.

- c. Heurística 2 derivada del MJS: consiste en asignar la mitad del tiempo de ejecución disponible al submodelo en curso. En el trabajo de Pastor (2007) se ejecuta el primer submodelo con un tiempo de ejecución límite de 9.000 segundos; a continuación se ejecuta el segundo submodelo con la mitad del tiempo de ejecución disponible, que será como mínimo de 4.500 segundos; y así sucesivamente.

En este proyecto, se diseñan diez nuevos procedimientos heurísticos para la resolución del LB-ALBP. Todos estos procedimientos se especifican en detalle en el punto 5.3. Para poder comprobar la eficacia y eficiencia de los diferentes métodos de resolución diseñados, se resuelven problemas ya predefinidos en la literatura. Las especificaciones concretas de cada uno de estos 16 problemas (precedencias, duraciones de las tareas, etc.), así como diferentes números de estaciones, lo que representa una base inicial de 274 ejemplares a resolver, se obtiene de la página web: <http://www.assembly-line-balancing.de/>, del Dr. Scholl, que reúne un conjunto de problemas test utilizados por la comunidad científica internacional para probar la calidad de sus trabajos y estudios sobre el ALBP.





5. Resolución del LB-ALBP

En este trabajo, el LB-ALBP se resuelve mediante heurísticas. Todos estos procedimientos se basan en un algoritmo “principal” que parte de unas heurísticas base existentes en la literatura para el SALBP-2. En total se desarrollan 10 nuevas heurísticas para el LB-ALBP que son el resultado de la combinación de dos variantes de la heurística principal con seis métodos de optimización local. Estos métodos son aplicados a las soluciones de cada una de las heurísticas base.

Todos los procedimientos se han programado usando el software Visual Basic 2005 Express Edition en su modo de consola. Para el análisis y el tratamiento de los resultados se utiliza el programa Microsoft Office Excel 2007.

En referencia a la tecnología utilizada, todo el experimento computacional se ha realizado en un ordenador personal con procesador Intel® Pentium® IV CPU 3.40GHz 512MB RAM.

A continuación se exponen los procedimientos heurísticos diseñados así como los resultados alcanzados.

5.1. Procedimientos heurísticos para la resolución del LB-ALBP

El objetivo de este apartado es: detallar la heurística “principal” y sus dos variantes; posteriormente, definir seis métodos de optimización local de soluciones; y, finalmente, exponer las diez heurísticas desarrolladas para la resolución del LB-ALBP.

5.1.1. Heurística “principal” para la resolución del LB-ALBP

5.1.1.1. Definición de la heurística “principal” para la resolución del LB-ALBP

A continuación se presenta, paso a paso, la heurística “principal” a partir de la cual se desarrollan las heurísticas planteadas en este proyecto.

- Paso 1: ejecutar un total de 14 heurísticas ya existentes en la literatura para el SALBP-2. Estas heurísticas son definidas en este trabajo como heurísticas base y se explican con detalle en el apartado 5.1.1.3.
- Paso 2: valorar las 14 soluciones obtenidas mediante el parámetro δ y quedarse con la mejor.



- **Paso 3:** fijar la estación más cargada y dividir el problema inicial en dos sub-problemas independientes, tal y como se puede apreciar en el gráfico 5.1.

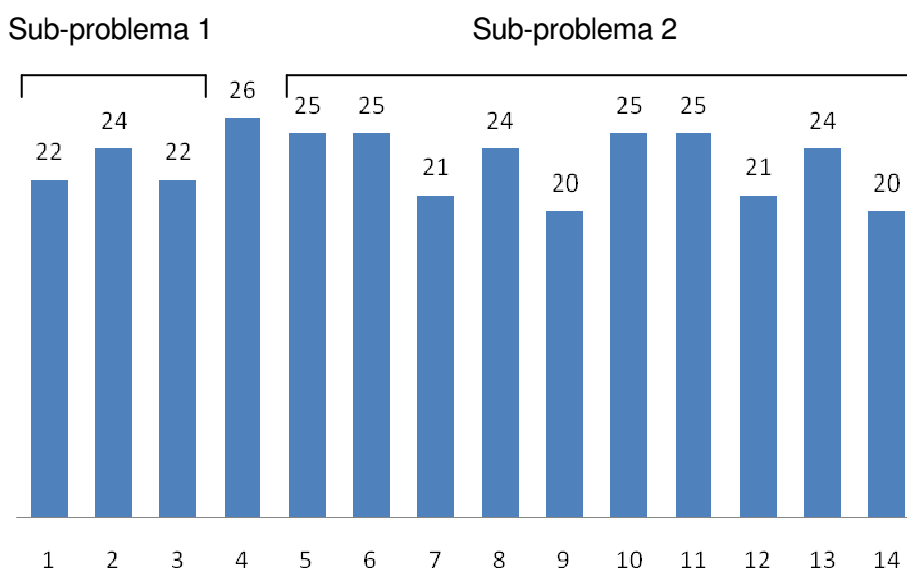


Gráfico 5.1 Ejemplo de distribución de carga para la explicación de la heurística principal

La estación más cargada es la 4, con 26 unidades de tiempo. De esta forma, las tareas que forman parte de esta estación son fijadas, y el problema inicial se subdivide en los dos sub-problemas que forman las tareas asignadas al grupo de estaciones 1-2-3, y las tareas asignadas a las estaciones 5-6-7-8-9-10-11-12-13-14, respectivamente.

- **Paso 4:** si quedan sub-problemas por resolver, volver al paso 1 ejecutando de nuevo las 14 heurísticas base para solucionar los dos nuevos sub-problemas de forma independiente.

5.1.1.2. Dos opciones de desarrollo de la heurística “principal”

Entre las diez heurísticas desarrolladas en este proyecto (apartado 5.1.3) para la resolución del LB-ALBP, hay un grupo en el que se aplica la heurística “principal” siguiendo el método definido en el apartado 5.1.1.1, y otro grupo en el que se aplica una modificación para tratar de mejorar las soluciones obtenidas.

La variante que se realiza en la heurística “principal” se puede observar a continuación en el paso 4:



- Paso 1: lanzar un total de 14 heurísticas ya existentes en la literatura llamadas heurísticas base (definidas en el punto 5.1.3) que solucionan el SALBP-2.
- Paso 2: usando la herramienta del parámetro δ quedarse con la mejor solución de todas las heurísticas base lanzadas.
- Paso 3: fijar la estación más cargada y dividir el problema inicial en dos sub-problemas independientes.
- Paso 4: si quedan sub-problemas por resolver, volver al paso 1 lanzando las 14 heurísticas base para solucionar los dos nuevos sub-problemas de forma independiente. Si al solucionar los dos sub-problemas se obtiene una solución global que mejora la mejor solución que se tenía hasta el momento, se vuelve al paso 1 reiniciando la aplicación de la heurística principal (considerando el problema global de nuevo, sin ninguna estación fijada). Con esto se pretende generar un espacio de búsqueda diferente.

Por ejemplo, considérese la distribución de carga del gráfico 5.1. Si al resolver de forma independiente el sub-problema 1 se pasa de la distribución inicial 22-24-22, a una distribución igual a 22-23-23, la nueva asignación de tareas será mejor que la anteriormente obtenida (se reduce en una unidad el tiempo de la estación más cargada). Al mejorarse la solución, se vuelve a iniciar la heurística "principal" (sin considerar ninguna estación fijada) partiendo de la solución mejorada.

De ahora en adelante se definen dos términos para distinguir las dos versiones posibles de la heurística "principal":

- Heurística principal 1: es la versión original de la heurística principal (apartado 5.1.1.1).
- Heurística principal 2: es la versión modificada de la heurística principal (apartado 5.1.1.2).

5.1.1.3. Heurísticas base

Tal y como se ha introducido anteriormente, la heurística "principal" se basa en la ejecución reiterada de unas heurísticas ya existentes en la literatura, y que han sido denominadas heurísticas base. Estas heurísticas han sido seleccionadas teniendo como referencia el artículo de Ponnambalam, Avarindam y Mogileeswar (1999, p.577-586), donde se analizan los distintos procedimientos heurísticos existentes en la literatura para la resolución del



SALBP-1 Debe recordarse que los procedimientos para el SALBP-2 son escasos y, en general, malos y largos.

Como se ha mencionado, el LB-ALBP tiene un objetivo semejante al SALBP-2 y así, para que las heurísticas que solucionan el SALBP-1 resuelvan el SALBP-2, y, consecuentemente, el LB-ALBP, se aplica el siguiente procedimiento:

- Paso 1: Cálculo de una cota inferior del tiempo de ciclo, ci :

$$ci = \max \left\{ t \max, \frac{tsum}{m} \right\}$$

Donde ci es el valor de la cota inferior del tiempo de ciclo, $t \max$ el máximo tiempo de proceso de todas las tareas que forman el ejemplar, $tsum$ la suma de todos los tiempos de proceso de las tareas y m el número de estaciones.

- Paso 2: Aplicar la heurística que soluciona el SALBP-1 considerando un tiempo de ciclo igual a ci y comprobar si se obtiene una solución factible, es decir, si todas las tareas han podido ser asignadas a las m estaciones disponibles.
 - 2A: si la solución es factible, fin del problema.
 - 2B: si la solución no es factible se incrementa el tiempo de ciclo en una unidad y se vuelve al paso 2. El incremento del tiempo de ciclo en una unidad es posible ya que, como se realiza en la literatura, todos los tiempos de proceso de las tareas son considerados números enteros.

A continuación se detallan las 14 heurísticas base.

5.1.1.3.1 Heurísticas de una sola pasada

En este trabajo, trece de las heurísticas base definidas son de una sola pasada. Las reglas de paso que las definen se especifican en la tabla 5.1. Para cada una de estas reglas se desarrolla una heurística que, brevemente, consta de los siguientes pasos:

- Paso 1: ordenar las tareas según el valor que tienen aplicando la regla de paso.
- Paso 2: actualizar el conjunto de tareas candidatas (aquellas cuyas antecesoras hayan sido ya asignadas y caben en la estación que está siendo completada). Al ser un proceso orientado a estaciones, si no hay ninguna tarea, abrir nueva estación y volver al paso 2; sino ir al paso 3.



- **Paso 3:** entre las tareas candidatas, asignar la tarea con mayor prioridad en la primera estación disponible siempre y cuando no se viole la capacidad de cada estación ni las relaciones de precedencia.
- **Paso 4:** repetir los pasos 2 y 3 hasta que se hayan asignado todas las tareas.

Número	Regla de paso	Fórmula para determinar la prioridad de la tarea
1	Máximo peso posicional	$PP_i = t_i + \sum_{j \in S_i} t_j$
2	Máximo tiempo de proceso de las tareas	t_i
3	Mínimo número de tareas predecesoras	NP_i
4	Máximo número de tareas sucesoras	NS_i
5	Máximo número de tareas sucesoras inmediatas	NSI_i
6	Mínima cota inferior	$LB_i = \frac{t_i + \sum_{j \in P_i} t_j}{T}$
7	Mínima cota superior	$UB_i = n + 1 - \frac{t_i + \sum_{j \in S_i} t_j}{T}$
8	Mínimo Slack time	$UB_i - LB_i$
9	Mínimo número de tarea	Número de la tarea i
10	Máximo peso posicional medio	$\frac{PP_i}{NS_i + 1}$
11	Mínima cota superior dividido por el número total de sucesores	$\frac{UB_i}{NS_i + 1}$
12	Máxima tiempo de tarea dividido por el límite superior	$\frac{t_i}{UB_i}$
13	Máximo de seguidores dividido por task slack	$\frac{NS_i}{UB_i - LB_i}$

Tabla 5.1 Reglas de paso de las heurísticas base de una sola pasada

Donde T es el tiempo de ciclo, n el número de tareas que deben ser asignadas a las estaciones, NS_i/NP_i el número de sucesoras/predecesoras de una tarea i, NSI_i el número de sucesoras inmediatas de una tarea i, S_i el conjunto de tareas sucesoras de la tarea i, t_i el tiempo de proceso de la tarea i y PP_i el peso posicional de la tarea i.



La regla de paso número 1, correspondiente al máximo peso posicional, fue definida por primera vez en Helgeson y Birnie (1961, p.394-398). Las reglas de paso número 2 y 3 fueron definidas por Moodie y Young (1965, p.23-29) y Kilbridge y Wester (1961, p.292-298) respectivamente. El resto de reglas de paso, es decir, de la número 4 a la número 13, son el resultado de una agrupación de otras heurísticas de una sola pasada realizada por Hackman et al. (1989, p.916-924).

5.1.1.3.2 Heurística de Hoffmann

Hoffmann (1963, p.551-562) propone un algoritmo para el equilibrado de líneas de montaje utilizando una matriz de precedencias y la enumeración de posibles soluciones.

En el algoritmo de Hoffman se usa una matriz de precedencias de ceros y unos y un vector para implementar el proceso de enumeración. Dentro de la matriz, aparece un uno en la fila i , columna j , si la tarea i es una predecesora inmediata de la tarea j .

Con el fin de utilizar esta matriz que permite generar diferentes asignaciones factibles de tareas a estaciones, cada columna de la matriz se suma, y estas sumas forman un vector denominado *code number* que se añade a la matriz creando una nueva fila. Posteriormente, a la diagonal de la matriz se le asigna un valor arbitrario D .

El primer *code number* es denominado K_1 y consta de n enteros (siendo n el número de tareas a asignar), al menos uno de los cuales es igual a cero. Las tareas cuya columna tenga un cero en la fila del *code number*, son candidatas para ser asignadas.

El esquema para generar las combinaciones tareas-estación factibles y equilibrar la línea es el siguiente:

- Paso 1: Buscar de izquierda a derecha un 0 en el *code number*.
- Paso 2: seleccionar la tarea (la posición en la que se localiza el 0 marca la tarea que se debe seleccionar).
- Paso 3: sustraer del tiempo restante de la estación a la que se está asignando el tiempo de la tarea seleccionada.
- Paso 4: si el resultado es positivo ir al paso 5.
- Paso 4A: si el resultado es negativo ir al paso 6
- Paso 5: eliminar de la matriz de precedencias la fila que representa a la tarea que se ha asignado (llenar de ceros) y actualizar el *code number*. Ir al paso 6.



- Paso 6: volver al paso 1 y empezar la búsqueda de un elemento a la derecha del que se había seleccionado previamente y repetir los pasos del uno al seis hasta que se hayan examinado todas las columnas. Ir al paso 7.
- Paso 7: substraer del tiempo restante de la estación el tiempo de la mejor combinación factible hallada (si es la primera combinación de la estación, substraer del tiempo de ciclo).
- Paso 8: si el resultado de la resta es cero o negativo ir al paso 9.
- Paso 8A: si es positivo, el conjunto de tareas que forma la combinación se asigna definitivamente a la estación en la que se está trabajando.
- Paso 9: volver un *code number* atrás y volver al paso 1 empezando un elemento a la derecha del elemento que había sido previamente seleccionado. Repetir este proceso hasta la última columna del primer *code number* testado. El resultado es que la última combinación generada en el paso 8 es la que tiene menos tiempo ocioso para esa estación en concreto (la mínima diferencia entre tiempo de ciclo y tiempo de carga de la estación).
- Paso 10: reemplazar el primer *code number* con el último *code number* correspondiente a la combinación definitiva del paso 9 (eliminar de la matriz de precedencias las tareas ya seleccionadas y calcular de nuevo el vector).
- Paso 11: repetir los pasos previos hasta que todos los elementos hayan sido asignados.

5.1.2. Procedimientos de optimización local de soluciones

Los procedimientos de optimización local de soluciones sirven para mejorar las soluciones obtenidas por las heurísticas base. Esto se hace mediante la generación de soluciones vecinas de la inicial, y, finalmente seleccionando la mejor de todas (incluida la inicial).

La base de los procedimientos de optimización local de soluciones desarrollados en este proyecto es el trabajo de Moodie y Young (1965, p.23-29), donde se define un método llamado *trade and transfer*, cuyo significado es:

- *Trade*: intercambio de tareas entre una estación y otra.
- *Transfer*: transferencia de una tarea de una estación a otra.

El *trade and transfer* consta de los siguientes pasos:



- Paso 1: determinar las estaciones más y menos cargadas del balance inicial obtenido de la aplicación de una heurística.
- Paso 2: definir el parámetro GOAL como $GOAL = (ST_{max} - ST_{min})/2$ donde ST_{max} y ST_{min} son el tiempo de trabajo de la estación más cargada y el de la menos cargada respectivamente.
- Paso 3: determinar todas las tareas de la estación más cargada cuya duración sea inferior a $2 \times GOAL$, y no violen las reglas de precedencia si se transfieren a la estación menos cargada.
- Paso 4: determinar todos los posibles intercambios y transferencias entre tareas que forman parte de la estación más cargada y tareas que forman parte de la estación menos cargada, que permitan que la reducción de tiempo de ST_{max} y el aumento de ST_{min} sea menor que $2 \times GOAL$.
- Paso 5: llevar a cabo el trade o el transfer con menor diferencia absoluta entre sí mismo y el valor GOAL.
- Paso 6: si no es posible ningún trade ni ningún transfer, entre la estación más cargada y la menos cargada, intentar *trade and transfer* entre la segunda más cargada y la menos cargada, y así sucesivamente; si es posible, volver al paso 2.
- Paso 7: si no se puede hacer ningún trade ni ningún transfer más, fijar la estación menos cargada e intentar *trade and transfer* entre la estación más cargada y la segunda menos cargada (volver a paso 2).

Para facilitar la comprensión del proceso anteriormente descrito, éste se esquematiza en el gráfico 5.2, donde se representa la evolución de la distribución de la carga de trabajo según se va aplicando el *trade and transfer*. Las tareas que componen cada estación están caracterizadas por un color, y el número que se encuentra en su interior es el tiempo de proceso de dicha tarea. Encima de cada columna se especifica un número que es la carga total de trabajo de cada estación.



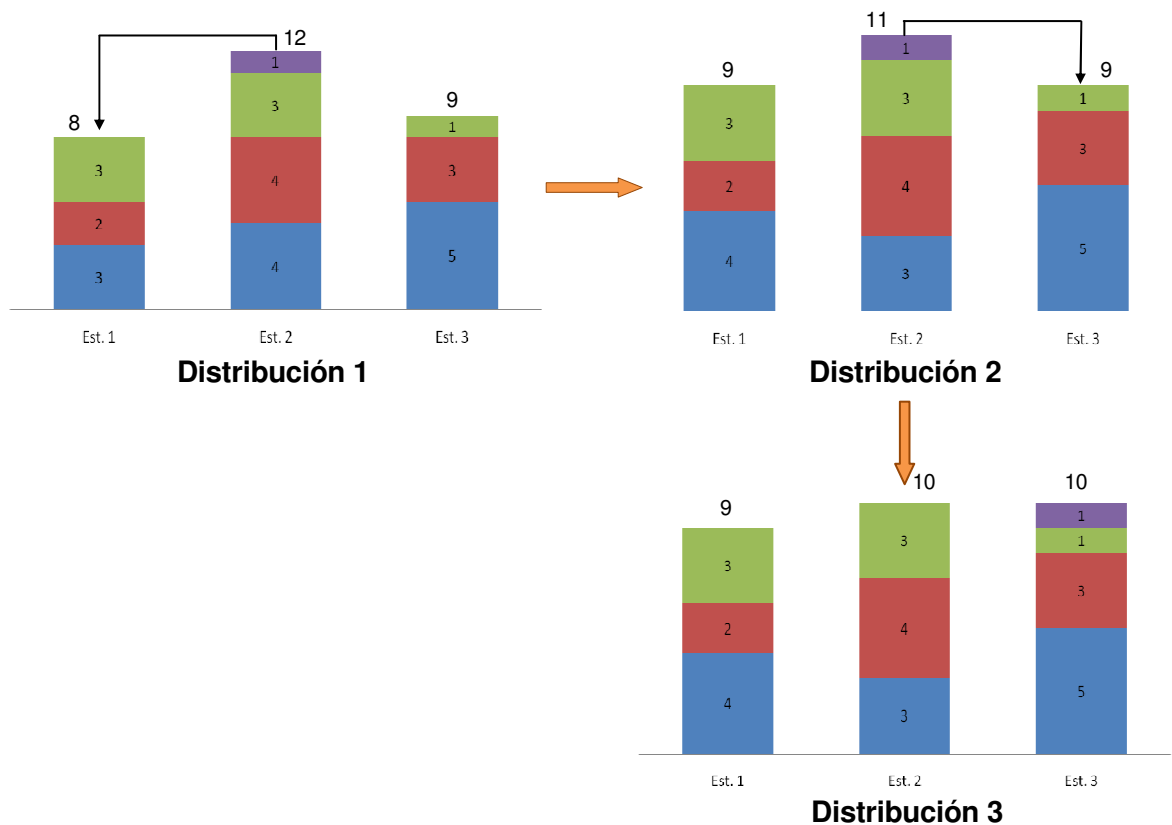


Gráfico 5.2 Representación del método *trade and transfer*

tareas entre la estación más cargada (Est. 2) y la estación menos cargada (Est. 1). Finalmente se lleva a cabo el intercambio entre las dos tareas de color azul de las estaciones 1 y 2. Al no poder realizarse ningún intercambio ni transferencia más (por restricciones de precedencia), se obtiene la distribución 2 de carga de trabajo. En la distribución 2 se busca el posible *trade* o *transfer* de tareas entre la estación más cargada (Est. 2) y la segunda estación menos cargada (Est. 3). Finalmente se lleva a cabo la transferencia de la tarea de color lila de la estación 2 a la 3, obteniendo así la distribución 3. La distribución 3 se da como definitiva ya que ningún intercambio ni transferencia de tareas permitiría mejorar esta solución.

A continuación se definen los 6 métodos de optimización local de soluciones que se han desarrollado en este proyecto. Como se ha comentado, estos procedimientos se basan en el procedimiento de *trade and transfer* explicado anteriormente.

5.1.2.1. Método 1

En el gráfico 5.3 se muestra un ejemplo de distribución de carga en una línea de montaje. Cada columna representa el tiempo de cada estación, y está dividida en tantas secciones



como tareas tiene asignada la estación en concreto. En cada sección se incluye un número que es el tiempo de proceso de cada tarea. Encima de cada columna se especifica un número, que es la carga de trabajo de cada estación



Gráfico 5.3 Gráfico para ayuda de explicación Método 1

Basándose en este gráfico se procede a explicar el funcionamiento del método 1 de generación de vecindarios, cuya principal característica es que **siempre se realiza el primer trade o transfer posible** que se encuentre cuando se buscan intercambios o transferencias de tareas entre dos estaciones:

- Paso 1: buscar y realizar el primer *trade* o *transfer* posible entre la estación más cargada y la estación menos cargada, de forma que se mejore la solución inicial. En el caso que se trata, entre las estaciones 2 (más cargada) y 1 (menos cargada).
- Paso 2: si no se puede realizar ningún *trade* ni ningún *transfer* entre la estación más cargada (2) y la menos cargada (1) o ya se han hecho todos los posibles, se debe fijar de forma provisional la estación más cargada (2), y volver a paso 1 considerando como estación más cargada la segunda más cargada (3).
- Paso 3: repetir sucesivamente el paso 2 hasta que se haya probado la posible transferencia o intercambio de tareas entre todas las estaciones y la menos cargada.
- Paso 4: fijar de forma definitiva la estación menos cargada (1) y volver al paso 1 considerando como estación menos cargada, la segunda menos cargada (3).
- Paso 5: repetir sucesivamente los pasos 1 a 4 hasta que se hayan considerado todas las estaciones, menos una (la más cargada de todas), como estación menos



cargada en algún momento. Siempre que se haya efectuado algún *trade* o algún *transfer* se vuelve a reiniciar el proceso, y consecuentemente, no se considera ninguna estación fija.

Todos estos pasos quedan reflejados de forma esquemática en el gráfico 5.4, donde las flechas negras simbolizan la generación de vecindario entre dos estaciones, y los números encima de las columnas, la carga de cada estación.

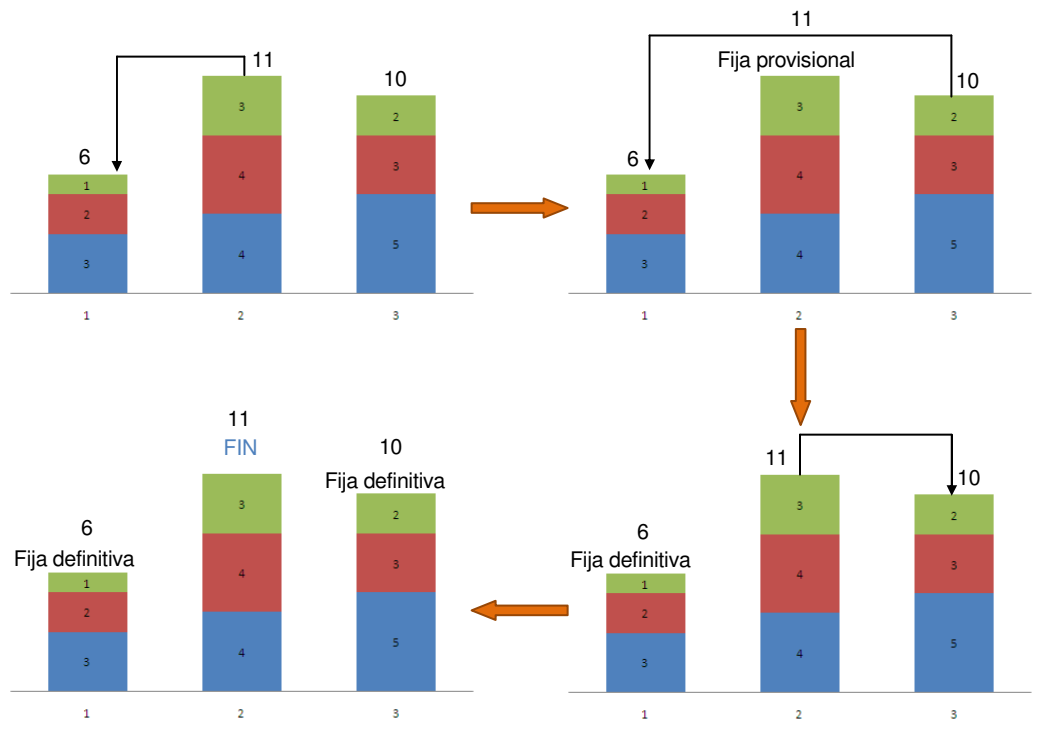


Gráfico 5.4 Representación del método 1 de generación de vecindarios

5.1.2.2. Método 2

El método 2 de generación de vecindarios consta de los mismos pasos que el método 1. El hecho que los diferencia es que en el caso 2, no se realiza el primer intercambio o transferencia posible, sino que se hace una búsqueda **de todos los posibles intercambios o transferencias entre las dos estaciones que se estudian, y se lleva a cabo el mejor (siempre que se mejore la solución inicial)**, es decir, aquél que dé una mejor solución (según el parámetro δ).

5.1.2.3. Método 3

Para el análisis del método 3 también se tiene en cuenta el ejemplo planteado en el gráfico 5.3. Este método de generación de vecindarios consta de los siguientes pasos:



- **Paso 1:** generar **todos los intercambios o transferencias posibles entre la estación más cargada (2) y el resto de estaciones (1 y 3)**. Se lleva a cabo aquella acción que permite obtener la mejor solución posible, siempre que se mejore.
- **Paso 2:** una vez se han hecho todos los *trade* y *transfers* posibles entre la estación más cargada y el resto se pasa a fijar definitivamente la estación más cargada. Posteriormente, se vuelve al paso 1 considerando como estación más cargada la segunda estación más cargada.
- **Paso 3:** repetir sucesivamente el paso 2 hasta que se han considerado todas las estaciones, menos una (la menos cargada de todas), como estación más cargada en algún momento. Siempre que se efectúe algún *trade* o algún *transfer* se vuelve a reiniciar el proceso, y consecuentemente, no se considera ninguna estación fija.

El siguiente gráfico (gráfico 5.5) muestra una representación esquemática de los pasos que componen el método 3:

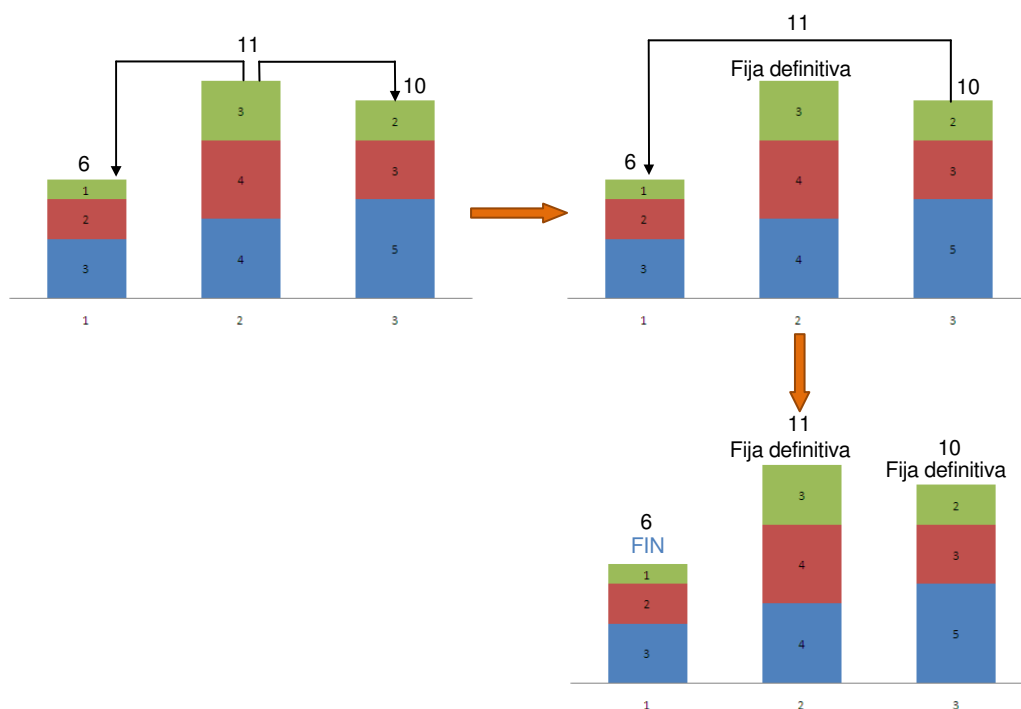


Gráfico 5.5 Representación del método 3 de generación de vecindarios



5.1.2.4. Método 4

El método 4 consta de dos partes: la primera de ellas es la aplicación del método 1 (definido en 5.1.2.1) y la segunda consiste en desarrollar una estrategia de búsqueda de vecinos nueva. Para la descripción de este método también se considera el ejemplo que ilustra el gráfico 5.3:

- Paso 1: aplicación del método 1. Por lo tanto, para el desarrollo de la segunda parte del método 4 se parte de la solución final obtenida de la aplicación del método 1.
- Paso 2: buscar y realizar el **primer *trade* o *transfer* posible entre la estación más cargada y la estación menos cargada**. En el caso que se trata, entre las estaciones 2 (más cargada) y 1 (menos cargada).
- Paso 3: si no se puede realizar ningún *trade* ni ningún *transfer* entre la estación más cargada (2) y la menos cargada (1) o ya se han hecho todos los posibles, se debe fijar provisionalmente la estación menos cargada (1), y volver a paso 1 considerando como estación menos cargada la segunda menos cargada (3).
- Paso 4: repetir sucesivamente el paso 3 hasta que se haya probado la posible transferencia o intercambio de tareas entre todas las estaciones y la más cargada.
- Paso 5: fijar de forma definitiva la estación más cargada (1) y volver al paso 1 considerando como estación más cargada, la segunda más cargada (3).
- Paso 6: repetir sucesivamente los pasos 2 a 5 hasta que se hayan considerado todas las estaciones, menos una (la menos cargada de todas), como estación más cargada en algún momento. Siempre que se efectúe algún *trade* o algún *transfer* se vuelve a reiniciar el proceso, y consecuentemente, no se considera ninguna estación fija.

El gráfico 5.6 ilustra los pasos necesarios para realizar la segunda parte del método 4, es decir, los pasos 2 a 6.



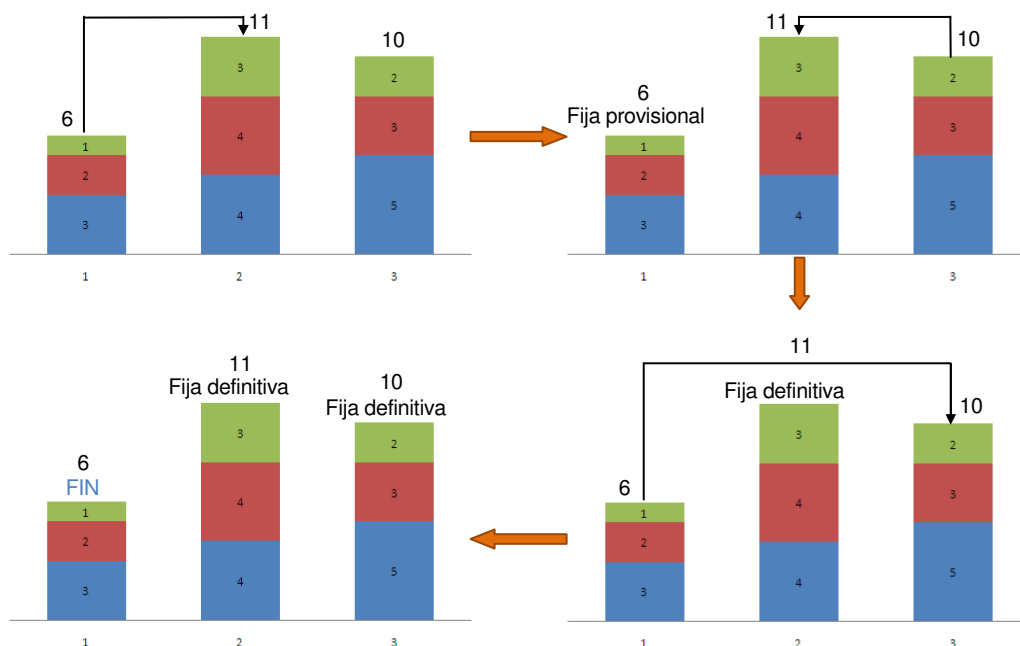


Gráfico 5.6 Representación del método 4 de generación de vecindarios

5.1.2.5. Método 5

El método 5 de generación de vecindarios consta de los mismos pasos que el método 4, pero con la diferencia de que en el caso 5, no se realiza el primer intercambio o transferencia posible cuando se analizan dos estaciones. En este caso se hace una búsqueda de **todos los posibles intercambios o transferencias entre las dos estaciones tratadas y se lleva a cabo el posible** (siempre que se mejore la solución inicial), es decir, aquel que proporciona una mejor solución (considerando el parámetro δ).

5.1.2.6. Método 6

El método 6 es una ampliación del método 3 de generación de vecinos. Los pasos que componen este método son:

- **Paso 1:** aplicar el método 3. Por lo tanto, para el desarrollo de la segunda parte del método 6 se parte de la solución final obtenida de la aplicación del método 3.
- **Paso 2:** generar **todos los intercambios o transferencias posibles entre la estación menos cargada (1) y el resto de estaciones (2 y 3)**. Se lleva a cabo



aquella acción que permita obtener la mejor solución posible (siempre que se mejore la solución de partida).

- **Paso 3:** una vez se han hecho todos los *trade* y *transfers* posibles entre la estación menos cargada y el resto se pasa a fijar definitivamente la estación menos cargada. Posteriormente, se vuelve al paso 1 considerando como estación menos cargada la segunda estación menos cargada.
- **Paso 4:** repetir sucesivamente el paso 2 hasta que se hayan considerado todas las estaciones menos una (la más cargada de todas) como estación menos cargada en algún momento. Siempre que se haya efectuado algún *trade* o algún *transfer* se vuelve a reiniciar el proceso, y consecuentemente, no se considera ninguna estación fija.

Para facilitar la comprensión de este proceso se puede observar el gráfico 5.7 donde aparece esquematizado el método 6.

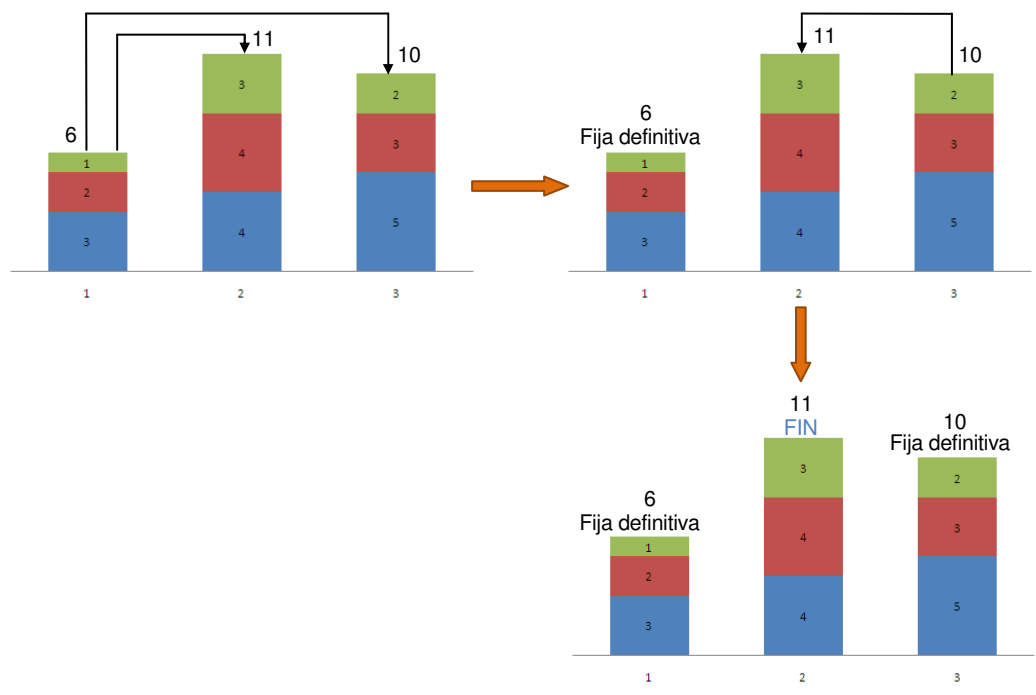


Gráfico 5.7 Representación del método 6 de generación de vecindarios

5.1.3. Heurísticas para la resolución del LB-ALBP

Con las dos variantes de la heurística "principal" y los 6 métodos de optimización local de soluciones, se puede detallar las características de las 10 heurísticas desarrolladas en este trabajo para el LB-ALBP. Los métodos de optimización local se aplican a la solución de



todas las heurísticas base. En la tabla 5.2 se clasifica cada una de las heurísticas diseñadas según el método de optimización local que se aplica y el tipo de heurística principal que se utiliza.

Heurística	Método de generación de vecindarios	Tipo de heurística principal aplicada
Heurística 1	Ninguno	Heurística principal 1
Heurística 2A	Método 1	Heurística principal 1
Heurística 2B	Método 2	Heurística principal 1
Heurística 2C	Método 3	Heurística principal 1
Heurística 3A	Método 4	Heurística principal 1
Heurística 3B	Método 5	Heurística principal 1
Heurística 3C	Método 6	Heurística principal 1
Heurística 4A	Método 4	Heurística principal 2
Heurística 4B	Método 5	Heurística principal 2
Heurística 4C	Método 6	Heurística principal 2

Tabla 5.2 Características de las 10 heurísticas desarrolladas para el LB-ALBP

En las heurísticas en las que se utiliza la heurística principal 2 (Heurísticas 4A, 4B y 4C) no se aplican los métodos de optimización local de soluciones 1, 2 y 3, debido a que estos ya están incluidos en los métodos 4, 5 y 6, respectivamente. Es decir, como el método 4 parte de la solución que se obtiene del método 1, la solución que da la ejecución del método 4, será siempre igual o mejor que la del método 1.

5.2. Experiencia computacional: ejemplares

Los procedimientos heurísticos diseñados han sido aplicados a 16 problemas test propuestos en la base de datos de <http://www.assembly-line-balancing.de/>. Para cada



problema test, se dispone de la totalidad de sus valores de definición: tareas, duración de cada tarea y relaciones de precedencia. Dichos problemas forman un grupo completo y variado de ejemplares a resolver, ya que presentan un amplio de número de estaciones (desde 3 hasta 30), de tareas (desde 29 hasta 148), de precedencias (desde 32 hasta 181) y de media de duraciones individuales de las tareas (desde 6 hasta 1355 unidades de tiempo). La base de problemas test resueltos se especifica en la tabla 5.3.

Problema	Ejemplares	Nº de estaciones	Nº de tareas	Order Strength
Buxey	8	7 a 14	29	50,74
Sawyer	8	7 a 14	30	44,83
Lutz1	5	8 a 12	32	83,47
Gunther	10	6 a 15	35	59,50
Killbridge	9	3 a 11	45	44,55
Hahn	8	3 a 10	53	83,82
Warnecke	27	3 a 29	58	59,10
Tonge	23	3 a 25	70	59,42
Wee-mag	28	3 a 30	75	22,67
Arcus1	20	3 a 22	83	59,09
Lutz2	20	9 a 28	89	77,55
Lutz3	21	3 a 23	89	77,55
Mukherje	24	3 a 26	94	44,80
Arcus2	25	3 a 27	111	40,38
Barthold	13	3 a 15	148	25,80
Barthold2	25	27 a 51	148	25,80
TOTAL	274			

Tabla 5.3 Base de datos de los ejemplares resueltos



En la tabla anterior se especifica, por este orden, el nombre del problema test, el número de ejemplares total que se resuelven, el número de estaciones con el que se resuelve cada ejemplar, el número de tareas del problema test, y el *order strength*. El parámetro *order strength* considera el número de relaciones de precedencia y el número de tareas y se detalla el apartado 5.2.2.2.

5.3. Experiencia computacional: análisis de resultados

En este apartado se analizan los resultados obtenidos con las diez heurísticas diseñadas después de solucionar, con cada una de ellas, los 274 ejemplares detallados en el apartado 5.2. En primer lugar, los resultados de las 10 heurísticas, se comparan con las soluciones resultantes de aplicar los procedimientos heurísticos existentes en la literatura (las aquí denominadas heurísticas base), para comprobar el grado de mejora que se obtiene. Posteriormente, se comparan las diez heurísticas entre sí para estudiar cuál ofrece unos mejores resultados. Finalmente, las heurísticas diseñadas se comparan con 58 resultados óptimos obtenidos en el trabajo de Pastor (2007).

Todos los procedimientos han sido programados y ejecutados con el programa Microsoft Visual Basic 2005 Express Edition. Se ha trabajado con un ordenador personal con procesador Intel Pentium IV CPU 3.40GHz 512MB RAM.

5.3.1. Comparación: Heurísticas diseñadas – Heurísticas base

El objetivo de este apartado es valorar el grado de mejora que aportan las 10 heurísticas diseñadas en este trabajo en comparación con otras heurísticas de la literatura. Éstas han sido seleccionadas del trabajo de Ponnambalam, Avarindam y Mogileeswar (1999, p.577-586), y corresponden a las heurísticas base detalladas en el apartado 5.1.1.3.

Los resultados de la heurística 1 se comparan con el mejor de los resultados de las heurísticas base, sin aplicar ningún procedimiento de optimización local. El resto de heurísticas (2a, 2b, 2c, 3a, 3b, 3c, 4a, 4b y 4c), se comparan con la mejor de las soluciones obtenidas de las heurísticas base tras aplicar a todas el procedimiento *trade and transfer* de Moodie y Young (1965, p.23-29), definido en el apartado 5.1.2. Para realizar las comparaciones se utiliza el parámetro δ .

En la tabla 5.4 se muestran los resultados que se obtienen al comparar, para los 274 ejemplares, los resultados de la mejor de las heurísticas base con las soluciones de las 10 heurísticas nuevas. Los datos que aparecen en esta tabla son: el número de ejemplares en el que las heurísticas base obtienen la mejor solución (Mejor); y el valor mínimo (δ_{\min}), medio (δ_{med}) y máximo (δ_{\max}) del parámetro δ (que se calcula, para cada ejemplar,



comparando la mejor de las heurísticas base con la mejor de las soluciones obtenidas). Cabe remarcar, que las heurísticas diseñadas en este proyecto obtienen siempre la mejor solución en los 274 ejemplares solucionados (todas las δ son iguales a cero); esto es debido a que todas ellas parten de la mejor de las soluciones de las heurísticas base.

Heurística	Mejor	δ_{\min}	δ_{med}	δ_{\max}
Heurísticas base (sin <i>trade and transfer</i>) Heurística 1	– 161 (58,8%)	0	0,04	2,90
Heurísticas base + <i>trade and transfer</i> Heurística 2a	– 190 (69,3%)	0	0,0005	0,035
Heurísticas base + <i>trade and transfer</i> Heurística 2b	– 176 (64,2%)	0	0,0089	1,47
Heurísticas base + <i>trade and transfer</i> Heurística 2c	– 175 (63,9%)	0	0,0095	1,47
Heurísticas base + <i>trade and transfer</i> Heurística 3a	– 161 (58,8%)	0	0,037	2,89
Heurísticas base + <i>trade and transfer</i> Heurística 3b	– 155 (56,6%)	0	0,037	2,90
Heurísticas base + <i>trade and transfer</i> Heurística 3c	– 152 (55,5%)	0	0,036	2,90
Heurísticas base + <i>trade and transfer</i> Heurística 4a	– 160 (58,4%)	0	0,037	2,90
Heurísticas base + <i>trade and transfer</i> Heurística 4b	– 153 (55,8%)	0	0,037	2,90
Heurísticas base + <i>trade and transfer</i> Heurística 4c	– 151 (55,1%)	0	0,036	2,90

Tabla 5.4 Tabla comparativa en las heurísticas base y las heurísticas desarrolladas



El análisis de la tabla 5.4 se divide en dos partes: la correspondiente a la heurística 1 y la correspondiente al resto de procedimientos heurísticos (heurísticas 2a, 2b, 2c, 3a, 3b, 3c, 4a, 4b y 4c).

Al comparar las heurísticas base (sin *trade and transfer*) con la heurística 1 se obtiene que en 161 ejemplares las heurísticas de la literatura consiguen igualar a la heurística 1 y en 113 ejemplares la heurística 1 es mejor. Por lo tanto, en un 41,2% de los ejemplares, las nuevas heurísticas mejoran el mejor resultado obtenido por las heurísticas existentes en la literatura.

Analizando los resultados del segundo grupo de heurísticas, se observa que todas las heurísticas mejoran los resultados de las heurísticas base (con aplicación del *trade and transfer*). La heurística que obtiene un grado de mejora menor es la heurística 2a, esto se puede observar porque es la que mejora los resultados de la literatura en un menor número de ejemplares (84) y la que tiene una δ_{med} menor (0,0005). Por su parte, la heurística 4c es la que obtiene unos mejores resultados, ya que mejora la mejor de las soluciones de las heurísticas base en 123 ejemplares de los 274 posibles.

La principal conclusión parcial que se extrae de este análisis de los resultados obtenidos es que las heurísticas diseñadas igualan o mejoran los resultados de los procedimientos ya existentes en la literatura (las heurísticas base) en el 100% de los casos.

5.3.2. Comparación: heurísticas diseñadas

5.3.2.1. Análisis comparativo de las heurísticas diseñadas

En este apartado se analizan los resultados obtenidos para los 274 ejemplares solucionados con los diez procedimientos heurísticos diseñados en este trabajo.

Para cada heurística, la tabla 5.5 muestra el número de ejemplares en el que se obtiene la mejor solución (Mejor); el mínimo (δ_{min}), medio (δ_{med}) y máximo (δ_{max}) valor del parámetro δ (que es calculado, para cada ejemplar, mediante la comparación de cada solución con la mejor solución obtenida usando los diez procedimientos heurísticos); y el mínimo (t_{min}), medio (t_{med}) y máximo (t_{max}) tiempos de ejecución en segundos.



Heurística	Mejor	δ_{\min}	δ_{med}	δ_{\max}	t_{\min}	t_{med}	t_{\max}
Heurística 1	122	0	0,0811	3,15	0	443	8.200
Heurística 2a	174	0	0,0411	2,90	4	1.917	11.207
Heurística 2b	183	0	0,0411	2,90	4	1.935	11.118
Heurística 2c	188	0	0,0404	2,90	5	2.545	23.305
Heurística 3a	213	0	0,0122	2,06	7	2.880	15.577
Heurística 3b	227	0	0,0132	1,51	7	2.880	16.412
Heurística 3c	237	0	0,0134	2,06	7	4.639	49.569
Heurística 4a	215	0	0,0121	2,06	7	2.867	15.577
Heurística 4b	228	0	0,0131	1,51	7	2.865	16.287
Heurística 4c	241	0	0,0133	2,06	7	5.534	49.893

Tabla 5.5 Resultados de las heurísticas para 274 ejemplares

La heurística 1, a pesar de tener unos tiempos de cálculo inferiores, presenta un notable peor comportamiento que el resto de procedimientos, ya que solo obtiene la mejor solución en 122 ejemplares sobre los 274 posibles. Además, cuando la heurística 1 no obtiene la mejor solución, es la que peor comportamiento presenta en comparación al resto, tal y como se puede comprobar con los valores de δ_{med} (0,0811) y δ_{\max} (3,15). Dados estos resultados, a continuación se prescinde de dicha heurística y se vuelven a analizar las otras nueve heurísticas. En la tabla 5.6 se muestran estos resultados.



Heurística	Mejor	δ_{\min}	δ_{med}	δ_{\max}	t_{\min}	t_{med}	t_{\max}
Heurística 2a	174	0	0,0411	2,90	4	1.917	11.207
Heurística 2b	183	0	0,0411	2,90	4	1.935	11.118
Heurística 2c	188	0	0,0404	2,90	5	2.545	23.305
Heurística 3a	213	0	0,0122	2,06	7	2.880	15.577
Heurística 3b	227	0	0,0132	1,51	7	2.880	16.412
Heurística 3c	237	0	0,0134	2,06	7	4.639	49.569
Heurística 4a	215	0	0,0121	2,06	7	2.867	15.577
Heurística 4b	228	0	0,0131	1,51	7	2.865	16.287
Heurística 4c	241	0	0,0133	2,06	7	5.534	49.893

Tabla 5.6 Resultados de las heurísticas para 274 ejemplares

Como era de esperar, las heurísticas 2a, 2b y 2c también presentan peores resultados que el resto. Esto se puede comprobar mediante el número de ejemplares en los que obtienen la mejor solución, que es inferior al resto, y al valor de δ_{med} , que es del orden de cuatro veces el valor de δ_{med} de las otras heurísticas.

Consecuentemente, se puede extraer la conclusión de que, como ya se sabía, los métodos de optimización local de soluciones 1, 2 y 3 presentan un peor comportamiento que los 4, 5 y 6, que son los usados por las heurísticas 3a, 3b, 3c, 4a, 4b y 4c. Debido a esto, a continuación se vuelve a realizar la comparación de los resultados de las heurísticas (tabla 5.7), pero prescindiendo en este caso también de los resultados de las heurísticas 2a, 2b y 2c.



Heurística	Mejor	δ_{\min}	δ_{med}	δ_{\max}	t_{\min}	t_{med}	t_{\max}
Heurística 3a	215	0	0,0122	2,06	7	2.880	15.577
Heurística 3b	230	0	0,0132	1,51	7	2.880	16.412
Heurística 3c	239	0	0,0134	2,06	7	4.639	49.569
Heurística 4a	217	0	0,0121	2,06	7	2.867	15.577
Heurística 4b	231	0	0,0131	1,51	7	2.865	16.287
Heurística 4c	243	0	0,0133	2,06	7	5.534	49.893

Tabla 5.7 Resultados de las heurísticas para 274 ejemplares

Las heurísticas 4c y 3c son las que muestran un mejor comportamiento debido al número de ejemplares en el que obtienen la mejor solución (243 y 239 respectivamente). Estas dos heurísticas tienen en común que se les aplica el método 6 de optimización local de soluciones. Por lo tanto, se puede afirmar que este método es el que permite conseguir unos mejores resultados, aunque tienen peor δ_{med} y no tienen el mejor δ_{\max} .

Cabe remarcar el hecho de que la mejora de los resultados depende más del método de optimización local utilizado, que de la versión de la heurística principal aplicada (aunque también presenta cierta influencia). En la tabla 5.8, donde se muestra el número de ejemplares en los que cada heurística obtiene la mejor solución (Mejor), se evidencia esta conclusión.

Heurística	Mejor	Mejor	Heurística
Heurística 3a	215	217	Heurística 4a
Heurística 3b	230	231	Heurística 4b
Heurística 3c	239	243	Heurística 4c

Tabla 5.8 Tabla para el análisis de la mejora de resultados



Las heurísticas de la columna izquierda de la tabla 5.8 son aquellas a las cuales se les aplica la heurística principal 1, mientras que a las de la derecha se les aplica la heurística principal 2. Por su parte, en la primera, segunda y tercera fila aparecen las heurísticas a las que se les aplican los métodos de optimización local de soluciones 4, 5 y 6 respectivamente. Se observa que el número de veces en que una heurística obtiene la mejor solución varía más por filas (según el método de optimización local de soluciones utilizado), que por columnas (según la heurística principal aplicada). De todas formas, se puede afirmar que la heurística principal 2 permite obtener unos resultados ligeramente mejores.

A pesar de que se consideran las heurísticas 3c y 4c como las mejores heurísticas diseñadas por los resultados que ofrecen, se debe destacar que su tiempo de computación es, en ocasiones, muy grande (del orden de 50.000 segundos, aproximadamente 14 horas).

5.3.2.2. Criterios de utilización de las heurísticas diseñadas

En este apartado se realiza un análisis que permite comprobar si las características de los ejemplares influyen en el comportamiento de las diez heurísticas diseñadas. Las características que se estudian son:

- Número de tareas: n
- Número de estaciones: m
- *Order strength*: $OS = \frac{n^\circ \text{ relaciones de precedencia}}{n \cdot (n - 1)}$, siendo n el número de tareas.

5.3.2.2.1 Número de tareas

Los 274 ejemplares que se han resuelto se dividen en dos grupos según el número de tareas que tengan. El primer grupo está formado por los 211 ejemplares con un número de tareas comprendido entre 29 y 94, y el segundo grupo, por los 63 ejemplares con un número de tareas comprendido entre 95 y 148.

Para cada heurística, en la tabla 5.9 se muestra el número de ejemplares (con $29 < n < 94$) en el que se obtiene la mejor solución (Mejor); el mínimo (δ_{\min}), medio (δ_{med}) y máximo (δ_{\max}) valor del parámetro δ (que es calculado, para cada ejemplar, mediante la comparación de cada solución con la mejor solución obtenida usando los diez procedimientos heurísticos); y el mínimo (t_{\min}), medio (t_{med}) y máximo (t_{\max}) tiempos de ejecución en segundos.



Heurística	Mejor	δ_{\min}	δ_{med}	δ_{\max}	t_{\min}	t_{med}	t_{\max}
Heurística 3a	170	0	0,0147	2,06	7	906	4.907
Heurística 3b	181	0	0,0151	1,51	7	898	4.722
Heurística 3c	182	0	0,0173	2,06	7	925	5.195
Heurística 4a	172	0	0,0146	2,06	7	897	4.852
Heurística 4b	182	0	0,0151	1,51	7	893	4.689
Heurística 4c	186	0	0,0172	2,06	7	921	5.198

Tabla 5.9 Resultados para los 211 ejemplares con un número de tareas comprendido entre 29 y 94

La heurística 4c es la que logra la mejor solución en un mayor número de ejemplares (186 sobre 211 posibles). Además, los tiempos de cálculo que presenta, a pesar de ser los más altos (tanto t_{med} como t_{\max}), son del mismo orden de magnitud que el resto de procedimientos. Por lo tanto, se puede considerar la heurística 4c como la más recomendable cuando se tienen ejemplares con un número de tareas comprendido entre 29 y 94.

A continuación, en la tabla 5.10 se presenta la misma información que en la tabla 5.9 para los ejemplares con $95 < n < 148$.

Heurística	Mejor	δ_{\min}	δ_{med}	δ_{\max}	t_{\min}	t_{med}	t_{\max}
Heurística 3a	43	0	0,0147	0,221	2.911	9.493	15.777
Heurística 3b	46	0	0,0067	0,222	2.229	9.529	16.412
Heurística 3c	55	0	0,0004	0,013	2.565	17.076	49.569
Heurística 4a	43	0	0,0004	0,221	2.871	9.466	15.577
Heurística 4b	46	0	0,0067	0,222	2.207	9.468	16.287
Heurística 4c	55	0	0,0004	0,013	2.556	20.980	49.893

Tabla 5.10 Resultados para los 63 ejemplares con un número de tareas comprendido entre 95 y 148



Las heurísticas 3c y 4c son las que presentan unos mejores resultados ya que ambas tienen el máximo número de mejores soluciones (55 sobre 63 posibles). Además, presentan los mismos valores de $\bar{\delta}_{\min}$ (0), $\bar{\delta}_{\text{med}}$ (0,0004) y $\bar{\delta}_{\max}$ (0,013). Como los tiempos de cálculo de la heurística 3c son menores que los de la 4c, se recomienda el uso de la heurística 3c para los ejemplares con un número de tareas comprendido entre 95 y 148.

En la tabla 5.11 se especifican las heurísticas a utilizar si se tiene en cuenta el criterio de número de tareas.

Número de tareas (n)	Heurística a aplicar
$29 < n < 94$	Heurística 4c
$95 < n < 148$	Heurística 3c

Tabla 5.11 Resumen del criterio: número de tareas

5.3.2.2.2 Número de estaciones:

Los 274 ejemplares resueltos se dividen en dos grupos según el número de estaciones que tengan. El primero de estos dos grupos está formado por los ejemplares con un número de estaciones comprendido entre 3 y 26, y el segundo grupo, por los ejemplares con un número de estaciones comprendido entre 27 y 51.

Para cada heurística, la tabla 5.12 muestra el número de ejemplares (con $3 < m < 26$) en el que cada heurística obtiene la mejor solución (Mejor); el mínimo ($\bar{\delta}_{\min}$), medio ($\bar{\delta}_{\text{med}}$) y máximo ($\bar{\delta}_{\max}$) valor del parámetro $\bar{\delta}$ (que es calculado, para cada ejemplar, mediante la comparación de cada solución con la mejor solución obtenida usando los diez procedimientos heurísticos); y el mínimo (t_{\min}), medio (t_{med}) y máximo (t_{\max}) tiempos de ejecución en segundos.



Heurística	Mejor	δ_{\min}	δ_{med}	δ_{\max}	t_{\min}	t_{med}	t_{\max}
Heurística 3a	189	0	0,0139	2,06	7	1.861	11.752
Heurística 3b	197	0	0,0150	1,51	7	1.840	12.407
Heurística 3c	208	0	0,0153	2,06	7	2.032	15.503
Heurística 4a	191	0	0,0138	2,06	7	1.846	11.587
Heurística 4b	198	0	0,0150	1,51	7	1.827	12.293
Heurística 4c	211	0	0,0152	2,06	7	2.200	28.793

Tabla 5.12 Resultados para los 239 ejemplares con un número de estaciones comprendido entre 3 y 26

La heurística 4c es la que obtiene un mayor número de veces la mejor solución para los 239 ejemplares probados (211), por lo tanto, se considera este procedimiento como el mejor posible para los ejemplares con un número de estaciones comprendido entre 3 y 26. A pesar de esto, se debe destacar que su tiempo máximo de cálculo es muy elevado en comparación con el resto de heurísticas. En caso de que se considere este tiempo excesivo, la segunda opción debería ser la heurística 3c (208 mejores soluciones).

A continuación, en la tabla 5.13 se muestra la misma información que en la tabla 5.12 para los ejemplares con $27 < m < 51$.

Heurística	Mejor	δ_{\min}	δ_{med}	δ_{\max}	t_{\min}	t_{med}	t_{\max}
Heurística 3a	24	0	0,0008	0,0156	247	9.843	15.577
Heurística 3b	30	0	0,0005	0,0157	254	9.985	16.412
Heurística 3c	29	0	0,0005	0,0159	322	22.442	49.569
Heurística 4a	24	0	0,0008	0,0156	252	9.844	15.577
Heurística 4b	30	0	0,0005	0,0157	255	9.952	16.287
Heurística 4c	30	0	0,0005	0,0159	320	28.299	49.893

Tabla 5.13 Resultados para los 35 ejemplares con un número de estaciones comprendido entre 27 y 51



Las heurísticas 3b, 4b y 4c obtienen el mismo número de soluciones que son mejor solución (30), pero la heurística 4c tiene unos tiempo de cálculo muy elevados en comparación con los dos otros procedimientos. Es por este motivo que se descarta la heurística 4c.

Las heurísticas 3b y 4b presentan el mismo valor de $\bar{\delta}_{\min}$, $\bar{\delta}_{\text{med}}$ y $\bar{\delta}_{\max}$. A pesar de que los tiempos de cálculo de la heurística 3b son ligeramente superiores, esta diferencia no se considera significativa. Consecuentemente se recomienda el uso indistinto de las heurísticas 3b y 4b para líneas de producción con un número de estaciones comprendido entre 27 y 51.

En la tabla 5.14 se especifican las heurísticas a utilizar si se tiene en cuenta el criterio de número de estaciones.

Número de estaciones (m)	Heurística a aplicar
$3 < m < 26$	Heurística 4c
$27 < m < 41$	Heurística 3b
	Heurística 4b

Tabla 5.14 Resumen del criterio: número de estaciones

5.3.2.2.3 Criterio de *Order Strength* (OS)

La base de datos de ejemplares utilizada en este trabajo tiene valores de *order strength* entre 22,5 y 83,82. Por los resultados obtenidos se definen dos grupos de ejemplares: aquellos que tienen un $OS < 40$ (66 ejemplares), y aquellos con un $OS \geq 40$ (208 ejemplares).

La tabla 5.15 muestra, para cada heurística, el número de ejemplares (con $OS < 40$) en el que cada una obtiene la mejor solución (Mejor); el mínimo ($\bar{\delta}_{\min}$), medio ($\bar{\delta}_{\text{med}}$) y máximo ($\bar{\delta}_{\max}$) valor del parámetro $\bar{\delta}$ (que es calculado, para cada ejemplar, mediante la comparación de cada solución con la mejor solución obtenida usando los diez procedimientos heurísticos); y el mínimo (t_{\min}), medio (t_{med}) y máximo (t_{\max}) tiempos de ejecución en segundos.



Heurística	Mejor	δ_{\min}	δ_{med}	δ_{\max}	t_{\min}	t_{med}	t_{\max}
Heurística 3a	49	0	0,0149	0,9512	222	6.699	15.577
Heurística 3b	51	0	0,0179	0,9611	197	6.922	16.412
Heurística 3c	55	0	0,0003	0,0159	185	13.820	49.569
Heurística 4a	49	0	0,0149	0,9513	219	6.702	15.777
Heurística 4b	51	0	0,0179	0,9611	186	6.888	16.288
Heurística 4c	56	0	0,0003	0,0159	197	17.479	49.893

Tabla 5.15 Resultados para los 66 ejemplares con un OS<40

La heurística 4c es la que permite obtener en un mayor número de ejemplares la mejor solución, y, por lo tanto, se recomienda este procedimiento para los ejemplares con un *order strength* inferior a 40. A pesar de esto, cabe destacar que su tiempo de cálculo máximo es muy elevado en comparación con el resto de procedimientos. En caso de que se considerara excesivo, no se recomendaría el uso de la heurística 3c (que es la segunda mejor en cuanto a mejores soluciones) porque su tiempo máximo de cálculo también es muy elevado, sino que se recomendaría el uso indistinto de las heurísticas 4b o 3b.

A continuación, en la tabla 5.16 se muestra la misma información que en la tabla 5.15 para los ejemplares con OS \geq 40.

Heurística	Mejor	δ_{\min}	δ_{med}	δ_{\max}	t_{\min}	t_{med}	t_{\max}
Heurística 3a	164	0	0,0113	2,06	7	1.669	11.752
Heurística 3b	176	0	0,0117	1,51	7	1.598	12.039
Heurística 3c	182	0	0,0176	2,06	7	1.726	13.031
Heurística 4a	166	0	0,0113	2,06	7	1.651	11.587
Heurística 4b	177	0	0,0116	1,51	7	1.588	12.114
Heurística 4c	185	0	0,0174	2,06	7	1.743	13.221

Tabla 5.16 Resultados para los 208 ejemplares con un OS \geq 40

La heurística 4c es el procedimiento que permite obtener un mayor número de mejores soluciones (185). Por lo tanto, se puede considerar esta heurística como la mejor posible para aquellos ejemplares con un OS mayor que 40.

En la tabla 5.17 se especifican las heurísticas a utilizar si se tiene en cuenta el criterio de *order strength*.

<i>Order Strength</i> (OS)	Heurística a aplicar
OS < 40	Heurística 4c
OS ≥ 40	Heurística 4c

Tabla 5.17 Resumen del criterio: *Order Strength*

5.3.3. Comparación: soluciones óptimas

Gracias a los trabajos de Pastor (2007) y de Sentinella y Solsona (2005), donde se trata la resolución del LB-ALBP mediante métodos exactos, se dispone de 58 ejemplares solucionados óptimamente. En este apartado se comparan estas soluciones óptimas con las soluciones obtenidas con las diez heurísticas diseñadas.

En la tabla 5.18 se muestran los resultados que se obtienen al comparar, para los 58 ejemplares, los resultados óptimos con las soluciones de las 10 heurísticas nuevas. Los datos que aparecen en esta tabla son: el número de ejemplares en el que las heurísticas diseñadas consiguen el óptimo (Óptimo); el valor mínimo (δ_{\min}), medio (δ_{med}) y máximo (δ_{\max}) del parámetro δ (que es calculado, para cada ejemplar, comparando el óptimo con las soluciones de las diez heurísticas diseñadas); y el tiempo mínimo (t_{\min}), medio (t_{med}) y máximo (t_{\max}) de ejecución.



Heurística	Óptimo	δ_{\min}	δ_{med}	δ_{\max}	t_{\min}	t_{med}	t_{\max}
Heurística 1	24	0	0,2310	2,14	0	202	4.330
Heurística 2a	27	0	0,2300	2,14	5	616	5.597
Heurística 2b	27	0	0,2297	2,14	4	600	5.564
Heurística 2c	27	0	0,2298	2,14	4	630	5.564
Heurística 3a	27	0	0,2292	2,14	8	967	7.316
Heurística 3b	27	0	0,2290	2,14	8	967	7.792
Heurística 3c	27	0	0,2289	2,14	7	932	7.803
Heurística 4a	27	0	0,2292	2,14	8	953	7.228
Heurística 4b	27	0	0,2288	2,14	7	924	7.715
Heurística 4c	27	0	0,2289	2,14	7	959	7.792
Óptimas	58	0	0	0	0	1.007	17.712

Tabla 5.18 Resultados para los 58 ejemplares cuyo óptimo es conocido

La heurística 1 consigue la solución óptima en 24 ejemplares (41,4%), mientras que el resto de procedimientos, las heurísticas 2a, 2b, 2c, 3a, 3b, 3c, 4a, 4b y 4c, consiguen el óptimo en 27 ejemplares (46,6%). Además, la δ_{med} es baja, por lo que se puede afirmar que, cuando las heurísticas desarrolladas no consiguen el óptimo, su solución es cercana a éste.

Los tiempos de cálculo de los procedimientos exactos son superiores al de los procedimientos heurísticos, este hecho se puede observar con los valores de t_{med} y t_{\max} . Esta diferencia es más evidente en los tiempos máximos de ejecución, ya que para los procedimientos exactos t_{\max} es igual a 17.712 unidades, y para las heurísticas diseñadas, t_{\max} oscila entre 7.792 y 4.330 unidades.

Con este análisis se puede afirmar que las heurísticas diseñadas son buenas, ya que se obtienen buenas soluciones (próximas al óptimo) con tiempos de cálculo razonables.





Conclusiones

En este trabajo se realiza un estudio sobre la resolución del *Lexicographic Bottleneck Assembly Line Balancing Problem* (LB-ALBP) mediante procedimientos heurísticos. En total se diseñan y prueban diez procedimientos heurísticos para la resolución del LB-ALBP.

La importancia del LB-ALBP dentro del problema de equilibrado de líneas de montaje es debida a que, además de considerar la carga de trabajo de la estación más cargada (el tiempo de ciclo de la línea), se considera el segundo valor máximo de carga de trabajo, el tercero, el cuarto, y así sucesivamente. Gracias a esto, se consigue mejorar la fiabilidad de la línea, ya que no es lo mismo una distribución de la carga de trabajo de 10, 10, 10, 4, 3, 3 que una que sea igual a 10, 6, 6, 6, 6, 6; a pesar de que ambas distribuciones presentan el mismo tiempo de ciclo (10), la segunda distribución es mucho más fiable, ya que reduce el número de estaciones que son cuello de botella.

Las diez heurísticas desarrolladas para la resolución del LB-ALBP son el resultado de la combinación de dos variantes de una heurística "principal" y seis métodos de optimización local de soluciones. La heurística "principal" consiste en ejecutar catorce heurísticas sencillas existentes en la literatura, llamadas heurísticas base, y quedarse con la mejor solución. A continuación, se fija la estación más cargada y se divide el problema inicial en dos sub-problemas que se resuelven de forma independiente ejecutando de nuevo las 14 heurísticas sencillas. Esta operativa se repite sucesivamente hasta fijar todas las estaciones. La optimización local se aplica a las soluciones obtenidas por todas las heurísticas base.

Los procedimientos heurísticos diseñados se denominan: 1, 2a, 2b, 2c, 3a, 3b, 3c, 4a, 4b y 4c. Como era de esperar, las cuatro primeras heurísticas (1, 2a, 2b y 2c), ofrecen unos resultados peores en comparación con los otros seis métodos de resolución, ya que obtienen la mejor solución en un número menor de ejemplares.

De las heurísticas 3a, 3b, 3c, 4a, 4b y 4c, las que dan los mejores resultados son la 3c y la 4c, ya que son las que permiten obtener la mejor solución en un mayor número de ejemplares.

Además, se realiza un análisis que permite comprobar si las características de los ejemplares influyen en el comportamiento de las diez heurísticas diseñadas. Las características que se consideran son: el número de tareas, el número de estaciones, el *order strength*.



Si se considera el criterio de número de tareas, se concluye que para ejemplares con un número de tareas comprendido entre 29 y 94, la mejor opción es aplicar la heurística 4c; mientras que si se tiene entre 95 y 148 tareas es mejor la aplicación de la heurística 3c.

Considerando el criterio de número de estaciones, se llega a la conclusión de que si se tiene entre 3 y 26 estaciones, ofrece mejores resultados la aplicación de la heurística 4c; mientras que si se tiene entre 27 y 51 estaciones, se obtienen mejores resultados con la aplicación de las heurísticas 3b o 4b indistintamente.

Cuando se considera como criterio el *order strength* (OS), se concluye que para OS menores que 40, la heurística 4c es la que permite obtener mejores resultados, aunque es remarcable que sus tiempos de cálculo son en ocasiones muy elevados. Para ejemplares que tengan un OS mayor o igual a 40 se recomienda el uso de la heurística 4c ya que ofrece también los mejores resultados.

Posteriormente, se comparan las soluciones obtenidas con las diez heurísticas diseñadas con 58 soluciones óptimas conocidas (obtenidas gracias a procedimientos exactos existentes en la literatura). Con este análisis se puede concluir que todas las heurísticas desarrolladas ofrecen un buen comportamiento, en especial las heurísticas 2a, 2b, 2c, 3a, 3b, 3c, 4a, 4b y 4c, ya que permiten obtener buenos resultados en tiempos de cálculo bajos.

Cabe destacar que la base de ejemplares utilizada constituye un conjunto representativo, ya que se ha trabajado con problemas con un gran abanico de número de estaciones (desde 3 hasta 30), de tareas (desde 29 hasta 148), de precedencias (desde 32 hasta 181) y de media de duraciones individuales de las tareas (desde 6 hasta 1355 unidades de tiempo).



Costes del estudio efectuado

El proyecto se ha desarrollado conjuntamente en las instalaciones públicas de la escuela (ETSEIB) y en el laboratorio de ingeniería de organización y logística industrial del Instituto de Optimización y Control. La evaluación económica del proyecto está basada en las siguientes partidas: coste del material, amortización, valor residual y tiempo de desarrollo.

El coste del material se expresa en la siguiente tabla:

Descripción	Coste de adquisición [€]	Coste sobre el proyecto [€]
Ordenador: HP dv1280	1.200	150
PC laboratorio de cálculo 1	1.190	133
PC laboratorio de cálculo 2	1.190	133
Impresora: HP 5940	90	35
Material Fungible de oficina	30	30
TOTAL MATERIAL		481

Tabla 1 Tabla de costes de materiales

A continuación se evalúa el coste derivado del tiempo de desarrollo del proyecto. Este trabajo ha sido realizado entre los meses de febrero y julio de 2008. Durante este periodo se han llevado a cabo las siguientes actividades: estudio del estado del arte del problema, diseño e implementación de los algoritmos en el software Microsoft Visual Basic Express Edition 2005, experimentación computacional, análisis de los resultados y realización de la memoria del proyecto.

Se considera que para la realización de este trabajo son necesarias dos personas: un ingeniero junior, cuya función es el desarrollo del proyecto; y un ingeniero sénior, que dirige y supervisa el trabajo realizado por el ingeniero junior. Además, se considera que el valor de una hora de trabajo de un ingeniero junior es de 35€, y la de un ingeniero sénior de 90€.

Así, los costes derivados del tiempo de desarrollo del proyecto se calculan de la siguiente forma:



$$6 \frac{\text{horas}}{\text{día}} \cdot 5 \frac{\text{días}}{\text{semana}} \cdot 4,2 \frac{\text{semanas}}{\text{mes}} \cdot 6 \text{ meses} = 756 \text{ horas}$$

$$756 \text{ horas Ing. junior} \cdot 35 \frac{\text{€}}{\text{hora}} = 26.460 \text{ €}$$

$$40 \text{ horas Ing. sénior} \cdot 90 \frac{\text{€}}{\text{hora}} = 3.600 \text{ €}$$

En la tabla 2 se observan en detalle los costes del proyecto, aplicando los costes indirectos, el beneficio industrial, el IVA y el beneficio propio.

Total previsto de costes	30.541,00 €
Costes indirectos (5%)	1.527,05 €
Subtotal 1	32.068,05 €
IVA (16%)	5.130,89 €
Subtotal 2	37.198,94 €
Beneficio Industrial (13%)	4.835,86 €
Subtotal 3	42.034,80 €
Beneficio Total (3.000 €)	3.000,00 €
TOTAL	45.034,80 €

Tabla 2 Resumen de costes del proyecto

Por lo tanto, el coste total del proyecto es de 45.034,80 €.



Impacto ambiental del estudio

Tal y como se ha especificado en la memoria del proyecto, el estudio efectuado, la exposición y la resolución del LB-ALBP, está aplicado sobre ejemplares genéricos de líneas de montaje. Por lo tanto, no se ha podido trabajar en una línea industrial real sobre la que se puedan extraer valores concretos para evaluar el impacto ambiental. Aún así, a continuación se exponen algunos indicadores del impacto ambiental que tendría la aplicación de los resultados obtenidos en un caso industrial real.

La principal consecuencia ambiental de la aplicación del LB-ALBP a un caso real sería la mejora de la eficiencia de la línea de montaje tratada. Esta mejora sería causada por la minimización de las cargas de trabajo de las estaciones que componen la línea. La minimización del tiempo de la estación más cargada (el tiempo de ciclo) repercute directamente en el consumo de energía y en la disminución del tiempo de trabajo necesario para realizar un producto (aumento de la productividad), minimizando así también, las emisiones atmosféricas. El impacto ambiental del proyecto se puede desglosar en los siguientes aspectos: económico, medioambiental, social y energético.

En el aspecto económico, la reducción de la energía requerida deriva en una reducción directa del coste de compra de la misma. Además, el incremento de la productividad permite obtener unos mayores beneficios.

En referencia al aspecto medioambiental, la reducción de la necesidad de energía, ya sea calorífica (derivada de combustibles) o eléctrica, afecta directamente en la disminución de las emisiones. Además, se contribuye a minimizar los riesgos de posibles accidentes derivados del transporte energético.

En el aspecto social, la reducción de emisiones permite mejorar la calidad de vida de las zonas industriales gracias a las mejoras medioambientales.

Finalmente, en referencia al aspecto energético, la reducción de la necesidad de energía evita el agotamiento prematuro de los recursos y permite las mejoras económicas, medioambientales y sociales nombradas.





Referencias bibliográficas

- AGRAVAL, P.K. The related activity concept in assembly line balancing. *International Journal of Production Research*. Vol. 29 (2), 1985, p.403-421.
- AJENBLIT, D., WAINWRIGHT, R.L. Applying genetic algorithms to the U-shaped assembly line balancing problem. Working paper, The University of Tulsa, 2003.
- ARCUS, A.L. COMSOAL: Computer Method of Sequencing Operations of Assembly Lines. *International Journal of Production Research*. Vol. 4 (3), 1966, p.259-278.
- BARD, J. Assembly line balancing with parallel workstations and dead times. *International Journal of Production Research*. Vol. 37 (4), 1999, p.721-736.
- BARTHOLDI, J.J. Balancing two-sided assembly lines: a case study. *International Journal of Production Research*. Vol. 31, 1993, p.2447-2461.
- BAUTISTA, J., PEREIRA, J. Algoritmos de hormigas para un problema de equilibrado de líneas. V Congreso de Ingeniería de Organización, Valladolid-Burgos, 2003.
- BAUTISTA, J., PEREIRA, J. Ant algorithms for a time and space constrained assembly line balancing problema. *European Journal of Operational Research*. Vol. 177, 2006, p. 2016-2032.
- BAYBARS, I. An efficient heuristic method for the simple assembly line balancing problem. *International Journal of Production Research*. Vol. 24 (1), 1986, p.149-166.
- BAYBARS, I. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*. Vol. 32, 1986, p.909-932.
- BAYKASOGLU, A., DERELI, T., EROL, R., SABANCU, I. An ant colony based optimization algorithm for solving assembly line balancing problems. *International XII Turkish Symposium on Artificial Intelligence and Neural Networks – TAINN*, 2003.
- BECKER, C., SCHOLL, A. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*. Vol. 168, 2006, p.694-715.
- BENNET, G.B., BYRD, J. A trainable heuristic procedure for the assembly line balancing problem. *AIIE Transactions*. 1976, p.195-201.
- BHATTARCHARJEE, T.K., SAHU, S. A heuristic approach to general assembly line balancing. *International Journal of Production Research*. Vol. 8 (6), 1988, p.67-77.



- BOYSEN, N., FLIEDNER, M., SCHOLL, A. A classification of assembly line balancing problems. Working paper 12/2006. Friedrich-Schiller-Universität Jena. 2006.
- BOYSEN, N., FLIEDNER, M., SCHOLL, A. A classification of assembly line balancing problems. *European Journal of Operational Research*, Vol. 183, 2007, 674-693
- BUKCHIN, J., TZUR, M. Design of flexible assembly line minimize equipment cost. *AIIE transactions*. Vol. 32, 2000, p.585-598.
- BUXEY, G.M. Assembly line balancing with multiple stations. *Management Science*. Vol. 20, 1974, p.1010-1021.
- COROMINAS, A., PASTOR, R., PLANS, J. Balancing assembly lines with skilled and unskilled workers. *Omega*. Vol. 36, 2008, p.1126-1132.
- DAR-EL, E.M. MALB-A heuristic technique for balancing large single-model assembly lines. *AIIE Transactions*. Vol. 5, 1973, p.343-356.
- DAR-EL, E.M. Solving large single-model assembly line balancing problem –a comparative study. *AIIE Transactions*. Vol. 7 (3), 1975, p.302-310.
- DAR-EL, E.M., MAZER, A. Predicting the performance of unpaced assembly lines where one station variability may be smaller than the others. *International Journal of Production Research*. Vol. 27 (12), 1989, p.2105-2116.
- EREL, E., GOKCEN, H. Theory and Methodology: shortest-route formulation of mixedmodel assembly line balancing problem. *European Journal of Operacional Research*. Vol. 116, 1999, p.194-204.
- EREL, E., SABUNCUOGLU, I., AKSU, B.A. Balancing of U-type assembly systems using simulated annealing. *International Journal of Production Research*. Vol. 39, 2001, p.3003-3015.
- EREL, E., SARIN, S.C. A survey of the assembly line balancing procedures. *Production Planning & Control*. Vol. 9, 1998, p.414-434.
- GASS, S. *An illustrated guide to linear programming*. New York, Dover Publications, 1990.
- GEN, M., TSUJIMURA, Y., LI, Y. Fuzzy assembly line balancing using genetic algorithms. *Computers & Industrial Engineering*. Vol. 31 (3/4), 1996, p.631-634.
- GHOSH, S., GAGNON, R.J. A comprehensive literature review and analysis of design, balancing and scheduling of assembly systems. *International Journal of Production Research*. Vol. 27 (4), 1989, p.637-670.



- GONÇALVES, J.F., ALMEIDA, J.R. A hybrid genetic algorithm for assembly line balancing. *Journal of Heuristics*. Vol. 8, 2002, p.629-642.
- GRAVES, A.C., LAMAR, B.W. An integer programming procedure for assembly system design problems. *Operations Research*. Vol. 31, 1983, p.522-545.
- HACKMAN, S.T., MAGAZINE, M.J., WEE, T.S. Fast, effective algorithms for simple assembly line balancing problems. *Operations Research*. Vol. 37 (6), 1989, p.916-923.
- HE, D.W., KUSIAK, A. Design of Assembly Systems for Modular Products. *IEEE Transactions on Robotics and Automation*. Vol. 13 (5), 1997.
- HELD, M., KARP, R.M., SHARESHIAN, R. Assembly line balancing-dynamic programming with precedents constraints. *Operations Research*. Vol. 11, 1963, p.442-459.
- HELGESON, W.B., BIRNIE, D.P. Assembly line balancing using Ranked Positional Weight Techniques. *Journal of Industrial Engineering*. Vol. 12, 1961, p.394-398.
- HILLIER, F.S., LIEBERMAN, G.J. *Introducción a la Investigación de Operaciones*. Ed. McGraw-Hill, 6th edition, 1995.
- HOFFMAN, T.R. Assembly line balancing with precedente matriz. *Management Science*. Vol. 9, 1963, p.551-562.
- HOFFMAN, K.L. Improving LP-Representations of zero-one linear programs for branch and bound. *Orsa Journal of Computing*, 1991.
- HOFFMAN, T.R. EUREKA: a hybrid system for assembly line balancing. *Management Science*. Vol. 38 (1), 1992.
- JOHNSON, R.V. A branch and bound algorithm for assembly line balancing with formulation irregularities. *Management Science*. Vol. 29, 1983, p.1309-1324.
- JOHNSON, R.V. Optimally balancing large assembly lines with FABLE. *Management Science*. Vol. 34, 1988, p.240-253.
- KIM, Y.K., KIM, Y.J., KIM, Y. Genetic algorithms for the assembly line balancing with various objectives. *Computers Industrial Engineering*. Vol. 30 (3), 1996, p.397-409.
- KLEIN, R., SCHOLL, A. Maximizing the production rate in simple assembly line balancing. A branch and bound procedure. *European Journal of Operations Research*. Vol. 91, 1996, p.367-385.



- KLEIN, R., SCHOLL, A. SALOME: A bidirectional branch & bound procedure for assembly line balancing. *INFORMS Journal on Computing*. Vol. 9, 1997, p.319-334.
- Mc.CLAIN, J.O., MOODIE, D.R. A comment on buffer space allocation in automated assembly lines. *Operations Research*. Vol. 39 (5), 1991.
- MILTENBURG, J., WIJNGAARD, J. The U-line balancing problem. *Management Science*. Vol. 40 (10), 1994.
- MILTENBURG, J. Balancing U-lines in a multiple U-line facility. *European Journal of Operational Research*. Vol. 109, 1998, p.1-23.
- MIYAZAKI, S., OHTA, H. An optimal solution of the assembly line balancing problem with the balancing index as a primary objective function. *Collection: System Modelling and Optimization*. 1987, p.681-690.
- MOODIE, C.L., YOUNG, H.H. A heuristic method of assembly line balancing for assumptions of constant or variable work element times. *Journal of Industrial Engineering*. Vol. 16, 1965, p.23-29.
- MOON, C., LEE, M., SEO, Y., LEE, Y.H. Integrated machine tool selections and operation sequencing capacity and precedence constraints using genetic algorithm. *Computers and Industrial Engineering*. Vol. 43, 2002, p.605-621.
- NICOSIA, G., PACCIARELLI, D., PACIFICI, A. Optimally balancing assembly lines with different workstations. *Discrete Applied Mathematics*. Vol. 118, 2002, p.99-113.
- PARK, K., PARK, S., KIM, W. A heuristic for an assembly line balancing problem with incompatibility, range and partial precedence constraints. *Computers Industrial Engineering*. Vol. 32 (2), 1997, p.321-332.
- PASTOR, R., COROMINAS, A. Assembly line balancing with incompatibilities and bounded workstation loads. *Journal of the Italian Operations Research Society*. Vol. 30 (93), 2000, p.23-45.
- PASTOR, R., ANDRES, C., DURAN, A., PEREZ, M. Tabu search algorithms for an industrial multi-product, multi-objective assembly line balancing problem, with reduction of task dispersion. *Journal of Operational Research Society*. Vol. 53, 2002, p.1317-1323.
- PASTOR, R. Resolución del problema del equilibrado de líneas de montaje con incompatibilidades: aplicación a una empresa japonesa de montaje de motos. PFC. Departament d'Organització d'Empreses. ETSEIB, UPC. Barcelona, 1994.



- PASTOR, R., ANDRES, C., DURAN, A., PEREZ, M. Tabu search algorithms for an industrial multi-product and multi-objective assembly line balancing problema, with reduction of the task dispersion. *Journal of Operational Research Society*. Vol. 53, 2002, p. 1317-1323.
- PASTOR, R. Resolución del problema multi-min-max en el equilibrado de líneas de montaje. Documento de Trabajo IOC-DT-I-2004-03. Instituto de Organización y Control de Sistemas Industriales, UPC. Octubre 2004.
- PASTOR, R. LB-ALBP: The lexicographic bottleneck assembly line balancing problem. Documento de trabajo IOC-DT-P-2007-10. Instituto de Organización y Control de Sistemas Industriales, UPC. Septiembre 2007.
- PASTOR, R., FERRER, L. An improved mathematical program to solve the simple assembly line balancing problem. *International Journal of Production Research* (to be published, doi: 10.1080/00207540701713832). 2008.
- PHONGANANT, S., YANG, Y.N., LEEP, H.R., PARSEI, H.R. Expert System for Mixed-Model Assembly Line Balancing. Dallas, 10th Annual Industrial Engineering Research Conference, 2001.
- PINNOI, A., WILHEM, W. Assembly System Design: a Branch and Cut approach. *Management Science*. Vol. 44 (1), 1998.
- PINTO, P.A., DANNENBRING, D.G., KHUMAWALA, B.M. Assembly line balancing with processing alternatives: an application. *Management Science*. Vol. 29 (7), 1983.
- PLANS, J. Classificació, modelització i resolució dels problemes de disseny i assignació de tasques en línies de producció. Tesis Doctoral, UPC. 1999.
- PONNAMBALAM, S.G., ARAVINDAN, P., MOGILEESWAR NAIDU, G. A comparative evaluation of assembly line balancing heuristic. *The International Journal of Advance Manufacturing Technology*. Vol. 15, 1999, p.577-586.
- PONNAMBALAM, S.G., ARAVINDAN, P., SUBBA RAO, M. Genetic algorithms for sequencing problems in mixed model assembly lines. *Computer & Industrial Engineering*. Vol. 45 (4), 2003, p.669-690.
- REKIEK, B., DOLGUI, A., DELCHAMBRE, A., BRATCHU, A. State of art of optimization methods for assembly line design. *Annual Reviews in Control*. Vol. 26, 2002, p.163-174.



- RUBINOVITZ, J., BUKCHIN, J. RALBP – A heuristic algorithm for design and balancing of robotic assembly lines. *Anal. of the CIRP*. Vol. 42, 1993, p.497-500.
- RUBINOVITZ, J., LEVITIN, G. Genetic algorithm for assembly line balancing. *International Journal of Production Economics*. Vol. 41, 1995, p.343-354.
- SARKER, B.R., HARRIS, R.D. The effect of imbalance in a just-in-time production system. *International Journal of Production Research*. Vol. 26 (1), 1988.
- SARIN, S., EREL, E., DAR-EL, E. A methodology for solving single-model, stochastic assembly line balancing problem. *International Journal of Management Science*. 1999, p.525- 535.
- SCHOLL, A. *Balancing and sequencing of assembly lines*. Heidelberg: Physica-Verlag, 1999.
- SCHOLL, A., VOSS, S. Simple assembly line balancing - heuristic approaches. *Journal of Heuristics*. Vol. 2, 1996, p.217-244.
- SCHOLL, A., KLEIN, R. SALOME: a bidirectional branch and bound procedure for assembly line balancing. *Inform. Journal on Computing*. Vol. 9, 1997, p.319-334.
- SCHOLL, A., KLEIN, R. ULINO: optimally balancing U-shaped JIT assembly lines. *International Journal of Production Research*. Vol. 37 (4), 1999, p.721-736.
- SCHOLL, A., BECKER, C. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*. Vol. 168, 2006, p.666-693.
- SENTINELLA, C., SOLSONA, J. Exposición y resolución del caso Multi-Min-Max en el problema de equilibrado de líneas de Montaje. PFC. Escola Tècnica Superior d'Enginyeria Industrial de Barcelona. Universitat Politècnica de Catalunya. Noviembre 2005.
- SPINA, R., GALANTUCCI, M., DASSISTI, M. A hybrid approach to the single line scheduling problem with multiple products and sequence-dependent time. *Computers & Industrial Engineering*. Vol. 45 (4), 2003, p.573-583.
- SUER, G. Designing parallel assembly lines. *Computers in Industrial Engineering*. Vol. 35 (3,4), 1998, p.467-470.
- SURESH, G., SAHU, S. Stochastic assembly line balancing using simulated annealing. *International Journal of Production Research*. Vol. 32, 1994, p.1801-1810.



- TALBOT, F.B., PATTERSON, J.H. An integer programming algorithm with network cuts for solving the assembly line balancing problem. *Management Science*. Vol. 30 (1), 1984, p.85- 99.
- TALBOT, F.B., PATTERSON, J.H., GEHRLEIN, W.V. A comparative evaluation of heuristic line balancing techniques. *Management Science*. Vol. 32 (4), 1986, p.430-454.
- THANGAVELU, S.R., SHETTY, C.M. Assembly line balancing by zero-one integer programming. *AIIE Transactions*. Vol. 3, 1971, p.61-68.
- TONGHE, F.M. Summary of a heuristic line balancing procedure. *Management Science*. Vol. 7, 1960, p.21-39.
- URBAN, T.L. Optimal balancing of U-shaped assembly lines. *Management Science*. Vol. 44, 1998, 738-741.
- VALERO, J. Modelos y algoritmos de PLE para el equilibrado de líneas de producción y montaje. PFC, Departamento de Organización de Empresas. ETSEIB-UPC. Barcelona,1991.
- VILARINHO, P., SIMARIA, A. A two stage heuristic methods for balancing mixed-model assembly lines with parallel workstations. *International Journal of Productions Research*. Vol. 40 (6), 2002, p.1405-1420.
- VOSS, S. Tabu search in manufacturing. *Operational Research Proceedings*. 1993, p.183-194.
- WHITE, W.W. Comment on a paper by Bowman. *Operations Research*. Vol. 9, 1961, p.274-276.

