

# Una heurística para el Equilibrado de Líneas de Montaje con Número Fijo de Estaciones

Joaquín Bautista<sup>1</sup>, Jordi Pereira<sup>1</sup>

*Resumen*--El presente artículo se centra en el estudio del problema de equilibrado de líneas de montaje simple con un número fijo de estaciones, problema conocido en la literatura como SALBP-2. Para su resolución se propone una aplicación de la heurística “Filter and Fan”, una propuesta reciente que puede verse como una generalización de los procedimientos de búsqueda local con profundidad variable. Tras introducir el problema y la literatura incumbente al presente estudio, se detallan los principios básicos de la metaheurística “Filter and Fan” y la implementación propuesta para el problema. Posteriormente se muestran los resultados del procedimiento en una experiencia computacional preliminar, comparando los resultados obtenidos con los mejores presentes en la literatura, para finalizar con algunas conclusiones que se desprenden de los resultados obtenidos.

*Palabras clave*—Líneas de Montaje, Producción, Búsqueda Local.

## 1. INTRODUCCIÓN

El problema de equilibrado de líneas de montaje (ALBP) es un problema académico clásico que ha sido tratado al menos durante los últimos 50 años, [11] y [18]. Simplificadamente, equilibrar una línea de montaje consiste en asignar las tareas en que se divide el montaje de un producto entre las diferentes estaciones que componen la línea en la que se ensambla. Dichas estaciones se disponen, típicamente, en serie y el producto pasa de una estación a la siguiente cuando el tiempo concedido para elaborar las tareas asignadas a la estación finaliza. El tiempo concedido, idéntico para cada una de las estaciones, se conoce como tiempo de ciclo y su inversa corresponde con la tasa de producción de la línea de montaje en su conjunto. Los problemas clásicos tratados en la literatura, conocidos como problemas de equilibrado simples, buscan encontrar una solución factible a dos o tres de los siguientes conjuntos de restricciones:

- (1) restricciones asociadas al tiempo disponible en cada estación para trabajar sobre el producto;
- (2) restricciones asociadas al número de estaciones disponibles en la línea de montaje;

(3) restricciones asociadas a que ciertas operaciones deben realizarse tras la elaboración de otras, llamadas restricciones de precedencia; Convirtiendo el otro elemento en el objeto a optimizar, aunque en los casos reales sería necesario tener en cuenta un mayor número de restricciones, y, en ocasiones, otras funciones objetivo.

Sea cual sea la función utilizada, el objetivo buscado es la minimización del tiempo muerto en la línea, ya sea minimizando el número de estaciones necesarias, restricciones (1) y (3) con objetivo (2), o el tiempo concedido a cada estación, restricciones (2) y (3) con objetivo (1). También puede plantearse un problema de factibilidad dado un tiempo concedido y un número de estaciones. Según la nomenclatura utilizada en este tipo de problemas, [3] y [19], estos problemas se conocen como SALBP-1, SALBP-2 y SALBP-F respectivamente.

La literatura se ha centrado en el caso SALBP-1, excepto en raras ocasiones. Aparte de motivaciones históricas, ya que la formulación SALBP-1 fue la primera tratada, la fuerte interrelación entre los objetivos y restricciones de cada uno de los problemas y el mayor desarrollo algorítmico para la resolución del caso, han hecho que los casos SALBP-2 y SALBP-F se hayan tratado mediante la resolución de una o más instancias equivalentes tipo SALBP-1. Aún así, a nivel práctico, las formulaciones SALBP-1 y SALBP-2 o SALBP-F cubren versiones reducidas de aspectos diferentes del diseño de líneas de montaje. Simplificadamente, aun a costa de perder gran parte de los matices reales que deben tenerse en cuenta durante el equilibrado de una línea de montaje real, la minimización del número de estaciones, caso SALBP-1, intenta describir una fase de diseño de línea, mientras que la minimización del tiempo de ciclo, o la búsqueda de la factibilidad, se asocian a las decisiones de reingeniería y rediseño de una línea ya establecida. Si bien a nivel teórico puede ser adecuado tratarlo tal como se ha realizado hasta el momento en la literatura, ya que el desarrollo de procedimientos para el caso SALBP-1 es mayor, a nivel práctico los problemas y necesidades reales, y la más que necesaria modificación de los procedimientos usados para resolver problemas académicos cuando se intenta tratar las características reales de un problema industrial, nos lleva a pensar que algoritmos que se planteen directamente el objetivo propio del rediseño de

<sup>1</sup> Càtedra Nissan. Universitat Politècnica de Catalunya, Avda. Diagonal 647, 08028 Barcelona. E-mail: joaquin.bautista@upc.edu, jorge.pereira@upc.edu

líneas, el caso mas normal en el ciclo de vida de una línea de montaje, tengan ventajas para su uso real, como la posibilidad de tratar otros objetivos y restricciones específicas, aunque sean menos competitivos ante la resolución de instancias de la literatura.

El resto de este trabajo se estructura tal como sigue. En la sección II se estudia la formulación del problema, mientras la sección III presenta la bibliografía relativa al equilibrado de líneas de montaje. La sección IV se dedica a mostrar los diferentes componentes básicos procedentes de la literatura, procedimientos constructivos y de mejora local, en los que se basará la propuesta original del presente trabajo. La sección V está destinada a describir el algoritmo implementado y a mostrar las características más importantes de éste. La sección VI muestra los resultados obtenidos en una experiencia computacional con instancias de la literatura y finalmente la sección VII muestra las conclusiones del presente trabajo y las líneas de desarrollo futuro.

## II. FORMULACIÓN DEL PROBLEMA

Formalmente, una instancia del problema de equilibrado de líneas de montaje SALBP-2 queda definida por un conjunto de tareas  $V$ , en que se ha subdividido el montaje de un producto. Cada tarea debe asignarse a una de las  $S_j$ , ( $j=1, \dots, m$ ), estaciones disponibles. Cada una de las tareas tiene una duración determinista y conocida,  $d_i$ , ( $i=1, \dots, |V|$ ) que, sin pérdida de generalidad, puede considerarse un valor entero. Adicionalmente, algunas tareas presentan relaciones de precedencia entre ellas, por ejemplo es necesario montar los asientos de un coche antes de instalar las puertas, que pueden representarse mediante un grafo acíclico  $G(V,A)$  donde existe un arco entre la tarea  $i$ , ( $i=1, \dots, |V|$ ), y la tarea  $k$ , ( $k=1, \dots, |V|$ ), si la tarea  $i$  debe realizarse con anterioridad a la tarea  $k$ , esto es si  $i \in S_j$  y  $k \in S_l$ , se debe cumplir  $j \leq l$ .

El objetivo del problema es encontrar una asignación de tareas que minimice el tiempo idéntico concedido a cada estación, conocido como tiempo de ciclo  $c$ , que fija la tasa de producción de la línea de montaje, respetando las siguientes restricciones:

- (1)  $\sum_{i \in S_j} d_i \leq c$ , para todo  $S_j$ , ( $j=1, \dots, m$ ).
- (2) Si  $(i,k) \in G(V,A)$ ,  $i \in S_j$  y  $k \in S_l$  entonces debe cumplirse que  $j \leq l$ .
- (3)  $i \in S_1 \cup \dots \cup S_m$ , para todo  $i$ , ( $i=1, \dots, |V|$ ).
- (4)  $S_j \cap S_l = \emptyset$ , para todo  $j \neq l$ .

Dónde el conjunto de restricciones (1) indica las restricciones asociadas al tiempo concedido en cada estación, el conjunto (2) indica las restricciones de precedencia entre tareas, las restricciones (3) obligan a que cada tarea se asigne al menos a una estación y el conjunto (4) asegura que ninguna tarea se asigna a más de una estación.

De la formulación puede verse que el problema equivale a un problema de Bin Packing, BP-2 siguiendo la nomenclatura de Martello y Toth, [15], al que se le añaden las restricciones de precedencia, conjunto de restricciones (2).

Por otra parte el problema es reversible, esto es, si el sentido de los arcos del grafo de precedencias  $G(V,A)$  se invierte, cualquier solución obtenida para la nueva instancia puede transformarse en una solución para la instancia original cambiando la ordenación de las estaciones obtenidas,  $S_j \leftarrow S_{m+1-j}$ , para todo  $S_j$ , ( $j=1, \dots, m$ ). Esta propiedad se utiliza en el presente trabajo, así como en la mayoría de trabajos anteriores, ya que se conoce que el comportamiento de los procedimientos de resolución varía según se resuelva la instancia directa o inversa.

## III. ESTADO DEL ARTE

La literatura incluye un gran número de propuestas para resolver el problema de equilibrado de líneas de montaje simple, véase el reciente estado del arte de Scholl y Becker, [20], así como trabajos de descripción y resolución de problemas mas generales, véase [4], o la clasificación de Boysen, Fliedner y Scholl, [6]. El presente apartado intenta recoger únicamente aquellos trabajos que resultan de interés para el presente estudio, incluyendo los procedimientos para el SALBP-2, en los que la literatura es mucho mas reducida.

Los intentos de resolución de problemas de equilibrado de líneas de montaje pueden dividirse en tres tipologías principales según el procedimiento utilizado:

(1) Un primer grupo formado por procedimientos constructivos, que utilizan una o varias reglas de prioridad estáticas o dinámicas para asignar las tareas a estaciones cumpliendo con las reglas de prioridad y un tiempo de ciclo dado para cada estación, véase [22], o bien basados en algún tipo de enumeración truncada, [10], o metaheurísticas de tipo constructivo, [1] y [5]. Todos ellos utilizan como uno de los parámetros del algoritmo el tiempo de ciclo asignado a las estaciones, por lo que la aplicación de estos procedimientos al caso tratado en el presente trabajo, en que el tiempo de ciclo es el objetivo a minimizar, obliga a usar el procedimiento de forma iterativa hasta encontrar una solución que cumpla con la limitación en el número de estaciones fijado para el problema, ya que el parámetro, el tiempo de ciclo, es el objetivo buscado.

(2) Un segundo grupo formado por procedimientos de enumeración, basados generalmente en procedimientos de exploración arborescente tipo *Branch and Bound*, [19], o de exploración de grafos tipo Programación Dinámica, [2], que son los que han ofrecido los mejores resultados con instancias de la literatura. En la

mayoría de ocasiones, al igual que con el primer tipo de procedimientos, la aplicación a problemas con número fijo de estaciones se centran en la resolución iterativa de problemas con tiempo de ciclo fijo hasta encontrar una solución con el número de estaciones deseado, aunque existe al menos una aproximación directa al problema tratado, [12].

(3) Finalmente, el grupo formado por procedimientos de búsqueda local, y las metaheurísticas basadas en la búsqueda local. Entre las propuestas anteriores cabe remarcar dos trabajos específicamente dirigidos al caso SALBP-2, la búsqueda tabú propuesta por Scholl y Voss, [21] y un procedimiento de mejora basado en la programación lineal de Urdurdag et al., [23], cuyos resultados distan de aquellos obtenidos por los mejores procedimientos de las dos tipologías de algoritmo anteriores.

#### IV. EL PROBLEMA DE EQUILIBRADO DE LÍNEAS DE MONTAJE SALBP-2

Las dos siguientes subsecciones se dedican a estudiar los elementos procedentes de la literatura que serán utilizados para el desarrollo de la heurística propuesta.

##### A. Construcción de soluciones

El procedimiento de construcción de soluciones iniciales se basa en el procedimiento de Hoffman, [10]. Este procedimiento resuelve una instancia SALBP-1 con un tiempo de ciclo dado, aplicando un algoritmo de enumeración en que se busca una asignación de tareas a una estación con el menor tiempo libre posible, la asignación más satisfactoria siguiendo una técnica "greedy". Tras encontrar la asignación satisfactoria para una estación, se vuelve a aplicar el mismo procedimiento con el problema remanente, esto es se construye una nueva estación con una nueva asignación satisfactoria, teniendo en cuenta sólo aquellas tareas que no han sido previamente escogidas para formar parte de una estación previa, hasta que no quedan más tareas por asignar, en cuyo caso se ha encontrado una solución para la instancia.

Aunque el procedimiento se basa en la enumeración de todas las asignaciones posibles para una estación, y por tanto en el peor de los casos puede llevar a requerir tiempo exponencial, en la práctica, véase [2] y [7], el algoritmo no sufre de este efecto, obteniendo mejores resultados que las heurísticas basadas en reglas de prioridad.

##### B. Búsqueda local

Los procedimientos de mejora para los problemas de equilibrado se basan en operadores de intercambio y movimiento, véase [16]. Para explicar

estos operadores es necesario definir una serie de conceptos.

Dada una solución del caso SALBP-2, se dice que una estación  $j$  es crítica en la solución cuando es la estación con mayor carga de trabajo, esto es:  $\sum_{i \in s_j} d_i = c$ .

Dada una solución y para cada tarea  $i$ , se define la primera,  $ES_i$ , y la última,  $LP_i$ , estación en las que una tarea  $i$  puede ser asignada en consonancia con la asignación de sus predecesoras y sucesoras inmediatas.

El conjunto de tareas candidatas a mejorar una solución del SALBP-2 está formado por aquellas tareas que forman parte de una estación crítica, y por tanto éstas son las únicas tareas que deben ser tenidas en cuenta durante el procedimiento de mejora local.

Un movimiento  $(i, j, l)$  describe el cambio de asignación de la tarea  $i$  en la estación  $j$  a la estación  $l$ , donde  $j \neq l$  y  $l \in [ES_i, LP_i]$ , mientras que un intercambio  $(i, j, k, l)$  intercambia la asignación de las tareas  $i$  y  $k$ , no relacionadas por precedencia, siendo el intercambio factible si los movimientos  $(i, j, l)$ ,  $(k, l, j)$  lo son individualmente.

Para que un movimiento o un intercambio mejore la solución en curso, debe existir una sola estación crítica en la solución en curso y la nueva carga de trabajo en las estaciones involucradas en el cambio debe ser menor que el tiempo de ciclo de la solución original. Como es posible que una solución tenga más de una estación crítica, normalmente se permiten movimientos e intercambios, que aunque no mejoren la solución en su conjunto, mejoren la carga de trabajo de las estaciones involucradas en las operaciones.

Para reducir un poco más el vecindario de exploración, la implementación realizada escoge una única estación crítica y se comprueban todos los posibles movimientos e intercambios de las tareas que conforman dicha estación. En caso de existir más de una estación crítica, la estación crítica seleccionada para los intercambios se escoge mediante una regla que mide la 'distancia' mínima entre una estación crítica y la estación con mayor holgura.

Se entiende como estación con mayor holgura a aquella estación que tiene la mayor diferencia entre la carga asignada y el tiempo de ciclo de la solución en curso, resolviendo empates según este criterio de forma aleatoria. Al estar las estaciones ordenadas,  $1, \dots, m$ ; la distancia entre dos estaciones  $j, l$ , se mide a través de la diferencia en el orden  $|l-j|$ .

#### V. UNA APLICACIÓN DE LA HEURÍSTICA "FILTER AND FAN" AL PROBLEMA

En la presente sección, tras comentar brevemente las bases de la heurística utilizada en el trabajo, se pasa a describir los elementos del algoritmo implementado.

La idea central de la heurística “Filter and Fan” es tratar la búsqueda local como un procedimiento de exploración arborescente.

El árbol asociado a la búsqueda local, al contrario que en un procedimiento *Branch and Bound*, tiene como vértice raíz una solución válida del problema. Cada uno de los arcos que emergen de un vértice representa la aplicación de un operador de búsqueda local, para este problema un movimiento o un intercambio en la solución representada por el vértice origen, que lo transforma en un vértice destino que, en el caso tratado, representa obligatoriamente una solución factible del problema. Un camino entre el vértice raíz y otro vértice del árbol representa un operador compuesto de búsqueda local que, en caso que el vértice destino tenga mejor valor de función objetivo para el problema, lleva a una mejora de la solución en curso.

La idea básica de este planteamiento puede trazarse en la heurística Lin-Kernighan para el problema del viajante de comercio (TSP), [13]. En el caso del viajante del comercio los intercambios corresponden a modificar uno o más de los arcos que componen la solución. Un intercambio básico, consistente en intercambiar dos arcos, denominado 2-opt, compone el primer nivel del árbol de búsqueda, el segundo nivel se compone por intercambios de tres arcos, 3-opt, que pueden componerse mediante dos movimientos tipo 2-opt, y así sucesivamente.

Bajo este planteamiento, la heurística “Filter and Fan” es un procedimiento de exploración del árbol de búsqueda local de profundidad variable en que se construye y explora el árbol siguiendo una política primero en anchura, en contraposición con la heurística Lin-Kernighan que es un procedimiento con una política primero en profundidad. La heurística fue propuesta inicialmente por Glover, [8], como método de mejora para la metaheurística Scatter Search, y posteriormente extendido por Rego y Glover, [17]. Otra manera de ver la heurística es como una combinación de las estrategias existentes para las listas de candidatos de la búsqueda tabú, [9].

Lo que define y diferencia a la heurística “Filter and Fan” es su política de exploración del árbol con el objetivo de limitar la explosión exponencial de vértices que deberían ser explorados.

El uso de este tipo de metaheurística nos llevará a la necesidad de establecer, en primer lugar, cómo obtener una solución inicial para partir desde ella en el proceso de exploración, a lo que se dedicará la subsección IV.C, el método de construcción del árbol de exploración y cómo se reducirá su tamaño, subsección IV.D, y finalmente a algunos mecanismos implementados para introducir un factor de diversificación en la búsqueda en la subsección IV.E.

### C. Generación de soluciones iniciales

Tal como se ha mencionado en la subsección III.A, la construcción de soluciones para el problema debe realizarse mediante la resolución iterativa de problemas SALBP-1 con tiempos de ciclo cada vez superiores hasta encontrar una solución con el número de estaciones deseado. En este caso se utiliza una adaptación del algoritmo de Hoffmann mostrado en la subsección III.A.

La búsqueda de una solución con el número de estaciones deseado se inicia con un tiempo de ciclo tentativo  $c'$ , que corresponde al valor mínimo que puede adoptar si la suma de tiempo libre de las estaciones fuera mínimo, tal como se puede obtener mediante (5).

$$c' = \left\lceil \frac{\sum_{1 \leq i \leq n} p_i}{m} \right\rceil \quad (5)$$

En caso que el número de estaciones que obtiene el procedimiento sea superior al buscado, se aumenta el tiempo de ciclo tentativo  $c'$  en uno y se vuelve a ejecutar el procedimiento constructivo.

### D. Árbol de exploración y reducción del árbol

Partiendo de los operadores clásicos de exploración de entornos para el problema de equilibrado de líneas de montaje expuestos en la subsección III.B, la aproximación desarrollada basada en la heurística Filter and Fan extiende y generaliza ambos tipos de vecindarios mediante el uso de movimientos compuestos. Gráficamente, la heurística puede ilustrarse a través de un árbol de exploración donde las ramas representan la aplicación de alguno de los operadores mostrados anteriormente y los vértices del árbol representan las soluciones producidas por dichos operadores. Debe realizarse una excepción con el vértice raíz del árbol que representa una solución inicial. El número máximo de niveles considerados en una secuencia define la profundidad del árbol. El método construye el árbol según lo explora mediante la ejecución de operadores a cada uno de los vértices. Cada operador aplicado genera implícitamente un vértice sucesor en el árbol. Como los operadores aplicados a un vértice se trasmite a todos sus sucesores, el método puede considerarse adaptativo porque el resultado de cada operador depende del estado de la búsqueda. El método también es dinámico ya que el número de operadores para construir el vecindario compuesto depende del nivel del árbol en que se ha encontrado una nueva solución inicial, que varía de iteración en iteración.

Como el número de vértices del árbol de exploración podría llegar a ser prohibitivamente elevado, la construcción de vértices se limita mediante el uso de listas restringidas de candidatos y el filtrado de vértices construidos debido al valor

de función objetivo de tal forma que aquellas soluciones que parecen más prometedoras son utilizadas para la generación de un nuevo nivel. Esta técnica de reducción es parecida a la metaheurística Beam Search aunque en este caso no se base en un sistema constructivo.

En la implementación realizada, el procedimiento empieza con una búsqueda local estándar descrita en la subsección III.B hasta encontrar un óptimo local, siendo esta solución el vértice raíz del árbol.

Los siguientes niveles se crean como sigue. Sea  $x$  el último nivel desarrollado del árbol y  $\eta$  el número de vértices que forman parte del conjunto  $M(x)$  de soluciones que forman parte del nivel  $x$  del árbol. Para cada solución, se identifica la estación crítica según la definición de la subsección III.B, y se generan todas las soluciones asociadas a los operadores de búsqueda local partiendo de tareas asignadas a dicha estación, y se evalúa el nuevo tiempo de ciclo de cada solución. Esta fase corresponde, metafóricamente, con la generación de un nuevo abanico de soluciones, de esto viene el 'fan' o abanico del nombre de la metaheurística. Al contrario que en otras propuestas de la metaheurísticas, el presente trabajo no limita el número de soluciones generadas desde un vértice.

Posteriormente se aplica un 'filtrado' a las soluciones generadas, el 'filter' o filtro del nombre de la metaheurística, consistente en mantener en el árbol de exploración un número de vértices, soluciones, igual a  $\eta_{max}$ , un parámetro del algoritmo, formado por el conjunto de mejores soluciones según el valor de función objetivo, que además no correspondan a soluciones desarrolladas en etapas anteriores del árbol. Las soluciones que pasan el filtro componen los vértices del nivel  $x+1$  del árbol de búsqueda. El procedimiento se repite para el siguiente nivel excepto si se ha generado una solución con valor mejor que el de partida, en cuyo caso esta solución pasa a ser el nuevo vértice raíz del árbol de exploración, o se llega a un máximo de profundidad en la búsqueda  $x_{max}$ , en cuyo caso se para la búsqueda.

Cabe destacar que una búsqueda local basada en los operadores expuestos en la subsección III.B equivale al procedimiento filter and fan con  $x_{max}=1$ , sólo se aceptan movimientos de mejora generados por la aplicación de un único operador de búsqueda local.

#### E. Diversificación de la búsqueda

El requerimiento del procedimiento de mantener las soluciones factibles a lo largo del árbol de búsqueda omite el tratamiento de uno de los elementos más importantes del problema, las restricciones de precedencia. El procedimiento de mejora puede ser incapaz de tratar las dificultades derivadas de respetar dichas restricciones. Es por

ello que se ha añadido un elemento de diversificación, que intenta encontrar nuevas soluciones iniciales para volver a lanzar la metaheurística manteniendo, parcialmente, características de las soluciones encontradas. El procedimiento desarrollado se centra en las ideas básicas de la búsqueda local iterada, Iterated Local Search [14]. Partiendo de la solución óptima local, se altera la solución para volver a aplicar el procedimiento de búsqueda local.

Para ello se ha modificado parcialmente el procedimiento de Hoffmann, subsección III.A y IV.C del presente trabajo. Tras alcanzar una solución óptima local por el árbol de búsqueda, un subconjunto de tareas de la solución, formado por aquellas que forman parte de estación crítica y de un número de estaciones contiguas escogidas aleatoriamente, en los experimentos realizados un número igual a 4, son reasignadas como si constituyeran una instancia separada del problema. Se entiende por conjunto de estaciones contiguas a aquél conjunto de estaciones formado por estaciones tales que para cada estación la distancia con alguna estación del conjunto es 1.

Tras construir una solución al subproblema con un número de estaciones idéntico a las estaciones reasignadas, éstas se reintroducen en la solución original. La nueva solución puede, incluso, mejorar la solución de partida y genera diversidad en la búsqueda, permitiendo volver a aplicar la metaheurística de búsqueda.

Este paso se repite un número máximo de veces marcado como un parámetro del procedimiento e igual a  $restarts-1$ , e indicado en la experiencia computacional.

## VI. EXPERIENCIA COMPUTACIONAL

Para comprobar la calidad del algoritmo propuesto se ha implementado el algoritmo en C++ utilizando el compilador GCC versión 4.3. El ordenador utilizado en esta experiencia computacional ha sido un Macintosh imac con un procesador 2.33 Ghz. Intel Core 2 Duo y 3 GB. de memoria RAM utilizando MAC OS X 10.4.11 como sistema operativo. Para hacer uso de la propiedad de inversibilidad de las instancias que se ha expuesto en sección IV, el algoritmo resuelve cada instancia en sentido directo e inverso de forma separada, anotándose en la presente experiencia computacional el valor de la mejor solución obtenida por ambos procedimientos.

Las instancias utilizadas en la experiencia computacional corresponden a la colección disponible en la página web [www.assembly-line-balancing.de](http://www.assembly-line-balancing.de), y que son las instancias utilizadas en la mayoría de trabajos de la literatura para comparar la calidad de las soluciones ofrecidas por los procedimientos propuestos. Esta colección cuenta con 302 instancias del problema SALBP-2 con un

número de tareas entre 29 y 297, de las que se conoce la solución óptima para 287 de ellas.

El algoritmo presentado se compara con las mejores soluciones conocidas en la literatura, para diferentes juegos de parámetros del algoritmo,  $\eta_{max}=(1, 10, 50, 100)$ ,  $x_{max}=(1, 10, 20)$  y  $restarts=(1,5,10)$ .

La TABLA I reporta los resultados obtenidos, indicando el número de instancias para la que se obtiene la mejor solución conocida y la desviación media respecto a los mejores valores conocidos. La desviación para cada instancia se ha calculado mediante  $(obtenido-mejor)/mejor \cdot 100$ , dónde obtenido corresponde al tiempo de ciclo reportado por el algoritmo y mejor corresponde al mejor valor conocido en la literatura. La última fila reporta los resultados procedentes de la ejecución de la versión de SALOME, [19], disponible en la página web [www.assembly-line-balancing.de](http://www.assembly-line-balancing.de)

TABLA I

RESULTADOS OBTENIDOS EN LA EXPERIENCIA COMPUTACIONAL. PARA CADA VALOR DE LOS PARÁMETROS DEL ALGORITMO, SE REPORTA EL NÚMERO DE SOLUCIONES ÓPTIMAS ENCONTRADAS Y LA DESVIACIÓN MEDIA. TAMBIÉN SE REPORTA EL RESULTADO OBTENIDO POR SALOME CON LÍMITE DE TIEMPO 3600 s.

$\eta_{MAX}$	$X_{MAX}$	STARTS	#OPT	DESV.
1	1	1	131	0,72
10	10	1	155	0,52
10	10	5	157	0,49
10	10	10	158	0,45
50	10	1	151	0,54
50	10	5	162	0,45
50	10	10	167	0,41
50	20	1	144	0,6
50	20	5	162	0,45
50	20	10	168	0,39
100	10	1	153	0,53
100	10	5	163	0,42
100	10	10	167	0,4
100	20	1	154	0,52
100	20	5	164	0,41
100	20	10	169	0,38
SALOME			277	0.12

Como puede verse, los resultados obtenidos no igualan a los procedentes de las implementaciones anteriores, obtenidos mediante procedimientos Branch and Bound, [19], aunque las desviaciones son bastante reducidas. Sin embargo si se compara el resultado sin búsqueda local, el que corresponde a la terna de parámetros (1,1,1), puede verse que el algoritmo mejora ostensiblemente las soluciones obtenidas, aunque el efecto de aumentar el número de soluciones que se buscan en paralelo, tal como reportan otros autores, por ejemplo en [17], no es el elemento mas importante en la obtención de buenos

resultados para esta metaheurística. Aún así los resultados distan mucho, en número de soluciones óptimas encontradas, de los reportados por procedimientos constructivos.

Es destacable el efecto que tiene la aplicación de los reinicios en la búsqueda. Como ya se comentó en la descripción de dicho elemento, uno de los problemas de los procedimientos de búsqueda local para este tipo de problemas reside en el tratamiento de las relaciones de precedencia. La inclusión dentro del procedimiento de mecanismos que tengan en cuenta esta problemática durante la búsqueda local puede llevar a la mejora del procedimiento.

VII. CONCLUSIONES

El presente trabajo presenta una aplicación de la heurística “filter and fan” al problema de equilibrado de líneas de montaje con un número fijo de estaciones. Al contrario que las propuestas anteriores de la literatura, se opta por una aproximación directa del problema, no basada en la resolución de múltiples instancias con tiempo de ciclo fijo. Los resultados obtenidos, aunque no son comparables con las soluciones procedentes de algoritmos constructivos, muestran que las desviaciones obtenidas son relativamente pequeñas, lo que permite pensar en la aplicabilidad de este tipo de algoritmos en aquellos casos en que se conoce que los procedimientos enumerativos, que requieren del uso de cotas para la obtención de buenos resultados, no sean aplicables.

AGRADECIMIENTOS

Los autores agradecen la colaboración de Nissan Spanish Industrial Operations así como de la cátedra Nissan UPC por financiar parcialmente esta investigación. Este trabajo también ha sido parcialmente financiado por el proyecto DPI2007-63026 del gobierno español.

REFERENCIAS

- [1] Bautista, J., Pereira, J. (2007) Ant algorithms for a time and space constrained assembly line balancing problem, European Journal of Operational Research 177, 2016-2032.
- [2] Bautista, J., Pereira, J. (2008) A dynamic programming based heuristic for the assembly line balancing problem, European Journal of Operational Research, to appear. doi:10.1016/j.ejor.2008.01.016.
- [3] Baybars, I. (1986) A survey of exact algorithms for the simple assembly line balancing problem, Management Science 32, 909-932.
- [4] Becker, C., Scholl, A. (2006) A survey on problems and methods in generalized assembly line balancing, European Journal of Operational Research 168, 694-715.
- [5] Blum, C., Bautista, J., Pereira, J. (2006) Beam-ACO Applied to Assembly Line Balancing, Lecture Notes in Computer Science 4150, 96-107.
- [6] Boysen, N., Flidner, M., Scholl, A. (2007) A classification of assembly line balancing problems, European Journal of Operational Research, 183, 674-693.
- [7] Fleszar, K., Hindi, K.S. (2003) An enumerative heuristic and reduction methods for the assembly line balancing problem, European Journal of Operational Research 145, 606-620.

- [8] Glover F. (1998), A template for scatter search and path relinking. en: Hao J-K, Lutton E, Ronald E, Schoenauer M, Snyers D, eds. Artificial evolution. Lecture notes in computer science, 1363, 3-51.
- [9] Glover, F., Laguna, M. (1997) Tabu search, Kluwer Academic Publishers, Boston.
- [10] Hoffmann, T.R. (1963) Assembly line balancing with a precedence matrix, *Management Science* 9, 551-562.
- [11] Jackson, J.R. (1956) A computing procedure for a line balancing problem, *Management Science* 2, 261-271.
- [12] Klein, R., Scholl, A. (1996), Maximizing the production rate in simple assembly line balancing—A branch and bound procedure, *European Journal of Operational Research* 91, 367-385
- [13] Lin, S., Kernighan, B.W. (1973), An effective heuristic algorithm for the traveling-salesman problem, *Operations Research* 21, 498-516.
- [14] Lourenço, H.R., Martin, O., Stützle, T. (2002), Iterated local search, in F. Glover and G. Kochenberger, eds., *Handbook of Metaheuristics*, 321-353, Kluwer Academic Publishers.
- [15] Martello, S., Toth, P. (1990) *Knapsack problems – Algorithms and computer implementations*, Wiley, New York.
- [16] Rachamadugu, R., Talbot B. (1991) Improving the equality of workload assignments in assembly lines, *International Journal of Production Research* 29, 755-768.
- [17] Rego, C., Glover, F. (2002) Local search and metaheuristics for the traveling salesman problem. en: Gutin, G., Punnen, A., eds., *The traveling salesman problem and its variations*, Kluwer Academic Publishers, Boston, 309-368.
- [18] Salveson, M.E. (1955) The assembly line balancing problem, *Journal of Industrial Engineering* 6, 18-25.
- [19] Scholl, A. (1999) *Balancing and sequencing assembly lines*, 2a edición, Physica, Heidelberg.
- [20] Scholl, A., Becker, C. (2006) State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, *European Journal of Operational Research* 168, 666-693.
- [21] Scholl, A., Voss, S. (1996) Simple assembly line balancing - Heuristic approaches, *Journal of Heuristics* 2, 217- 244.
- [22] Talbot, F.B., Patterson J.H., Gehrlein, W.V. (1986), A comparative evaluation of heuristic line balancing techniques, *Management Science* 32, 430-454.
- [23] Ugurdag, H.F., Rachamadugu, R., Papachristou, C.A. (1997), Designing paced assembly lines with fixed number of stations, *European Journal of Operational Research* 102 , 488-501.