

# Heurísticas constructivas multiobjetivo para el problema de equilibrado de líneas de montaje considerando tiempo y espacio

M. Chica<sup>(1)</sup>, O. Cordon<sup>(2)</sup>, S. Damas<sup>(2)</sup>, J. Bautista<sup>(3)</sup>, J. Pereira<sup>(3)</sup>

*Resumen*— La extensión del tradicional problema industrial de equilibrado simple de líneas de montaje para incluir también el espacio disponible hacen que este problema sea más interesante pues se asemeja más a las condiciones reales de la planta. En este trabajo lo abordamos mediante una nueva propuesta multiobjetivo basada en optimización mediante colonias de hormigas. Mostramos resultados prometedores correspondientes a diez instancias realistas del problema.

*Palabras clave*— Optimización por colonia de hormigas, Optimización multiobjetivo, Líneas de montaje, Metaheurísticas, Producción.

## I. INTRODUCCIÓN

En una línea de montaje, la fabricación de un producto se divide en un conjunto de tareas. Cada una de ellas requiere un tiempo de operación para su ejecución que se determina en función de las tecnologías de producción y los recursos. Un subconjunto de tareas se asigna a cada estación de la planta. Un problema habitual y complejo es determinar qué tareas asignar a las estaciones cumpliendo todas las restricciones.

Una familia de este tipo de problemas es la conocida como Problema Simple de Equilibrado de Línea de Montaje (PSELM<sup>1</sup>). Tomando esta familia como base, Bautista y Pereira propusieron recientemente una subfamilia más realista [2], el Problema de Equilibrado de Línea de Montaje considerando Tiempo y Espacio (PELMTE<sup>2</sup>) que considera una restricción adicional relativa al espacio disponible. Esta variante extendida se ajusta mejor al escenario de la industria del automóvil en el que se centra nuestro trabajo.

Al igual que muchos problemas reales, las formulaciones del TSALBP tienen una naturaleza multicriterio porque contienen tres objetivos en conflicto que deben minimizarse: el tiempo de ciclo de la

línea de montaje, su área y el número de estaciones. En esta contribución hemos seleccionado la variante TSALBP-1/3 que trata de minimizar el número de estaciones y su área dado un tiempo de ciclo del producto. Nuestra elección se basa en criterios de realismo dentro del sector de la automoción. Asimismo, el diseño de un enfoque multiobjetivo para resolver el problema es una de las novedades de esta publicación.

TSALBP-1/3 tiene un conjunto de restricciones importante para cada estación como precedencias entre estaciones o límites de tiempo de ciclo. Por tanto, el uso de metaheurísticas constructivas como la Optimización por Colonias de Hormigas (ACO, en inglés) [10] es más conveniente que otros como los procedimientos de búsqueda local o global [14].

Debido a estas razones, es decir, la naturaleza multiobjetivo del problema y la conveniencia de resolverlo mediante algoritmos constructivos, trabajaremos con un algoritmo ACO multiobjetivo (MOACO) [13]. Esta familia incluye diversas variantes de algoritmos ACO que tratan de hallar no sólo una solución sino un conjunto de las mejores soluciones de acuerdo a varias funciones objetivo en conflicto. Los algoritmos MOACO basados en Pareto se incluyen en esta categoría y parecen ser los más interesantes.

Hemos elegido el Sistema Múltiple de Colonia de Hormigas (MACS, en inglés) [1] para resolver el TSALBP-1/3 por su buen rendimiento en el estudio experimental desarrollado en [13]. Hemos seleccionado cuatro instancias realistas del problema para probar y analizar el rendimiento de nuestra nueva propuesta.

Esta contribución se estructura como sigue. En la Sección 2, se explican con más profundidad el problema y sus variantes. En la Sección 3 se detalla el algoritmo MOACO aplicado (MACS). En la Sección 4, se explica nuestro enfoque MACS para resolver el TSALBP-1/3. Los experimentos desarrollados junto con un análisis del rendimiento del algoritmo se muestran en la Sección 5. Finalmente, la Sección 6 se dedica a exponer las conclusiones finales de este estudio.

<sup>(1)</sup>ITAV Tecnologías Avanzadas Inspiralia, Madrid, ESPAÑA. E-mail: manuel.chica@itav.es.

<sup>(2)</sup>European Centre for Soft Computing, Mieres, ESPAÑA. E-mail: {oscar.cordon, sergio.damas}@softcomputing.es.

<sup>(3)</sup>Càtedra Nissan - Universitat Politècnica de Catalunya, Barcelona, ESPAÑA, E-mail: {joaquin.bautista,jorge.pereira}@upc.edu

<sup>1</sup>Dada su difusión, en este trabajo usaremos el acrónimo inglés: *Simple Assembly Line Balancing Problem (SALBP)*.

<sup>2</sup>En este trabajo usaremos el acrónimo inglés: *Time and Space Assembly Line Balancing Problem (TSALBP)*.

## II. TSALBP Y TSALBP-1/3

### A. Una breve introducción al problema de equilibrio de líneas de montaje

La fabricación de un producto se divide en un conjunto  $V$  de  $n$  tareas. Cada tarea  $j$  requiere un tiempo de operación para su ejecución  $t_j > 0$  que se determina como una función de las tecnologías de producción y los recursos empleados. A cada estación  $k$  ( $k = 1, 2, \dots, m$ ) se asigna un subconjunto de tareas  $S_k$  ( $S_k \subseteq V$ ), llamado su carga.

Cada tarea  $j$  se asignará a una sola estación  $k$  y tendrá un conjunto de tareas predecesoras,  $P_j$ , que deben finalizarse antes de comenzar a abordarla. Estas restricciones se representan normalmente mediante un gráfico de precedencias acíclico, cuyos vértices representan las tareas y en el que un arco dirigido  $arc < i, j >$  indica que la tarea  $i$  debe finalizar antes de comenzar la tarea  $j$  en la línea de producción. Por tanto, si  $i \in S_h$  y  $j \in S_k$ , entonces  $h \leq k$  debe cumplirse. Cada estación  $k$  presenta un tiempo de carga de la estación  $t(S_k)$  que viene dado por la suma de la duración de las tareas asignadas a la estación  $k$ . Una vez que se alcanzan las condiciones de producción permanentes, los productos fluyen a lo largo de la línea a una tasa constante y cada estación  $k$  tiene un tiempo  $c$  llamado tiempo de ciclo, para llevar a cabo las tareas que se le asignaron. Los items se transfieren entonces a la siguiente estación en un período de tiempo insignificante, iniciándose así un nuevo ciclo. El tiempo de ciclo  $c$  determina la tasa de producción  $r$  de la línea ( $r = 1/c$ ) y no puede ser menor que el máximo tiempo de carga de las estaciones:  $c \geq \max_{k=1,2,\dots,m} \{t(S_k)\}$ .

En general, SALBP [21] se centra en la agrupación de tareas del conjunto  $V$  en estaciones de forma eficiente y coherente. Brevemente, el objetivo es alcanzar un grupo de tareas que minimice la ineficiencia de la línea o su período de inactividad satisfaciendo todas las restricciones impuestas sobre las tareas y las estaciones. La literatura incluye una gran variedad de procedimientos de resolución exactos y heurísticos así como metaheurísticas aplicadas al SALBP [20], [22].

### B. TSALBP: necesidad de restricciones de espacio

La necesidad de introducir restricciones de espacio en el diseño de líneas de montaje se basa en tres razones fundamentales:

(1) La longitud de las estaciones de trabajo es limitada en la mayoría de las situaciones. Los trabajadores comienzan su trabajo tan próximos al punto inicial de la estación como sea posible y deben completar sus tareas mientras siguen el producto. Precisan llevar herramientas y materiales a ensamblar en la unidad. En ambos casos, hay restricciones del máximo movimiento de los trabajadores permisible que limita directamente la longitud de la estación y el espacio disponible.

(2) Las herramientas y componentes que se requieren para que sean ensamblados deberían distribuirse a lo largo de ambos lados de la línea. Además, hay ciertas operaciones en automoción que sólo pueden ejecutarse en un lado de la línea, condicionando el espacio donde las herramientas y materiales pueden colocarse. Asimismo, si varias tareas que precisan gran espacio se agrupan la estación sería infactible.

(3) Otra fuente habitual de restricciones espaciales proviene del cambio de productos. Centrándose de nuevo en la industria del automóvil, cuando un modelo es sustituido por otro, se suele mantener la misma planta de producción, que se diseñó teniendo en cuenta el modelo original. Sin embargo, la demanda de espacio adicional para la nueva planta dará lugar a restricciones espaciales.

Basándonos en estas características realistas, surge un nuevo problema realista. Para modelarlo, Bautista y Pereira [2] extendieron SALBP a TSALBP mediante la siguiente formulación: la restricción de área puede considerarse asociando a cada tarea  $j$  un área requerida  $a_j$ . Cada estación  $k$  requerirá un área de estación  $a(S_k)$  igual a la suma de las áreas de todas las tareas asignadas a dicha estación. Dicha área requerida no podrá ser nunca mayor que el área  $A_k$  disponible para cada estación  $k$  que, por simplicidad, asumiremos que será idéntica para todas las estaciones e igual a  $A$ , siendo  $A = \max_{k=1,2,\dots,m} \{A_k\}$ .

En general, TSALBP puede formularse de la siguiente forma: dado un conjunto de  $n$  tareas con sus atributos temporales  $t_j$  y espaciales  $a_j$  ( $j = 1, 2, \dots, n$ ) y un grafo de precedencias, cada tarea debe asignarse a una estación particular de forma que:

1. Las restricciones de precedencia se satisfagan.
2. Ninguno de los tiempos de carga de las estaciones,  $t(S_k)$ , es mayor que el tiempo de ciclo  $c$ .
3. Ninguna de las áreas requeridas por cualquiera de las estaciones,  $a(S_k)$ , es mayor que el área disponible por estación,  $A$ .

TSALBP presenta ocho variantes dependiendo de los tres criterios de optimización manejados:  $c$  (el tiempo de ciclo),  $m$  (el número de estaciones) y  $A$  (el área de las estaciones). De estas variantes, hay cuatro problemas multiobjetivo dependiendo de la variable que se fije:  $c$ ,  $m$ ,  $A$  o ninguna. En esta contribución, abordaremos la variante 1/3 del TSALBP, que trata de minimizar el número de estaciones junto con su área, para un tiempo de ciclo de producto dado. Hemos seleccionado la variante 1/3 del TSALBP pues es bastante realista puesto que la producción anual de una planta industrial (y por tanto, su tiempo ciclo  $c$ ) suele fijarse por ciertos objetivos de mercado. Además, la búsqueda del mejor número de estaciones y áreas es lógica si queremos reducir costes y facilitar un día a día mejor a los trabajadores configurando estaciones con menos gente.

### III. SISTEMA MÚLTIPLE DE COLONIA DE HORMIGAS

Como hemos dichos anteriormente, seleccionamos MACS [1] por su buen rendimiento en el estudio experimental llevado a cabo en [13]. Asimismo, en esta primera aproximación al problema queríamos un algoritmo MOACO que alcanzara una buena convergencia a todo el frente del Pareto y no exclusivamente en la parte central o en los extremos, como en otros enfoques [13].

MACS se propuso como una variación de MACS-VRPTW [12], ambos basados en ACS [9]. Sin embargo, MACS usa sólo una matriz de feromona  $\tau$  y varias funciones de información heurística  $\eta_k$  (en nuestro caso,  $\eta^0$  y  $\eta^1$  para el tiempo de duración de cada tarea  $t_j$  y su área  $a_j$ , respectivamente). De ahora en adelante centraremos esta descripción en el caso de dos objetivos. De este modo, una hormiga se mueve de un nodo  $i$  a un nodo  $j$  aplicando la siguiente regla de transición:

$$j = \begin{cases} \arg \max_{j \in \Omega} (\tau_{ij} \cdot [\eta_{ij}^0]^{\lambda\beta} \cdot [\eta_{ij}^1]^{(1-\lambda)\beta}), & \text{si } q \leq q_0, \\ \hat{i}, & \text{en otro caso,} \end{cases}$$

donde  $\beta$  pondera la importancia relativa de la información heurística respecto al rastro de feromona,  $\lambda$  se calcula para cada hormiga  $h$  como  $\lambda = h/M$ , siendo  $M$  el número de hormigas y  $\hat{i}$  es un nodo seleccionado de acuerdo a la siguiente distribución de probabilidad:

$$p(j) = \begin{cases} \frac{\tau_{ij} \cdot [\eta_{ij}^0]^{\lambda\beta} \cdot [\eta_{ij}^1]^{(1-\lambda)\beta}}{\sum_{v \in \Omega} \tau_{iv} \cdot [\eta_{iv}^0]^{\lambda\beta} \cdot [\eta_{iv}^1]^{(1-\lambda)\beta}}, & \text{si } j \in \Omega, \\ 0, & \text{en otro caso,} \end{cases}$$

En esta fórmula,  $\Omega$  representa el vecindario factible actual de la hormiga. Cada vez que una hormiga cruza el arco  $\langle i, j \rangle$ , lleva a cabo la actualización local de feromona como sigue:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0$$

Inicialmente,  $\tau_0$  se calcula a partir de un conjunto de soluciones heurísticas tomando sus costes medios en cada una de las dos funciones objetivo,  $f^0$  y  $f^1$ , aplicando la siguiente expresión:

$$\tau_0 = \frac{1}{\hat{f}^0 \cdot \hat{f}^1}$$

Sin embargo, el valor de  $\tau_0$  no se fija durante la ejecución del algoritmo, como suele ocurrir en ACS, sino que se va adaptando. Al final de cada iteración, cada solución completa construida por las hormigas en la colonia se compara con el conjunto Pareto  $P$  generado hasta ese momento para comprobar si la solución citada es no dominada. En tal caso, se incluye en el archivo y se eliminan todas las soluciones dominadas. Entonces se calcula  $\tau'_0$  aplicando la ecuación anterior con los valores medios de cada función objetivo tomada de las soluciones incluidas en ese

momento en el conjunto Pareto. Entonces, si  $\tau'_0 > \tau_0$ , el valor de feromona inicial actual, se reinician los rastros de feromona al nuevo valor  $\tau_0 \leftarrow \tau'_0$ . En otro caso, la actualización global se lleva a cabo con cada solución  $S$  del conjunto Pareto optimal actual  $P$  aplicando la siguiente regla de transición sobre sus arcos componentes:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \frac{\rho}{f^0(S) \cdot f^1(S)}$$

### IV. NUESTRA PROPUESTA: ALGORITMOS MOACO Y GRASP PARA EL TSALBP-1/3

En nuestro caso, una solución es una asignación de diversas tareas a distintas estaciones. Al contrario que en un problema de asignación simple como el *bin packing*, nosotros debemos tener en cuenta la satisfacción de restricciones de precedencia. Mediante un enfoque constructivo, podemos afrontar el problema de precedencias de una forma correcta. A continuación describimos dos enfoques basados en MOACO y GRASP [11], comenzando por sus aspectos comunes y particularizando después las características específicas de cada uno de ellos.

#### A. Procedimiento constructivo aleatorio

Teniendo como base los procedimientos orientados a la estación ya existentes [20] y los enfoques voraces usados en [2] para abordar el SALBP y el TSALBP, precisamos introducir los siguientes elementos aleatorios en el esquema de construcción:

- Una regla de prioridad aleatoria para seleccionar entre las tareas candidatas a elegir en cada paso constructivo (que ya se presentó en [4] para el algoritmo MACS).
- Un nuevo mecanismo para decidir si una estación debe cerrarse o no.

Puesto que los algoritmos son constructivos y orientados a la estación<sup>3</sup>, abrirán una estación y seleccionarán una tarea entre las candidatas mediante una regla de prioridad aleatoria. Dependiendo del esquema del algoritmo, la estación actual se cerrará cuando se llene completamente (enfoque habitual en SALBP y TSALBP) o esa decisión se tomará aleatoriamente para incrementar la diversidad de la búsqueda. Entonces, se abrirá una nueva estación para que se rellene. Considerando este proceso y los dos aspectos anteriores, hemos diseñado dos enfoques de resolución del problema. El primero se basa en el algoritmo MACS estudiado en la Sección III y se detalla en las siguientes secciones. El segundo se inspira en la primera etapa de un algoritmo GRASP, por tanto se trata de un algoritmo voraz aleatorio. En adelante nos referiremos a este algoritmo como MORGA que es el acrónimo de *Multi-Objective Randomised Greedy Algorithm*.

<sup>3</sup>Los procedimientos orientados a la estación son aquellos que comienzan abriendo la primera estación y le asignan sucesivamente las tareas más apropiadas. Cuando la estación está completamente cargada se cierra y se abre la siguiente que, en ese momento, queda lista para ser rellena.

### B. Funciones objetivo y enfoque basado en Pareto

Además del procedimiento constructivo, ambos algoritmos también comparten algunos aspectos como los objetivos y el enfoque basado en Pareto.

De acuerdo a la formulación TSALBP, la variante 1/3 afronta la minimización del número de estaciones  $m$  y el área precisa  $A$ . Podemos formular matemáticamente nuestros objetivos de la forma:

$$Z_1(x) = m = \sum_{j=1}^{UB_m} \max_{j=1,2,\dots,n} \{x_{jk}\}$$

$$Z_2(x) = A = \max_{k=1,2,\dots,UB_m} \left\{ \sum_{j=1}^n a_j x_{jk} \right\}$$

donde  $UB_m$  es una cota superior del número de estaciones  $m$ ,  $a_j$  es la información de área de la tarea  $j$  y  $x_{jk}$  es una variable de decisión que toma el valor 1 si la tarea  $j$  se asigna a la estación  $k$ .

En este trabajo usamos un archivo de Pareto en cada algoritmo multi-objetivo considerado.

### C. Un algoritmos MACS para el TSALBP-1/3

#### C.1 Información heurística

MACS trabaja con dos valores de información heurística distintos, cada uno de ellos asociados a un criterio. En nuestro caso, el primero  $\eta_j^0$  está relacionado con el tiempo de operación requerido para cada tarea y el segundo  $\eta_j^1$  con el área requerida:

$$\eta_j^0 = \frac{t_j}{c} \cdot \frac{|F_j^*|}{\max_{i \in \Omega} |F_i^*|}$$

$$\eta_j^1 = \frac{a_j}{UB_A} \cdot \frac{|F_j^*|}{\max_{i \in \Omega} |F_i^*|}$$

donde  $UB_A$  es el límite superior para el área (la suma de las áreas de todas las tareas) y  $\Omega$  es el vecindario factible actual de la hormiga, es decir, las tareas que podrían seleccionarse. Ambas heurísticas toman valores en el intervalo  $[0, 1]$ , siendo 1 el más preferible.

Como es habitual en SALBP, se prefiere colocar primero en las estaciones aquellas tareas con un valor de tiempo alto (una gran duración) y área (ocupando un montón de espacio). Al margen de la información de área y tiempo, hemos añadido otra información que ya se usó en [2] relativa al número de tareas sucesoras de la tarea tratada. Es preferible asignar primero aquellas tareas con un gran número de sucesoras.

#### C.2 Rastro de feromona y cálculo de $\tau_0$

La información de rastro de feromona tiene que memorizar qué tareas son las más apropiadas para una estación. Por tanto, la feromona tiene que asociarse a un par  $(station_k, task_j)$ , siendo  $k = 1..n$  and  $j = 1..n$ . De este modo, al contrario que la información heurística, nuestra matriz de rastro de

feromona tiene una naturaleza bidimensional puesto que enlaza tareas y estaciones.

Asimismo, en los algoritmos ACO, se debe fijar un valor inicial para los rastros de feromona. Este valor se conoce como  $\tau_0$  y normalmente se obtiene según un algoritmo heurístico. Nosotros hemos usado dos algoritmos voraces mono-objetivo orientados a la estación, uno por heurística, que se comportan como sigue. Abren la primera estación y seleccionan la tarea mejor posible según sus informaciones heurísticas (relativas tanto a su duración y sus sucesoras  $\eta_j^0$ , como a su área y sus sucesoras  $\eta_j^1$ ), repitiendo este proceso hasta que no pueden incluirse debido a la limitación del tiempo de ciclo y por tanto se abre una nueva estación. Cuando no hay más tareas que asignar finaliza el algoritmo voraz.  $\tau_0$  se calcula entonces a partir de los costes de las dos soluciones obtenidas usando la expresión MACS:  $\tau_0 = \frac{1}{f^0(S_{time}) \cdot f^1(S_{area})}$ .

#### C.3 Cierre de estaciones y regla de transición

Al principio, decidimos cerrar la estación cuando ésta se completara según el tiempo de ciclo  $c$  fijado, como es habitual en las aplicaciones del SALBP y TSALBP. Entonces apreciamos que este esquema no era el más apropiado porque los frentes de Pareto obtenidos no tenían suficiente diversidad. Por tanto, introducimos un nuevo mecanismo en la construcción del algoritmo para cerrar la estación de acuerdo con una probabilidad, dada por la tasa de llenado de la estación:  $p(cierre) = \sum_{i \in S_k} t_i / c$

Esta distribución de probabilidad se calcula en cada paso constructivo. Así, su valor se aumenta progresivamente. Una vez que se ha calculado, se genera uniformemente un número aleatorio en  $[0, 1]$  para decidir si la estación se cierra o no según la citada probabilidad. Si la estación no se cierra aún, elegimos la siguiente tarea entre todas las candidatas usando una regla de transición modificada.

Dada la naturaleza unidimensional de la información heurística, modificamos la regla de transición original (ver Sección III) que elige entre las tareas candidatas para la hormiga actual en ese instante:

$$j = \begin{cases} \arg \max_{j \in \Omega} (\tau_{kj} \cdot [\eta_j^0]^{\lambda\beta} \cdot [\eta_j^1]^{(1-\lambda)\beta}), & \text{si } q \leq q_0, \\ \hat{i}, & \text{en otro caso,} \end{cases}$$

donde  $\hat{i}$  es un nodo seleccionado mediante la siguiente distribución de probabilidad:

$$p(j) = \begin{cases} \frac{\tau_{kj} \cdot [\eta_j^0]^{\lambda\beta} \cdot [\eta_j^1]^{(1-\lambda)\beta}}{\sum_{u \in \Omega} \tau_{ku} \cdot [\eta_u^0]^{\lambda\beta} \cdot [\eta_u^1]^{(1-\lambda)\beta}}, & \text{si } j \in \Omega, \\ 0, & \text{en otro caso.} \end{cases}$$

#### C.4 Enfoque multi-colonia

Con un procedimiento orientado a la estación pura la intensificación es demasiado alta en una parte del frente del Pareto (la correspondiente a soluciones con un número de estaciones pequeño y valores

de área altos) ya que el procedimiento constructivo sólo cierra estaciones cuando están completas (ver los resultados obtenidos en [4]). Introducimos entonces una distribución de probabilidad de acuerdo a la tasa de relleno para solucionar esta convergencia local e inducir más diversidad en los algoritmos con el fin de generar frentes de Pareto más distribuidos (ver Sección IV-A). A pesar de inducir diversidad, la aplicación de este esquema de cierre aleatorio de estaciones cuando se considera un umbral de llenado particular no proporciona intensificación en ciertas áreas del Pareto puesto que hay una baja probabilidad de rellenar estaciones completamente.

Por tanto, es necesario buscar un mejor equilibrio intensificación-diversificación. Este objetivo puede alcanzarse mediante la introducción de diversos umbrales de llenado asociados a las hormigas que construyen la solución. Dichos umbrales hacen que las diversas hormigas de la colonia tengan un comportamiento distinto para la búsqueda y, por tanto, el algoritmo ACO pasa a ser multi-colonia [17]. En nuestro caso, los umbrales los fijamos entre 0.2 y 0.9 y se consideran como un paso previo antes de decidir cerrar una estación. Por tanto, se modifica el procedimiento de construcción de la solución. Calculamos la probabilidad de cierre de la estación como es habitual basándonos en su tasa de llenado actual, pero tiene que superar el umbral de relleno de la hormiga antes de decidir si cerrar una estación o no según una decisión aleatoria de acuerdo a esa distribución.

#### D. MORGA

Además del diseño del algoritmo MACS, hemos construido un MORGA. Nuestro mecanismo de generación de diversidad se comporta de manera similar a la fase construcción GRASP [11]. El elemento más importante en este tipo de construcción es que la selección de la tarea en cada paso debe guiarse por una función voraz que se adapta según las selecciones pseudoaleatorias realizadas en los pasos anteriores.

Usamos el mismo enfoque constructivo que en el algoritmo MACS, con umbrales de llenado y probabilidades de cierre en cada paso constructivo y sólo se cambia el criterio probabilístico para seleccionar la siguiente tarea en entrar en la estación actual para basarlo únicamente en información heurística, como se muestra en el siguiente párrafo.

##### D.1 Selección de candidatos y esquema basado en heurística

Para tomar una decisión entre todas las tareas candidatas factibles actuales usamos un valor heurístico particular dado por:

$$\eta_j = \frac{t_j}{c} \cdot \frac{a_j}{UB_A} \cdot \frac{|F_j^*|}{\max_{v_i \in \Omega} |F_i^*|}$$

Pero a pesar del algoritmo voraz, la decisión se toma aleatoriamente entre las tareas seleccionadas en la lista de candidatos restringida (RCL) mediante el

siguiente procedimiento: calculamos el valor heurístico de que cada tarea candidata factible sea asignada a la estación abierta en ese instante. Las ordenamos según sus valores heurísticos y asignamos un umbral de calidad para la heurística dado por:  $h = \max_{\eta_j} + \alpha \cdot (\max_{\eta_j} - \min_{\eta_j})$ .

Todas las tareas con un valor de heurística  $\eta_j$  igual o mayor que  $h$  se seleccionan para estar en la RCL.  $\alpha$  representa el parámetro de control de equilibrio diversificación-intensificación. Es un valor real entre 0 y 1, y cuando  $\alpha = 1$  hay una elección completamente aleatoria induciendo la mayor diversificación posible. Por contra, si  $\alpha = 0$  se aproxima a una elección voraz pura, con uná diversificación mínima.

##### D.2 Esquema de cierre de estación aleatorio

Como hicimos en el algoritmo MACS, hay un mecanismo en el algoritmo de construcción MORGA que nos permite cerrar una estación de acuerdo con una distribución de probabilidad, dada por la tasa de relleno de la estación:  $p(\text{cierre}) = \sum_{v_i \in S_k} t_i/c$

A pesar de la existencia de esta tasa de relleno, hemos explicado en las secciones anteriores que no es suficiente para conseguir un frente de Pareto diverso. Por tanto, usamos la misma técnica de umbrales de llenado. La diferencia radica en el uso paralelo de los mismos por parte del enfoque multi-colonia del algoritmo MACS. En el caso de MORGA, se usan distintos umbrales en cada iteración (i.e., en la construcción de cada solución diferente).

## V. EXPERIMENTOS

En esta sección analizamos el comportamiento de los diferentes algoritmos. En primer lugar, presentamos el ajuste de parámetros llevado a cabo y las instancias de problemas usadas. A continuación, mostramos el análisis experimental desarrollado.

### A. Instancias de problema

Consideramos diez instancias SALBP-1<sup>4</sup> para ejecutar todos nuestros experimentos TSALBP-1/3 con instancias realistas. Originalmente, estas instancias sólo tenían información temporal pero hemos creado su información de área a partir de aquella invirtiendo el grafo de tareas para hacerlo biobjetivo (como en [2]).

En la Tabla I se muestran todas las instancias junto con sus valores característicos principales. OS representa el *order strength* del grafo de precedencia. Cuanto mayor es su valor, mayor es el número restricciones de precedencias que encontraremos en una instancia. TV es la variabilidad temporal (*time variability*), es decir, la diferencia entre el tiempo de operación de la tarea mayor y menor.

### B. Diseño experimental

El algoritmo evolutivo multiobjetivo estado del arte, NSGA-II [8], se presenta como umbral de calidad

<sup>4</sup><http://www.assembly-line-balancing.de>

TABLA I  
 INSTANCIAS DE PROBLEMA CONSIDERADAS

Instancia y nombre	#T	OS	TV
P1 arc111 ( $c = 5755$ )	111	40.38	568.90
P2 arc111 ( $c = 7520$ )	111	40.38	568.90
P3 barthol2 ( $c = 85$ )	148	25.8	83
P4 barthold ( $c = 805$ )	148	25.80	127.6
P5 heskia ( $c = 342$ )	28	22.49	108
P6 lutz2 ( $c = 16$ )	89	77.55	10
P7 lutz3 ( $c = 75$ )	89	77.55	74
P8 mukherje ( $c = 351$ )	94	44.8	21.38
P9 scholl ( $c = 1394$ )	297	58.16	277.2
P10 wee-mag ( $c = 56$ )	75	22.67	13.50

#T: número de tareas,  
 OS: *order strength* del grafo de precedencia,  
 TV: variabilidad temporal.

para comparar cada algoritmo. A continuación, se presentan los valores de parámetros escogidos.

### B.1 Un enfoque basado en NSGA-II

Con respecto a las características dependientes del problema, las hemos adoptado del algoritmo genético propuesto en [19]. Se diseñó para resolver el problema SALBP y, en pocas palabras, sus características se resumen como sigue:

- Codificación: se usa un esquema de codificación de orden. La longitud del cromosoma será el número de tareas y el procedimiento para agruparlas para formar estaciones se guía por completar el tiempo de ciclo disponible de cada estación.
- Población inicial: generada aleatoriamente asegurando la factibilidad de las relaciones de precedencia.
- Cruce: se seleccionan dos padres y se cortan en dos puntos de corte aleatorios. Los descendientes toman los mismos genes fuera de los puntos de corte de sus padres y los genes entre los puntos de corte se mezclan de acuerdo al orden que tienen en el otro padre. El descendiente será una solución factible.
- Mutación: se selecciona un punto de corte aleatorio y los genes tras el punto de corte se sustituyen aleatoriamente asegurando la factibilidad.

### B.2 Valores de parámetros

Se han llevado a cabo diez ejecuciones para cada uno de los algoritmos considerados y para cada una de las once instancias de problema estudiadas. Los valores de los parámetros considerados se muestran en la Tabla II. De acuerdo con un estudio experimental previo [5], identificamos que el mejor comportamiento de la versión original de MACS se obtenía con  $q_0 = 0,2$ . Con respecto a los mecanismos de introducción de diversidad mediante umbrales de llenado variables, si comparamos con respecto a la versión original de MACS el mejor enfoque es MACS 0.2 con umbrales. Por su parte, el mejor rendimiento de MORGA se obtuvo con  $\alpha = 0,3$ .

 TABLA II  
 VALORES DE PARÁMETROS CONSIDERADOS

Parámetro	Valor
<b>GENERAL</b>	
Nº ejecuciones	10
Max. tiempo ejecución	900 segundos
PC Specs.	Intel Pentium <sup>TM</sup> D 2 CPUs a 2.80GHz
OS	CentOS Linux 4.0 GCC 3.4.6
<b>MACS</b>	
Nº de hormigas	10
$\beta$	2
$\rho$	0.2
$q_0$	0,2
Umbrales de hormigas	{0,2, 0,4, 0,6, 0,7, 0,9} (2 hormigas por umbral)
<b>MORGA</b>	
$\alpha$	0,3
Umbrales de diversidad	{0,2, 0,4, 0,6, 0,7, 0,9}
<b>NSGA-II</b>	
Tamaño población	100
Probabilidad cruce	0,8
Probabilidad mutación	0,01

### C. Estudio del uso de información heurística en las variantes de MACS

El estudio experimental previo citado nos ayudó a fijar ciertos parámetros y poder centrarnos en la influencia del uso de la información heurística en el rendimiento de la versión de MACS que nos dio mejores resultados, esto es, MACS 0.2 con umbrales de llenado variables (MACS con un enfoque multi-colonia y con  $q_0 = 0,2$ ). Consideramos diversas combinaciones para las definiciones de los valores heurísticos  $\eta_0$  y  $\eta_1$  en tres variantes del algoritmo:

- MACS c-succ. (c con sucesores):

$$\eta_j^0 = \frac{t_j}{c} \cdot \frac{|F_j^*|}{\max_{v_i \in \Omega} |F_i^*|} \quad \eta_j^1 = \frac{a_j}{UB_A}$$

- MACS a-succ. (a con sucesores):

$$\eta_j^0 = \frac{t_j}{c} \quad \eta_j^1 = \frac{a_j}{UB_A} \cdot \frac{|F_j^*|}{\max_{v_i \in \Omega} |F_i^*|}$$

- MACS no-succ. (sin sucesores):

$$\eta_j^0 = \frac{t_j}{c} \quad \eta_j^1 = \frac{a_j}{UB_A}$$

Además, consideramos eliminar la información heurística, guiando la búsqueda en MACS sólo según el rastro de feromona (MACS 0.2 no-heur.).

Hemos usado la métrica binaria  $C$  [6] para comparar los conjuntos Pareto obtenidos. Usamos *box-plots* basados en la métrica  $C$  que comparan cada par de algoritmos calculando el grado de dominancia de sus respectivos conjuntos Pareto (ver Figura 1).

Además de la comparación entre las distintas variantes de MACS, se consideran la mejor configuración de MORGA y el enfoque NSGA-II (descrito en la Sección V-B) para medir el rendimiento de los algoritmos MACS. Cada rectángulo contiene diez *box-*

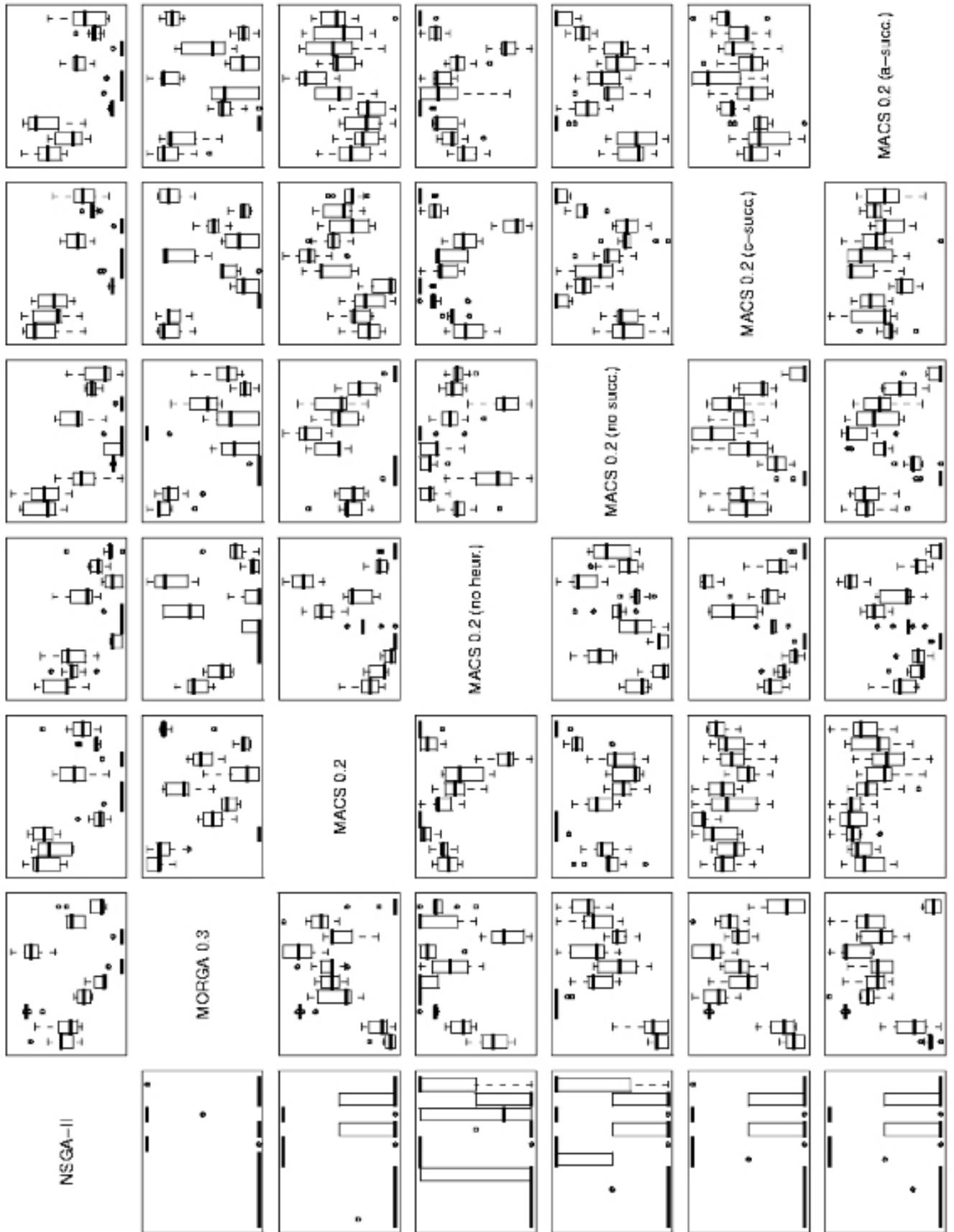


Fig. 1. Valores de la métrica C representados mediante box-plots comparando los mejores algoritmos y diversas configuraciones para la información heurística.

plots representando la distribución de los valores de  $C$  para un cierto par de algoritmos en las diez instancias de problema (P1 a P10).

La idea principal que derivamos de estos resultados es el buen rendimiento de MACS sin considerar información heurística, que es significativamente mejor que MORGA, NSGA-II y las variantes de MACS que sí se basan en la información heurística. Podríamos pensar que la información heurística no ayuda a que el algoritmo cubra toda la extensión del frente de Pareto. Sin embargo, hay algunas instancias de problemas en los que el uso de tal información sí es de utilidad, como en P3 y P8. En dichas instancias, sin el uso de información heurística nuestros algoritmos no son capaces de alcanzar ciertas áreas de los frentes de Pareto como se puede observar en el gráfico de la Figura 2.

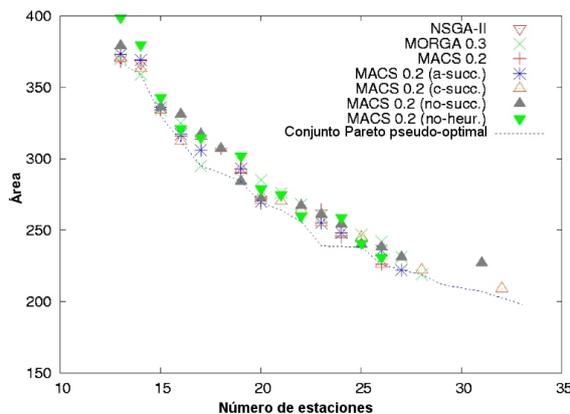


Fig. 2. Frentes de Pareto para la instancia de problema P8.

A pesar de estos buenos resultados, es importante resaltar que cada conjunto de Pareto pseudo-optimal incluye soluciones que no se han obtenido mediante MACS no-heur. Por ejemplo, el enfoque NSGA-II, que no es capaz de hallar un buen conjunto Pareto, suele conseguir un par de soluciones no dominadas pertenecientes al conjunto Pareto pseudo-optimal.

## VI. CONCLUSIONES

En esta contribución hemos propuesto enfoques multi-objetivo para abordar el TSALBP-1/3. Se ha presentado el rendimiento de dos procedimientos de solución basados en los algoritmos MACS y MORGA. Hemos considerado variantes bi-objetivo de diez instancias de problemas de líneas de montaje. De los resultados obtenidos deducimos que el mejor comportamiento para solucionar globalmente el problema corresponde al algoritmo MACS usando un esquema multi-colonia. Asimismo, el uso de una variante sin información heurística ha alcanzado los mejores resultados para la mayoría de las instancias consideradas. MACS y MORGA se comportan mejor si usamos un valor 0.2 para  $q_0$  en el caso de MACS y 0.3 como parámetro de control  $\alpha$  en la RCL de MORGA. Por tanto, es preciso incluir diversidad en nuestro problema para alcanzar mejores resultados.

## AGRADECIMIENTOS

Agradecemos a la Cátedra Nissan así como al gobierno español por financiar parcialmente este trabajo a través del proyecto PROTHIUS-II project: DPI2007-63026 incluyendo fondos FEDER.

## REFERENCIAS

- [1] B. Barán, M. Schaerer. A multiobjective ant colony system for vehicle routing problem with time windows. In Proc. 21st IASTED International Conference, 97-102, Innsbruck (Alemania), February 2003.
- [2] J. Bautista, J. Pereira. Ant algorithms for a time and space constrained assembly line balancing problem. *EJOR* 177, 2016-2032, 2007.
- [3] I. Baybars. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science* 32 (8), 909-932, 1986.
- [4] M. Chica, O. Cerdón, S. Damas, J. Bautista, J. Pereira. A Multiobjective Ant Colony Optimization Algorithm for the 1/3 Variant of the Time and Space Assembly Line Balancing Problem. In Proc. 12th Intl. Conf. IPMU'08, Malaga (España), 1454-1461, June 2008.
- [5] M. Chica, O. Cerdón, S. Damas, J. Bautista, J. Pereira. Multi-objective, constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and randomised greedy. Tech. Rep. AFE-08-01, European Centre for Soft Computing (sometido a *IEEE Trans. on Evolutionary Computation*).
- [6] C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont. *Evolutionary Algorithms for Solving Multi-objective Problems* (2nd edition). Springer, 2007.
- [7] O. Cerdón, F. Herrera, T. Stützle. A review on the ant colony optimization metaheuristic: Basis, models and new trends. *Mathware-Soft Computing* 9, 141-175, 2002.
- [8] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation* 6, 182-197, 2002.
- [9] M. Dorigo, L. Gambardella. Ant colony system: A cooperative learning approach to the travelling salesman problem. *IEEE Trans. on Evolutionary Computation* 1 (1), 53-66, 1997.
- [10] M. Dorigo, T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, 2004.
- [11] T.A. Feo, M.G.C. Resende. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization* 6, 109-134, 1995.
- [12] L. Gambardella, E. Taillard, G. Agazzi. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In *News ideas in optimization*, 73-76, Londres (Reino Unido), 1999.
- [13] C. García Martínez, O. Cerdón, F. Herrera. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *EJOR* 180, 116-148, 2007.
- [14] G.A. Kochenberger, F. Glover. *Handbook of Metaheuristics*. Kluwer Academic, 2003.
- [15] T.C. Koopmans, M.J. Beckmann. Assignment problems and the location of economic activities. *Econometrica* 25, 53-76, 1957.
- [16] J.A. Lozano, P. Larrañaga, M. Graña. Partitional cluster analysis with genetic algorithms: searching for the number of clusters. In *Data Science, Classification and Related Methods*, 117-125. Springer, 1998.
- [17] M. Middendorf, F. Reischle, H. Schmeck. Multi colony ant algorithms. *Journal of Heuristics* 8, 305320, 2002.
- [18] A.M. Robertson, P. Willett. Generation of Equifrequent Groups of Words Using a Genetic Algorithm. *Journal of Documentation* 50 (3), 213-232, 1994.
- [19] I. Sabuncuoglu, E. Erel, M. Tayner. Assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing* 11, 295310, 2000.
- [20] A. Scholl, S. Voss. Simple assembly line balancing- Heuristic approaches. *Journal of Heuristics* 2, 217-244, 1996.
- [21] A. Scholl. *Balancing and sequencing of assembly lines*. Physica-Verlag, Heidelberg, 1999.
- [22] A. Scholl, C. Becker. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *EJOR* 180, 116-148, 2006.