

**XIII Congreso Español
de Metaheurísticas,
Algoritmos Evolutivos y
Bioinspirados
(XIII MAEB)**

**MAEB 6: SESIÓN ESPECIAL:
METAHEURÍSTICAS EN
PRODUCCIÓN, LOGÍSTICA Y
DIRECCIÓN DE OPERACIONES**

Organizador:

JOAQUÍN BAUTISTA-VALHONDO





A GRASP algorithm to optimize Operational Costs and Regularity of Production in mixed-model sequencing problems with forced interruption of operations*

***Note:** The full contents of this paper have been published in the volume *Lecture Notes in Artificial Intelligence 11160* (LNAI 11160)

Joaquín Bautista-Valhondo
Universidad Politècnica de Catalunya
Barcelona, Spain
joaquin.bautista@upc.edu

Rocío Alfaro-Pozo
EAE Business School
Barcelona, Spain
ralfaro@eae.es

Abstract—We present a GRASP algorithm to solve a problem that involves the sequencing of mixed products in an assembly line. The objective of the problem is to obtain a manufacturing sequence of models that generates a minimum operational cost with a forced interruption of operations and that is regular in production. The implemented GRASP is compared with other procedures using instances of a case study of the Nissan engine manufacturing plant in Barcelona.

Index Terms—GRASP, Bounded Dynamic Programming, Work overload, Operational Cost, Mixed-model Sequencing Problems

Metaheurísticas constructivas para *Car Sequencing Problem* con Flotas de vehículos especiales

Ignacio Moya*, Joaquín Bautista†, Manuel Chica*, Sergio Damas*, Oscar Cordon*

*Instituto Andaluz Interuniversitario DaSCI (Data Science and Computational Intelligence), Universidad de Granada, España

† Nissan Chair, Universidad Politécnica de Catalunya, España

Emails: imoya@ugr.es, manuelchica@ugr.es, sdamas@ugr.es, joaquin.bautista@upc.edu, ocordova@decsai.ugr.es

Resumen—*Car sequencing problem* (CSP) es un problema tradicional de satisfacción de restricciones que refleja los problemas que surgen cuando una serie de vehículos se introducen en una cadena de producción. Sin embargo no considera vehículos no regulares o fuera de catálogo, pese a que en plantas reales pueden llegar a representar entre el 10 % y el 20 % de la producción. Estos vehículos se distinguen de los regulares por ser bajo pedido, estar orientados al servicio público, y/o requerir componentes poco frecuentes. La extensión *Robust CSP* (r-CSP) sí que considera estos vehículos, modelando la incertidumbre que introducen los vehículos no regulares como escenarios de producción. Para resolver r-CSP proponemos emplear metaheurísticas constructivas que ya han sido efectivas en el CSP tradicional: GRASP y VNS. Evaluamos el rendimiento de estos algoritmos en r-CSP extendiendo las instancias disponibles en la literatura añadiendo la semántica de flotas de vehículos especiales. Nuestros experimentos reflejan que GRASP muestra mejor rendimiento en las instancias más sencillas, mientras que VNS se muestra superior en las más complejas.

Index Terms—Metaheurísticas, Líneas de montaje, Secuenciación, Car Sequencing Problem, Flotas de vehículos

I. INTRODUCCIÓN

Car Sequencing Problem (CSP) fue introducido por Parello *et al.* [1] a consecuencia de la transición de la industria hacia la personalización masiva. Es un problema de satisfacción de restricciones que representa las dificultades que surgen al introducir vehículos con opciones diferentes en una misma línea de montaje, transformando restricciones de tiempo, espacio, o técnicas en opciones binarias de modo que todo vehículo requiera de un conjunto arbitrario de estas opciones. Estas opciones se modelan indicando las paradas que pueden aparecer en una línea de montaje si muchos vehículos seguidos requieren una determinada opción, por lo que cada opción se representa usando un ratio de carga. Este ratio se representa como p_i/q_i , donde p es el número de vehículos que pueden requerir la opción i en una secuencia de longitud q sin causar paradas en la línea. El objetivo del CSP es encontrar una secuencia completa de vehículos que no sobrecargue ninguna opción en ninguna de sus subsecuencias.

Sin embargo el CSP no considera vehículos no regulares o fuera de catálogo [2]. Estos vehículos se distinguen de los regulares por ser bajo pedido, estar orientados al servicio público, y/o requerir componentes poco frecuentes. El impacto de estos vehículos en la cadena genera incertidumbre y afecta a la producción, ya que pueden alcanzar entre el 10 % y el 20 %

de la producción. Una alternativa para tratar esta incertidumbre es definir distintos escenarios de producción que sean realistas.

La extensión *Robust CSP* (r-CSP) [2] sigue este enfoque, donde la demanda se define en distintos planes de producción y el impacto de la variabilidad se mide observando los conflictos que ocurren cuando se cambia de un plan a otro. En este artículo estudiamos la resolución de r-CSP utilizando como referencia el trabajo anterior para el CSP tradicional, que comúnmente se resuelve usando metaheurísticas. Distintas metaheurísticas se han aplicado con éxito al CSP, de las cuales nosotros destacamos las metaheurísticas constructivas GRASP [3] y VNS [4], [5].

El resto del texto sigue la siguiente estructura. La Sección II revisa la bibliografía relevante relacionada con la resolución del CSP tradicional. Después presentamos brevemente la formalización del r-CSP en la Sección III. La Sección IV describe las metaheurísticas empleadas en nuestros experimentos. En la Sección V presentamos la configuración y los resultados de nuestros experimentos para el r-CSP. Por último, en la Sección VI presentamos nuestras conclusiones.

II. TRABAJO RELACIONADO

Existen diferentes alternativas para resolver el CSP tradicional. Encontramos varios métodos exactos basados en programación con restricciones [6], [7], *integer programming* [8], algoritmos de ramificación y poda [9], o *beam search* [10], [11]. También encontramos muchos ejemplos de resolución usando metaheurísticas [5], [12], [13], [14], [15]. Todos estos métodos se suelen comparar usando el *framework* de referencia CSPLib [16].

De entre los distintos trabajos que utilizan metaheurísticas, podemos destacar los métodos basados en búsqueda local, VNS y GRASP. De entre los trabajos que emplean búsqueda local, el trabajo de Puchta y Gottlieb [17] es especialmente relevante dado que introduce nuevos operadores que han sido empleados posteriormente en otros trabajos [3], [5], [12], [13]. Uno de estos operadores, llamado operador de inversión o *Lin2Opt* se ha convertido en un operador recurrente. Los métodos de búsqueda local normalmente incluyen heurísticas voraces para construir la solución inicial, de las cuales encontramos que la propuesta por Gottlieb *et al.* [12] es la más empleada [5], [13], [14], [18], [19].

En el caso de VNS y GRASP, las contribuciones más relevantes se aplican a extensiones del CSP tradicional. En



el caso de VNS, destacamos el trabajo propuesto de Ribeiro *et al.* [4]. Su versión de VNS combinada con ILS terminó en segunda posición en el ROADEF'2005 [20], donde el problema a resolver era una extensión del CSP tradicional que incluye nuevas restricciones relacionadas con la pintura. Hemos incluido algunas de las características de su trabajo en nuestro diseño de VNS, como el uso del operador de inserción (explicado en la Sección IV-C) como operador en la perturbación de la solución. En el caso del GRASP podemos encontrar trabajos donde se utiliza para resolver extensiones del CSP tradicional [3], por lo que incluimos estos diseños en nuestro estudio.

III. DESCRIPCIÓN DEL MODELO

En esta sección describimos las particularidades del CSP con flotas de vehículos especiales (r-CSP). Este modelo sigue las siguientes hipótesis de trabajo:

1. Hay dos conjuntos separados de familias de vehículos: los regulares I_X y los especiales $I_{X'}$.
2. Existe una demanda de vehículos regulares D_X fija para cada plan de producción $\varepsilon \in E$.
3. Existe una demanda de vehículos especiales $D_{X'}$ fija para cada plan de producción $\varepsilon \in E$.
4. La demanda total de vehículos $T (T \equiv D \equiv D_X + D_{X'})$, se respeta en todos los planes de producción $\varepsilon \in E$.
5. La demanda para cada vehículo regular $d_i, i \in I_X$ es la misma para cada plan de producción $\varepsilon \in E$. Esto es, $d_{i,\varepsilon} = d_i, \forall \varepsilon \in E$.
6. La demanda de cada vehículo especial $i \in I_{X'}$, puede cambiar para dos planes de producción distintos $\{\varepsilon, \varepsilon'\} \subseteq E$.
7. Los cambios en la cadena de producción deberían ser tan pocos como sea posible, por lo que las secuencias de producción $\pi_\varepsilon(T)$ y $\pi_{\varepsilon'}(T)$ deberían ser lo más similares posible para cada par de planes $\{\varepsilon, \varepsilon'\} \subseteq E$. Esto es: $\pi_\varepsilon(T) \approx \pi_{\varepsilon'}(T), \forall \{\varepsilon, \varepsilon'\} \subseteq E$.
8. Cada vehículo regular mantendrá la misma posición en todas las secuencias $\pi_\varepsilon(T), \forall \varepsilon \in E$.

A continuación incluimos la definición formal de r-CSP en el ámbito de programación lineal entera mixta. También incluimos un conjunto de restricciones que cualquier secuencia válida debe cumplir. Las restricción (1) impone que solo un vehículo (regular o especial) puede asignarse en cada ciclo de producción $t \in [1, T]$ para todos los planes de producción $\varepsilon \in E$. Las restricciones (2) y (3) fuerzan a que se satisfaga la demanda de vehículos regulares (I_X) y especiales ($I_{X'}$) en cada plan de producción. La restricción (4) determina la producción acumulada de la clase de vehículos regulares $i \in I_X$ hasta el ciclo de fabricación t . La restricción (5) es análoga a la anterior, referida a los vehículos especiales. Las restricciones (6) y (7) imponen que el número de veces que se requiere la opción $j \in J$ de manera consecutiva en cualquier ciclo de producción $t \in [1, T]$ para todo plan de producción $\varepsilon \in E$ no supere su ratio p/q . Las restricciones (8), (9), y (10) definen respectivamente las variables binarias $x_{i,t}$, $x_{i,t,\varepsilon}$,

y $z_{j,t,\varepsilon}$. Finalmente, la restricción (11) fija un valor nulo para las variables $Y_{j,0,\varepsilon} (\forall j \in J, \forall \varepsilon \in E)$ en el ciclo virtual $t = 0$.

$$\sum_{i \in I_X} x_{i,t} + \sum_{i \in I_{X'}} x_{i,t,\varepsilon} = 1, \forall t \in [1, T], \forall \varepsilon \in E \quad (1)$$

$$\sum_{t \in [1, T]} x_{i,t} = d_i \forall i \in I_X \quad (2)$$

$$\sum_{t \in [1, T]} x_{i,t,\varepsilon} = d_{i,\varepsilon}, \forall i \in I_{X'}, \forall \varepsilon \in E \quad (3)$$

$$X_{i,t} - \sum_{\tau \in [1, t]} x_{i,\tau} = 0, \forall i \in I_X, \forall t \in [1, T] \quad (4)$$

$$X_{i,t,\varepsilon} - \sum_{\tau \in [1, t]} x_{i,\tau,\varepsilon} = 0, \quad (5)$$

$$\forall i \in I_{X'}, \forall t \in [1, T], \forall \varepsilon \in E$$

$$Y_{j,t,\varepsilon} - \sum_{i \in I_X} n_{j,i} X_{i,t} - \sum_{i \in I_{X'}} n_{j,i} X_{i,t,\varepsilon} = 0, \quad (6)$$

$$\forall j \in J, \forall t \in [1, T], \forall \varepsilon \in E$$

$$Y_{j,t,\varepsilon} - Y_{j,t-q_j,\varepsilon} \leq p_j + T \cdot z_{j,t,\varepsilon}, \quad (7)$$

$$\forall j \in J, \forall t \in [q_j, T], \forall \varepsilon \in E$$

$$x_{i,t} \in \{0, 1\}, \forall i \in I_X, \forall t \in [1, T] \quad (8)$$

$$x_{i,t,\varepsilon} \in \{0, 1\}, \forall i \in I_{X'}, \forall t \in [1, T], \forall \varepsilon \in E \quad (9)$$

$$z_{j,t,\varepsilon} \in \{0, 1\}, \forall j \in J, \forall t \in [q_j, T], \forall \varepsilon \in E \quad (10)$$

$$Y_{j,0,\varepsilon} = 0, \forall j \in J, \forall \varepsilon \in E \quad (11)$$

r-CSP define multiseuencias $\vec{\pi}(E, T)$ compuestas de secuencias individuales $\pi_\varepsilon(T) = (\pi_{1,\varepsilon}, \dots, \pi_{T,\varepsilon})$ para cada plan de producción $\varepsilon \in E$, como se muestra en la ecuación (12). Los vehículos regulares $i \in I$ se relacionan con los elementos $\pi_{t,\varepsilon} (\forall t \in [1, T], \forall \varepsilon \in E)$ de la multiseuencia $\vec{\pi}(E, T)$ por las variables binarias $x_{i,t} (\forall i \in I_X, \forall t \in [1, T])$ y $x_{i,t,\varepsilon} (\forall i \in I_{X'}, \forall t \in [1, T], \forall \varepsilon \in E)$, como se define en las ecuaciones (13) y (14). Debemos remarcar que todo vehículo regular ($i \in I_X$) mantiene su posición en la secuencia para cada plan de producción, pero las posiciones ocupadas por vehículos especiales dependen del plan de producción $\varepsilon \in E$. De este modo podemos decir que toda secuencia $\pi_\varepsilon(T)$ tiene un componente común (los vehículos regulares) y un componente exclusivo (sus vehículos especiales).

$$\pi_\varepsilon(T) = \left\{ \begin{array}{c} \pi_1(T) \\ \pi_2(T) \\ \dots \\ \pi_\varepsilon(T) \\ \dots \\ \pi_{|E|}(T) \end{array} \right\} = \left\{ \begin{array}{c} (\pi_{1,1}, \dots, \pi_{t,1}, \dots, \pi_{T,1}) \\ (\pi_{1,2}, \dots, \pi_{t,2}, \dots, \pi_{T,2}) \\ \dots \\ (\pi_{1,\varepsilon}, \dots, \pi_{t,\varepsilon}, \dots, \pi_{T,\varepsilon}) \\ \dots \\ (\pi_{1,|E|}, \dots, \pi_{t,|E|}, \dots, \pi_{T,|E|}) \end{array} \right\} \quad (12)$$

$$x_{i,t} = 1 \Rightarrow \pi_{t,\varepsilon} = i, \forall i \in I_X, \forall t \in [1, T], \forall \varepsilon \in E \quad (13)$$

$$x_{I,t} = 1 \Rightarrow \pi_{t,\varepsilon} = i, \forall i \in I_{X_I}, \forall t \in [1, T], \forall \varepsilon \in E \quad (14)$$

Si siguiendo esta formulación definimos la función objetivo para r-CSP. Esta función, que mostramos en la ecuación (15), minimiza el número de violaciones de restricciones para las opciones $j \in J$ usando sus ratios de carga p_j/q_j para todos los planes de producción $\varepsilon \in E$. Usando este enfoque tratamos r-CSP como un problema de satisfacción de restricciones y conectamos con la formulación original de Parello *et al.* [1].

$$\begin{aligned} \min Z &= \sum_{j \in J} \sum_{t \in [q_j, T]} \sum_{\varepsilon \in E} z_{j,t,\varepsilon} \\ &\Leftrightarrow \max Z_I = \sum_{j \in J} \sum_{t \in [q_j, T]} \sum_{\varepsilon \in E} (1 - z_{j,t,\varepsilon}) \end{aligned} \quad (15)$$

IV. MÉTODOS

Como ya hemos anticipado, existen distintos casos de éxito de la aplicación de metaheurísticas al CSP tradicional [4], [12], [13], [17], [21]. En muchas de estos casos se aplican metaheurísticas constructivas [22], por lo que proponemos resolver r-CSP también con estos métodos. En concreto, hemos elegido GRASP y VNS.

A. GRASP

Nuestro diseño de *Greedy Randomized Adaptive Search Procedure* (GRASP) [23] está basado en la heurística propuesta por Gottlieb *et al.* [12]. De este modo, la lista de candidatos para la fase constructiva se compone en tres etapas. En el primer paso, se descartan los vehículos que violarían alguna restricción de carga p/q , salvo que no existan vehículos que no violen alguna restricción, que en ese caso deberán incluir el mínimo de nuevas violaciones posible. En el segundo paso, si existe más de un elemento en la lista (muy posible sobretodo en los primeros pasos), esos elementos se ordenan según su valor heurístico usando la heurística de Gottlieb *et al.* [12]. De este modo se favorece incluir primero los vehículos que incluyen opciones más restrictivas. Finalmente la lista de candidatos se trunca dependiendo del parámetro $\alpha \in (0, 1]$, dejando en la lista los elementos con mejor valor heurístico. La selección final en este paso de la fase constructiva será un elemento aleatorio de la lista de candidatos resultante. Exponemos la fase de refinado posterior usando búsqueda local en la Sección IV-C.

B. VNS

Nuestra implementación de *Variable Neighborhood Search* (VNS) [24] construye su solución inicial siguiendo una estrategia voraz basada en la heurística de Gottlieb *et al.* [12]. A diferencia del diseño usado por GRASP, en este caso el proceso es determinista y siempre se elige el vehículo que incluya el menor número de violaciones posible con el mejor valor de la heurística.

Con respecto al diseño de la fase de perturbación de la solución, usamos un operador distinto al que después se utiliza para la fase de refinado. En concreto elegimos utilizar un operador de inserción, que se aplica k veces durante la fase de perturbación donde k es igual a la iteración actual del algoritmo. Por ejemplo, con $k = 3$ aplicamos tres veces el operador de manera secuencial. Este operador se describe en detalle junto con el resto de operadores de búsqueda local en la Sección IV-C.

C. Búsqueda local

Elegimos nuestros operadores para la búsqueda local basándonos en trabajo previo [12], [17] y el resultado de nuestros experimentos. Estos operadores son: *Swap*, *Inserción*, y *Inversión*. Para mostrar como estos operadores modifican una secuencia dada usaremos como referencia la cadena π descrita en la Ecuación 16. El operador de *Swap* intercambia dos posiciones aleatorias en una secuencia. En la Ecuación 17 se puede ver el resultado de aplicar el operador de *Swap* a la cadena π , intercambiando los elementos de los índices 3 y 7. El operador de *Inserción* selecciona un índice aleatoriamente y lo mueve a otra posición aleatoria de la cadena, desplazando las posiciones intermedias de manera que mantengan su orden en la secuencia. En la Ecuación 18 el operador de *Inserción* mueve el elemento en la posición 5 a la posición 9. El operador de *Inversión* invierte una subcadena definida por dos índices seleccionados aleatoriamente. En la Ecuación 19 se muestra cómo el operador de *Inversión* invierte una subcadena de π entre las posiciones 4 y 6.

$$\pi = 5, 3, 4, 2, 3, 4, 1, 5, 2, 0 \quad (16)$$

$$\pi_{Swap} = 5, 3, \mathbf{1}, 2, 3, 4, \mathbf{4}, 5, 2, 0 \quad (17)$$

$$\pi_{Insert} = 5, 3, 4, 2, 4, 1, 5, 2, \mathbf{3}, 0 \quad (18)$$

$$\pi_{Invert} = 5, 3, 4, \mathbf{4}, \mathbf{3}, \mathbf{2}, 1, 5, 2, 0 \quad (19)$$

Independientemente del operador, nuestra búsqueda local siempre se mueve al primer vecino que mejore la solución (*primer-mejor*). De este modo, en cada paso de la búsqueda local se genera y evalúa un vecino aleatorio de la solución actual, aceptándolo si mejora a la anterior y deshaciendo el cambio si no. Nuestra búsqueda local aplica dos operadores de manera secuencial: primero explora el espacio de búsqueda usando *Swap* hasta que la búsqueda termina; después se repite el proceso utilizando el operador de *Inversión*.



V. EXPERIMENTACIÓN

A. Configuración

Comparamos el rendimiento de las metaheurísticas seleccionadas usando las instancias publicadas en CSPLib [16], que es el *benchmark* generalmente utilizado por la comunidad del CSP [12], [13], [14]. Este benchmark contiene instancias con diferentes grados de complejidad que se podrían dividir en tres categorías: las factibles, las clásicas y las avanzadas. Las instancias factibles son un conjunto de 70 instancias de 200 vehículos con 5 opciones y entre 17 y 30 clases. Podemos argumentar que estas instancias son las más básicas dado que están diseñadas para ser factibles. Estas instancias están divididas en grupos de 10 según el porcentaje de requerimiento de sus opciones. Las instancias menos exigentes de esta categoría tienen un 60% y los distintos grupos de instancias aumentan su complejidad gradualmente hasta un 90%. Las instancias clásicas [16] son un grupo de 9 instancias con 100 vehículos con 5 opciones y entre 18 y 24 clases. Este grupo de instancias es más complejo porque sólo 4 de ellas son factibles, dado que su porcentaje de requerimiento de sus opciones está cercano al 100%. Finalmente las instancias avanzadas, propuestas por [8], son 30 secuencias de 200, 300, y 400 vehículos. Estas instancias tienen configuraciones de opciones y clases similares a las instancias clásicas. Igual que en el caso de las instancias clásicas, las instancias avanzadas están diseñadas para tener un porcentaje de requerimiento muy alto y para la mayoría de ellas no se conoce una solución factible [15].

Como estas instancias no consideran demanda parcial incierta, extendemos su semántica generando apéndices que incluyan flotas de vehículos especiales para las instancias ya existentes. De esta manera maximizamos la compatibilidad con los trabajos anteriores. El enfoque que seguimos es el siguiente: para cada instancia existente, consideramos como no regulares el 20% de sus clases de vehículos. La suma de la demanda original de los vehículos que pasan a tratarse como no regulares será la demanda de vehículos especiales en el plan de producción de referencia. En concreto, las clases de vehículos no regulares serán las últimas clases definidas en la instancia original. Por ejemplo, tomando secuencia original con 10 clases, las clases de vehículos 1 a 8 seguirán siendo vehículos regulares y las clases 9 y 10 serán consideradas no regulares. De este modo, la demanda de vehículos que consideran flotas de vehículos especiales depende de la demanda para las clases 9 y 10. Para terminar, se generan aleatoriamente 10 planes de producción utilizando una distribución uniforme. Continuando el ejemplo anterior, si la demanda original de las clases 9 y 10 era de 20 vehículos cada una, la suma de ambas será la demanda del plan de referencia y se generan 10 combinaciones aleatorias de 40 vehículos en total.

En cuanto a los cálculos de fitness, en el caso del CSP tradicional los valores de fitness representan el número de violaciones de restricciones p/q en una secuencia. Este es un caso especial de la función objetivo definida en la Ecuación 15 con un único plan de producción (i.e. $|E| = 1$). Podemos deno-

minar a esta función objetivo $regular(\pi_\varepsilon)$. Calcular el fitness para $|E| > 1$ implica generar y calcular cada combinación de planes de producción y vehículos no regulares contando las nuevas violaciones ($\forall \varepsilon \in E$). Dado que generar cada posible combinación para todas las secuencias implicaría tiempos de ejecución muy costosos, en su lugar optamos por una solución intermedia: simular este valor siguiendo un enfoque similar a las *simheurísticas* [25]. De este modo aproximamos el valor generando aleatoriamente $|M|$ secuencias posibles para cada plan de producción partiendo de la secuencia encontrada usando el plan de referencia π_ε y calculamos la media del mínimo de violaciones para cada plan $\varepsilon \in E$. La función $fleets(\pi_\varepsilon)$, definida en la Ecuación 20, es por tanto una aproximación para calcular la función objetivo de r-CSP (definida en la Ecuación 15 de la Sección III). Nuestra función de fitness final para resolver r-CSP combina el número de violaciones sobre el plan de referencia con la aproximación de las violaciones de las flotas de vehículos especiales usando el parámetro θ . Este parámetro θ se incluye para potenciar la relevancia del plan de referencia con respecto al resto de planes de producción. Ajustamos los parámetros θ y $|M|$ durante la experimentación y finalmente se fijan a $\theta = 0,6$ y $|M| = 30$. En la Ecuación 21 mostramos la función objetivo final que utilizamos en nuestros experimentos ($fitness(\pi_\varepsilon)$).

$$fleets(\pi_\varepsilon) = \frac{\min \sum_{j \in J} \sum_{t \in [q_j, T]} \sum_{\varepsilon \in E} z_{j,t,\varepsilon}}{|M|}, \quad \forall m \in M \quad (20)$$

$$fitness(\pi_\varepsilon) = \theta * regular(\pi_\varepsilon) + (1 - \theta) * fleets(\pi_\varepsilon) \quad (21)$$

Tanto GRASP como VNS ejecutan durante su búsqueda local un máximo de 10,000 pasos, y contemplan 400 iteraciones respectivamente. De este modo, nuestra experimentación fija un criterio de parada de 4,000,000 evaluaciones para las metaheurísticas, que se ejecutan 30 veces utilizando diferentes semillas. Además, fijamos el parámetro $\alpha = 0,15$ para las ejecuciones de GRASP. En cuanto a la implementación, hemos implementado cada metaheurística en Java y hemos empleado JAMES [26] para la búsqueda local.

B. Comparación algorítmica para el r-CSP

En esta sección discutimos el rendimiento de las metaheurísticas sobre las instancias modificadas para incluir flotas de vehículos especiales. Por razones de espacio los resultados sobre estas instancias se muestran divididos en las Tablas I (instancias básicas desde 60-01 hasta 85-10) y II (instancias básicas entre 90-01 y 90-10, instancias clásicas p^* , e instancias avanzadas pb_{200}^* , pb_{300}^* , y pb_{400}^*).

En la Tabla I vemos que prácticamente todos los valores resaltados son de GRASP, tanto en mínimo valor como en valor promedio de *fitness*. La superioridad de GRASP en estas instancias podría deberse a la baja carga de sus opciones. Además con la configuración de parámetros elegida puede que



Tabla I
VALORES MÍNIMOS Y PROMEDIO DE LAS 30 EJECUCIONES DE CADA
METAHEURÍSTICA APLICADA A LAS INSTANCIAS ENTRE 60-01 Y 85-10.

Instancias	GRASP		VNS	
	Min	Med	Min	Med
60-01	3.08	3.63	4.64	5.97
60-02	59.48	60.50	62.92	63.73
60-03	4.32	4.66	6.00	6.82
60-04	30.56	31.23	31.28	32.87
60-05	1.48	2.40	3.44	4.62
60-06	2.24	3.01	2.68	3.56
60-07	1.72	2.18	2.92	3.74
60-08	3.52	4.22	6.56	7.71
60-09	2.96	3.44	3.88	5.03
60-10	2.92	3.71	9.24	10.81
65-01	6.28	7.04	7.24	7.82
65-02	59.96	60.72	62.68	62.68
65-03	5.60	6.24	9.68	11.02
65-04	26.60	27.57	27.32	29.10
65-05	4.96	5.53	5.52	7.42
65-06	5.20	5.99	8.76	10.13
65-07	5.08	5.68	7.60	9.05
65-08	5.24	5.99	8.04	9.07
65-09	2.32	2.75	3.60	4.59
65-10	4.00	4.99	7.16	9.43
70-01	7.24	8.56	11.20	11.91
70-02	58.56	60.03	61.04	62.07
70-03	10.88	11.53	12.76	13.60
70-04	10.48	11.33	11.04	11.94
70-05	12.28	12.81	14.88	15.96
70-06	3.28	3.94	6.56	7.38
70-07	8.92	9.86	12.12	13.88
70-08	8.32	8.99	9.44	10.46
70-09	6.48	7.52	7.36	10.01
70-10	7.84	8.82	11.16	13.88
75-01	10.80	11.28	11.84	13.22
75-02	106.32	108.29	109.92	113.16
75-03	15.88	16.42	17.40	18.74
75-04	17.64	19.02	18.60	20.34
75-05	8.92	9.83	10.80	11.92
75-06	9.52	10.08	12.40	13.84
75-07	14.04	14.86	14.96	16.77
75-08	17.96	19.12	20.52	21.65
75-09	6.64	7.95	9.12	10.52
75-10	11.72	12.97	16.52	18.39
80-01	6.40	7.11	7.80	8.41
80-02	110.92	112.89	113.12	115.61
80-03	13.24	13.82	15.64	16.67
80-04	34.72	35.68	34.88	36.65
80-05	19.16	20.09	23.28	24.79
80-06	9.44	10.40	12.76	14.29
80-07	15.40	16.36	19.20	20.86
80-08	13.60	14.68	14.48	15.82
80-09	86.40	87.62	89.40	92.24
80-10	12.28	13.79	15.64	16.53
85-01	6.20	7.02	7.76	8.63
85-02	29.76	30.53	32.56	34.16
85-03	36.76	37.64	38.40	41.20
85-04	23.48	24.02	26.12	27.24
85-05	50.20	51.20	52.08	53.26
85-06	61.84	62.65	61.16	62.94
85-07	11.60	12.15	13.80	14.51
85-08	19.72	20.56	21.00	22.00
85-09	55.04	56.34	55.00	56.93
85-10	22.20	23.73	25.72	27.85

Tabla II
VALORES MÍNIMOS Y PROMEDIO DE LAS 30 EJECUCIONES DE CADA
METAHEURÍSTICA APLICADA A LAS INSTANCIAS 90-*, CLÁSICAS, 200-*,
300-*, Y 400-*.

Instancias	GRASP		VNS	
	Min	Med	Min	Med
90-01	12.76	14.27	13.60	14.35
90-02	26.28	27.37	26.44	30.13
90-03	18.52	21.91	21.36	23.05
90-04	13.40	14.35	14.28	15.35
90-05	42.40	44.85	40.52	42.15
90-06	61.80	63.43	66.56	69.52
90-07	26.48	28.57	25.92	27.22
90-08	65.60	67.01	67.20	68.80
90-09	44.20	45.46	45.48	46.80
90-10	38.04	38.97	39.08	40.22
p10_93	15.88	17.53	10.72	11.87
p16_81	10.28	11.07	5.52	6.65
p19_71	9.52	10.94	7.08	7.59
p21_90	4.28	5.75	3.20	3.88
p26_82	12.80	13.53	10.16	10.98
p36_92	12.68	13.36	10.32	10.93
p41_66	6.40	7.29	6.08	7.03
p4_72	10.08	11.17	6.20	7.35
p6_76	7.28	7.83	7.16	7.58
pb_200_01	53.40	54.83	46.76	48.84
pb_200_02	20.68	22.87	14.68	15.73
pb_200_03	43.32	46.02	38.40	39.86
pb_200_04	25.64	27.65	19.72	21.25
pb_200_05	23.44	24.75	20.92	22.07
pb_200_06	24.52	25.70	20.96	22.09
pb_200_07	13.76	15.57	7.84	8.23
pb_200_08	30.28	32.29	27.24	29.33
pb_200_09	20.44	22.35	17.72	18.32
pb_200_10	34.76	37.17	34.32	35.44
pb_300_01	71.36	73.56	57.88	60.97
pb_300_02	44.16	47.69	30.00	32.87
pb_300_03	49.76	52.42	52.76	54.17
pb_300_04	55.08	57.41	44.52	47.26
pb_300_05	71.52	74.93	70.40	74.24
pb_300_06	44.16	47.46	41.12	44.09
pb_300_07	62.00	64.68	45.52	51.22
pb_300_08	38.40	41.16	27.28	29.87
pb_300_09	46.56	48.17	41.60	43.82
pb_300_10	113.16	116.20	106.48	110.25
pb_400_01	84.84	89.38	79.12	82.13
pb_400_02	70.64	74.18	65.20	67.85
pb_400_03	56.12	57.66	47.12	49.85
pb_400_04	106.04	108.47	95.44	97.38
pb_400_05	61.36	65.55	47.76	52.32
pb_400_06	87.16	89.49	72.32	75.24
pb_400_07	83.72	88.40	73.84	78.20
pb_400_08	45.80	48.24	40.60	43.71
pb_400_09	145.52	150.82	131.64	137.13
pb_400_10	126.72	132.31	120.28	124.20



GRASP tenga algo más de diversidad que VNS, al cual la heurística voraz de su fase constructiva puede estar guiando a un óptimo local donde el plan de referencia es factible. Esta explicación es consistente con los resultados de la Tabla II, donde GRASP reduce mucho su rendimiento en favor de VNS. En este caso VNS consigue prácticamente la mayoría de los mejores resultados desde p10_93 en adelante. Dado que estas instancias tienen un requerimiento de sus opciones mayor, se incrementa la importancia de encontrar una secuencia factible para el plan de referencia. Además GRASP sólo consigue uno de los mejores resultados en instancias que tengan más de 200 vehículos, lo que podría indicar que la escalabilidad del problema afecta a su rendimiento.

VI. CONCLUSIONES

En este artículo hemos propuesto cómo solucionar r-CSP usando metaheurísticas constructivas, específicamente GRASP y VNS. Para ello hemos presentado nuestra propuesta para generar instancias de r-CSP a partir de las instancias de CSP tradicionales, que hemos tomado del benchmark de referencia CSPLib. Los resultados al aplicar las metaheurísticas usando las instancias extendidas para r-CSP muestran que GRASP es más competitivo en las instancias para las cuales es más sencillo encontrar una secuencia factible para el plan de referencia. En cambio pierde efectividad gradualmente al incrementar la dificultad de las instancias y finalmente es sobrepasado por VNS.

Nuestro trabajo futuro se centrará en incluir nuevas metaheurísticas constructivas como *Ant Colony Optimization* [27], que podrían además combinarse con nuestros procedimientos de búsqueda local ya presentados. Futuras extensiones de nuestro trabajo también considerarán la definición de métricas de robustez más avanzadas que evalúen la calidad de las soluciones más allá de la violación de restricciones del CSP tradicional.

AGRADECIMIENTOS

Este trabajo está financiado por el Ministerio de Economía y Competitividad bajo los proyectos NEWSOCO (ref. TIN2015-67661-P) y FHI-SELM2 (ref. TIN2014-57497-P), incluyendo Fondos Europeos de Desarrollo Regional (ERDF).

REFERENCIAS

- [1] B. D. Parrello, W. C. Kabat, and L. Wos, "Job-shop scheduling using automated reasoning: A case study of the car-sequencing problem," *Journal of Automated Reasoning*, vol. 2, no. 1, pp. 1–42, 1986.
- [2] J. Bautista Valhondo, "Modelos y métricas para la versión robusta del car sequencing problem con flotas de vehículos especiales," *Dirección y organización*, vol. 60, no. 2016, pp. 57–65, 2016.
- [3] J. Bautista, J. Pereira, and B. Adenso-Díaz, "A GRASP approach for the extended car sequencing problem," *Journal of Scheduling*, vol. 11, no. 1, pp. 3–16, 2008.
- [4] C. C. Ribeiro, D. Aloise, T. F. Noronha, C. Rocha, and S. Urrutia, "An efficient implementation of a VNS/ILS heuristic for a real-life car sequencing problem," *European Journal of Operational Research*, vol. 191, no. 3, pp. 596–611, 2008.
- [5] U. Golle, "Fitness landscape analysis and design of metaheuristics for car sequencing," *On the Car Sequencing Problem: Analysis and Solution Methods*, p. 100, 2011.
- [6] M. Bergen, P. Van Beek, and T. Carchrae, "Constraint-based vehicle assembly line sequencing," *Advances in Artificial Intelligence*, pp. 88–99, 2001.
- [7] M. Siala, E. Hebrard, and M.-J. Huguet, "A study of constraint programming heuristics for the car-sequencing problem," *Engineering Applications of Artificial Intelligence*, vol. 38, pp. 34–44, 2015.
- [8] M. Gravel, C. Gagne, and W. L. Price, "Review and comparison of three methods for the solution of the car sequencing problem," *Journal of the Operational Research Society*, vol. 56, no. 11, pp. 1287–1295, 2005.
- [9] M. Fliedner and N. Boysen, "Solving the car sequencing problem via branch & bound," *European Journal of Operational Research*, vol. 191, no. 3, pp. 1023–1042, 2008.
- [10] J. Bautista, J. Pereira, and B. Adenso-Díaz, "A beam search approach for the optimization version of the car sequencing problem," *Annals of Operations Research*, vol. 159, no. 1, pp. 233–244, 2008.
- [11] U. Golle, F. Rothlauf, and N. Boysen, "Iterative beam search for car sequencing," *Annals of Operations Research*, vol. 226, no. 1, pp. 239–254, 2015.
- [12] J. Gottlieb, M. Puchta, and C. Solnon, "A study of greedy, local search, and ant colony optimization approaches for car sequencing problems," in *Applications of evolutionary computing*. Springer, 2003, pp. 246–257.
- [13] S. Morin, C. Gagné, and M. Gravel, "Ant colony optimization with a specialized pheromone trail for the car-sequencing problem," *European Journal of Operational Research*, vol. 197, no. 3, pp. 1185–1191, 2009.
- [14] A. Zinflou, C. Gagné, and M. Gravel, "Crossover operators for the car sequencing problem," in *Evolutionary Computation in Combinatorial Optimization*. Springer, 2007, pp. 229–239.
- [15] —, "Genetic algorithm with hybrid integer linear programming crossover operators for the car-sequencing problem," *INFOR: Information Systems and Operational Research*, vol. 48, no. 1, pp. 23–37, 2010.
- [16] I. P. Gent and T. Walsh, "CSPLib: a benchmark library for constraints," in *Principles and Practice of Constraint Programming—CP'99*. Springer, 1999, pp. 480–481.
- [17] M. Puchta and J. Gottlieb, "Solving car sequencing problems by local optimization," in *Applications of Evolutionary Computing*. Springer, 2002, pp. 132–142.
- [18] B. Estellon, F. Gardi, and K. Nouioua, "Two local search approaches for solving real-life car sequencing problems," *European Journal of Operational Research*, vol. 191, no. 3, pp. 928–944, 2008.
- [19] C. Solnon, "Combining two pheromone structures for solving the car sequencing problem with Ant Colony Optimization," *European Journal of Operational Research*, vol. 191, no. 3, pp. 1043–1055, 2008.
- [20] C. Solnon, V. D. Cung, A. Nguyen, and C. Artigues, "The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem," *European Journal of Operational Research*, vol. 191, no. 3, pp. 912–927, 2008.
- [21] M. Prandtstetter and G. R. Raidl, "An integer linear programming approach and a hybrid variable neighborhood search for the car sequencing problem," *European Journal of Operational Research*, vol. 191, no. 3, pp. 1004–1022, 2008.
- [22] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.
- [23] T. A. Feo and M. G. Resende, "Greedy randomized adaptive search procedures," *Journal of global optimization*, vol. 6, no. 2, pp. 109–133, 1995.
- [24] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.
- [25] A. A. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira, "A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems," *Operations Research Perspectives*, vol. 2, pp. 62–72, 2015.
- [26] H. De Beukelaer, G. F. Davenport, G. De Meyer, and V. Fack, "JAMES: An object-oriented Java framework for discrete optimization using local search metaheuristics," *Software: Practice and Experience*, vol. 47, no. 6, pp. 921–938, 2017.
- [27] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," *Handbook of metaheuristics*, 2010.

Secuenciación en líneas de fabricación no sincronizadas con ensamblajes

Modelado matemático y resolución heurística

Carlos Andrés

Departamento de Organización de Empresas
Universitat Politècnica de València
Valencia
candres@omp.upv.es

Julien Maheut

Departamento de Organización de Empresas
Universitat Politècnica de València
Valencia
Juma2@upvnet.upv.es

Abstract—En este trabajo se presenta el problema de establecer la secuencia en una línea de montaje no sincronizada en donde cada estación de trabajo se abastece por otra máquina. Entre cada estación de trabajo o máquina existen almacenes de capacidad limitada que pueden provocar bloqueos en las instalaciones precedentes si llegan a saturar su capacidad. Se modela el problema como un taller de flujo con ensamblajes y almacenes intermedios con el objetivo de minimizar C_{max} y se realiza un estudio sobre el rendimiento de diferentes heurísticas, presentándose una adaptación de la metaheurística VNS al problema.

Keywords—líneas de fabricación; taller de flujo; almacén limitado; heurística; Variable Neighbourhood Search

I. INTRODUCCIÓN

Desde el trabajo seminal de [1] se ha desarrollado una vastísima literatura relativa a la Programación de Producción en sistemas de fabricación y, en concreto, al estudio de modelos y algoritmos para resolver el problema de la secuencia en diferentes configuraciones. Una de las configuraciones más estudiadas es el taller de flujo o “flowshop” en el que todos los trabajos tienen la misma ruta de operaciones en las M máquinas donde se procesan y el objetivo es determinar el orden de paso de un conjunto de N trabajos por las máquinas que optimice cierta medida de eficiencia de la secuencia [2].

Dentro del conjunto de soluciones posibles a cualquier instancia de este tipo de problema, muchos trabajos se han centrado en el subconjunto formado por aquellas con la propiedad denominada “de permutación”, esto es, su secuencia de paso es idéntica en todas las máquinas [3], [4]. Esto es debido a que la solución óptima para ciertos casos de optimización es una secuencia de permutación y que, en muchos problemas reales, la disciplina FIFO es la que se aplica con más frecuencia en el transporte de los trabajos entre las máquinas debido a las políticas de transporte de los trabajos entre las máquinas.

Más recientemente, otro tipo de configuración denominada taller con ensamblajes “assembly shop” [5], [6], ha empezado a llamar la atención de los investigadores. En este tipo de problemas se considera que existe al menos una máquina o etapa que funciona como ensambladora de las partes y componentes

producidas por otras máquinas. De esta manera, no es posible empezar una operación de un trabajo en una máquina, si no han llegado a la misma todos los componentes necesarios para ejecutarla. Este tipo de configuraciones posibilita el estudio de muchas más variantes de sistemas productivos industriales que los que se estudian con el taller de flujo puesto que incide en el problema de la coordinación a nivel de programa de producción de la fabricación de los componentes necesarios para obtener un producto concreto.

En la mayoría de la investigación realizada sobre estas configuraciones se asume que existe una capacidad de almacenamiento infinito entre las máquinas del taller. Sin embargo, en los sistemas productivos reales siempre hay una limitación de capacidad debido a la necesidad física que los trabajos esperen antes de ser procesados en una máquina si ésta se encuentra ocupada procesando el trabajo previo de la secuencia. Esta limitación es más relevante cuanto mayor es el volumen o tamaño de las piezas o lotes procesados entre las máquinas. En general, la limitación que se considera en la literatura de secuenciación se establece en base al número máximo de trabajos que pueden almacenarse en cada instante entre dos máquinas (asumiendo de esta manera que todos los trabajos tienen un volumen parecido o se transportan en contenedores estandarizados). Según se recoge en [7] existe un número creciente de trabajos en los que ya se considera explícitamente la limitación de capacidad entre etapas del proceso de fabricación.

Un prometedor campo de investigación es el de la programación de unidades en líneas mixtas de fabricación y/o montaje no sincronizadas con limitación de capacidad. En esta configuración, un trabajo solamente puede ser liberado de una estación de la línea a la siguiente estación si existe capacidad de almacenamiento suficiente entre ambas estaciones. Este tipo de sistemas productivos, se pueden modelar como talleres de flujo con almacenes intermedios según se ha propuesto por [8].

El objetivo de este trabajo es profundizar en este tipo de configuraciones en las que, además de una línea de fabricación y/o montaje principal, existen una serie de líneas o estaciones de abastecimiento conectadas. Un ejemplo real de este tipo de sistemas productivos se muestra en la Fig. 1.

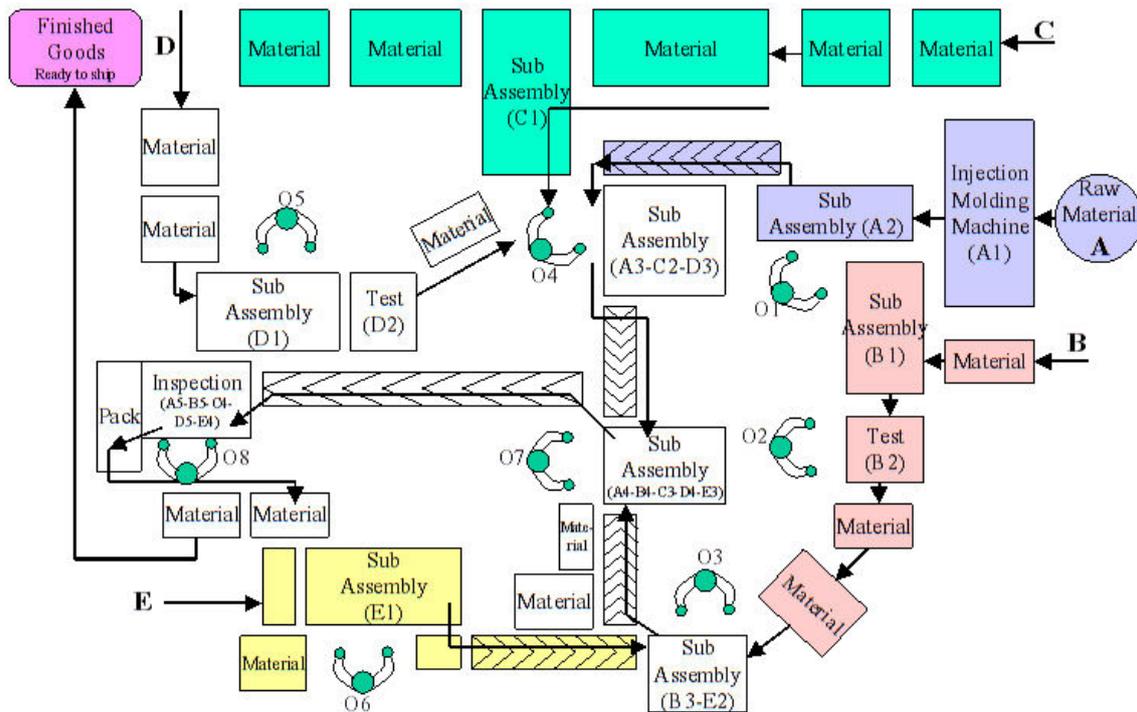


Fig. 1. Ejemplo de sistema productivo real con ensamblajes

II. PLANTEAMIENTO Y MODELADO DEL PROBLEMA

A. Descripción del problema

La configuración objeto de estudio está formada por una línea principal compuesta por M máquinas o estaciones. A cada una de estas máquinas le llegan componentes procedentes de una máquina de fabricación de componentes. Existen almacenes intermedios entre cada máquina o estación, tanto de la línea principal las máquinas de fabricación de componentes. Para poder llevar a cabo la operación correspondiente en cada estación de la línea principal es necesario disponer el producto suministrado por la estación anterior y el componente suministrado por la máquina de fabricación. En este trabajo se va a suponer solamente secuencias basadas en permutaciones, esto es, la secuencia es la misma en todas las estaciones de la línea principal y en cada máquina de fabricación de componentes.

B. Notación matemática

Para identificar los diferentes elementos del problema se usará la siguiente notación basada en la configuración mostrada en la Fig.2:

N trabajos o componentes.

M estaciones en la línea principal.

i es el subíndice asociado a los trabajos o componentes (i=1,...,N)

p es el subíndice asociado a la posición ocupada por cada trabajo (p=1,...,N)

j es el subíndice asociado a las estaciones de trabajo de la línea principal (j=1,...,M)

M_j identifica cada estación de trabajo de la línea principal (j=1,...,M). Además, se consideran M máquinas de abastecimiento que suministran MxN componentes en total debido a que solo hay una máquina abasteciendo a cada estación de trabajo. Cada una de estas máquinas se denotarán por MC_j j=1,...,M

pc_{ij} es el tiempo de proceso del componente necesario para el producto i en la máquina de abastecimiento MC_j.

p_i es el tiempo de proceso del producto i en cada estación M_j de la línea principal.

Entre cada par de estaciones de trabajo j-1 y j de la línea principal existe un almacén intermedio B_j. Del mismo modo, entre una estación de trabajo M_j y una máquina de abastecimiento de componentes MC_j existe un almacén intermedio BC_j.

De esta forma se considera que bc_j es la capacidad de almacenamiento (en unidades) del almacén que existe entre la máquina de abastecimiento MC_j y la estación de ensamblaje M_j y que b_j es la capacidad de almacenamiento (en unidades) del almacén que existe entre las estaciones de ensamblaje M_{j-1} y M_j.

SC_{pj} ≥ 0 es la fecha de inicio del componente que ocupa la posición p de la secuencia en cada máquina j de abastecimiento.

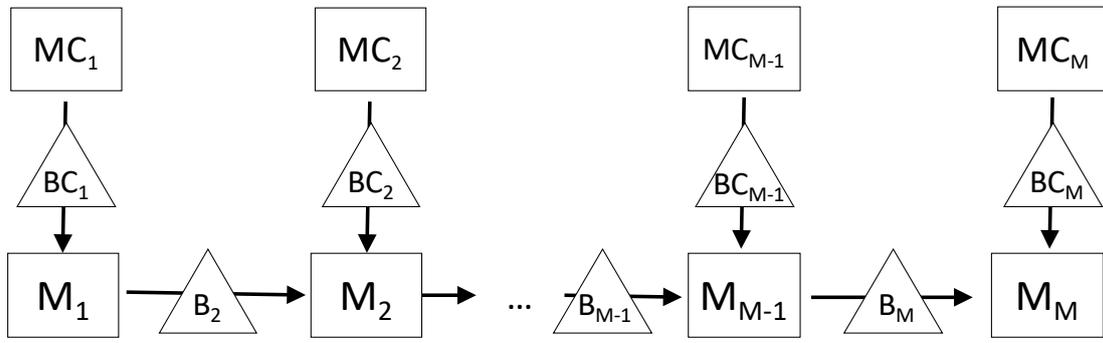


Fig. 2. Configuración estudiada

$S_{p,j} \geq 0$ es la fecha de inicio del producto que ocupa la posición p de la secuencia en cada estación j de la línea principal.

$x_{i,p}$ es una variable binaria que vale 1 si el producto i ocupa la posición p de la secuencia y 0 en caso contrario.

C. Modelo matemático

El modelo matemático de la configuración estudiada se basa en las definiciones anteriores y es el siguiente:

$$\text{Min } Cmax \tag{1}$$

s.a.

$$\sum_{i=1}^N x_{i,p} = 1 \quad p = 1, \dots, N \tag{2}$$

$$\sum_{p=1}^N x_{i,p} = 1 \quad i = 1, \dots, N \tag{3}$$

$$SC_{1,j} = 0 \quad j = 1, \dots, M \tag{4}$$

$$S_{1,1} \geq SC_{1,1} + \sum_{i=1}^N x_{i,1} \cdot pc_{i,1} \tag{5}$$

$$S_{1,j} \geq SC_{1,j} + \sum_{i=1}^N x_{i,1} \cdot pc_{i,j} \quad j = 2, \dots, M \tag{6}$$

$$S_{1,j} \geq S_{1,j-1} + \sum_{i=1}^N x_{i,1} \cdot p_{i,j-1} \quad j = 2, \dots, M \tag{7}$$

$$SC_{p,j} \geq SC_{p-1,j} + \sum_{i=1}^N x_{i,p-1} \cdot pc_{i,j} \quad \begin{matrix} p = 2, \dots, N; \\ j = 1, \dots, M \end{matrix} \tag{8}$$

$$SC_{p,j} \geq S_{p-bc_j-1,j} \quad \begin{matrix} p > bc_j + 1; \\ j = 1, \dots, M \end{matrix} \tag{9}$$

$$S_{p,1} \geq S_{p-1,1} + \sum_{i=1}^N x_{i,p-1} \cdot p_{i,1} \quad p = 2, \dots, N \tag{10}$$

$$S_{p,1} \geq SC_{p,1} + \sum_{i=1}^N x_{i,p} \cdot pc_{i,1} \quad p = 2, \dots, N \tag{11}$$

$$S_{p,1} \geq S_{p-b_2-1,2} \quad p > b_{j+1} + 1 \tag{12}$$

$$S_{p,j} \geq S_{p-1,j} + \sum_{i=1}^N x_{i,p-1} \cdot p_{i,j} \quad \begin{matrix} p = 2, \dots, N; \\ j = 2, \dots, M - 1 \end{matrix} \tag{13}$$

$$S_{p,j} \geq S_{p,j-1} + \sum_{i=1}^N x_{i,p} \cdot p_{i,j-1} \quad \begin{matrix} p = 2, \dots, N; \\ j = 2, \dots, M - 1 \end{matrix} \tag{14}$$

$$S_{p,j} \geq SC_{p,j} + \sum_{i=1}^N x_{i,p} \cdot pc_{i,j} \quad \begin{matrix} p = 2, \dots, N; \\ j = 2, \dots, M - 1 \end{matrix} \tag{15}$$

$$N \geq p \geq 2 + b_{j+1} + 1; \quad j = 2, \dots, M - 1 \tag{16}$$

$$S_{p,M} \geq S_{p-1,M} + \sum_{i=1}^N x_{i,p-1} \cdot p_{i,M} \quad p = 2, \dots, N \tag{17}$$

$$S_{p,M} \geq S_{p,M-1} + \sum_{i=1}^N x_{i,p} \cdot p_{i,M-1} \quad p = 2, \dots, N \tag{18}$$

$$S_{p,M} \geq SC_{p,M} + \sum_{i=1}^N x_{i,p} \cdot pc_{i,M} \quad p = 2, \dots, N \tag{19}$$

$$Cmax = S_{N,M} + \sum_{i=1}^N x_{i,N} \cdot p_{i,M} \tag{20}$$

La expresión (1) corresponde a la función objetivo a optimizar que en este trabajo es la fecha de fin del último trabajo secuenciado o makespan. (2) y (3) son restricciones de asignación biunívoca que obligan a que en cada posición de la secuencia haya solo un trabajo y que cada trabajo se asigne solamente a una posición. La restricción (4) define que el instante de inicio en las máquinas de abastecimiento del primer componente es cero. (5) indica que el producto en primera posición empieza en la primera estación de la línea cuando acaba la fabricación del componente en la primera máquina de abastecimiento.

Las restricciones (6) y (7) afectan a la fecha de inicio del producto que va en primera posición de la secuencia en todas las estaciones de trabajo en función de la máquina que fabrica su



componente y de la estación anterior de la línea donde se procesa.

Las ecuaciones (8) y (9) permiten calcular las fechas de inicio en las máquinas de abastecimiento de los componentes necesarios para ensamblar el resto de productos de la secuencia. En concreto, (8) se aplica a aquellos productos cuya posición en la secuencia es tal que no se ven afectados por el tamaño de los almacenes que hay entre las máquinas de abastecimiento y las máquinas de ensamblaje. En este caso, las fechas de inicio solo se ven condicionadas por la fecha de fin del componente perteneciente al producto de la posición previa de la secuencia, ya que antes de las máquinas MC_j no hay otra máquina. Sin embargo, en (9) además de este hecho se debe tener en cuenta que más allá de la posición bc_j+1 , los componentes solo pueden empezar cuando empiecen los que están bc_j posiciones antes.

Las ecuaciones (10) y (11) son similares a las dos anteriores, pero en este caso, afectan a las fechas de inicio en la primera máquina de la línea principal de los productos que están después de la posición 1. En ellas se tiene en cuenta las fechas de inicio de los trabajos que no se ven afectados por la capacidad del almacén que se encuentra entre las máquinas $M1$ y $M2$ de la línea principal (y solo dependen de la fecha de fin en la máquina de abastecimiento del componente necesario para montar el trabajo, y la fecha de fin del trabajo anterior en la $M1$).

En (12) se tiene en cuenta un término adicional que depende de la fecha en la que el trabajo que está en $b2$ posiciones antes, abandona el almacén $b2$.

Las expresiones (13) (14) (15) y (16) se aplican de la segunda máquina de la línea principal de ensamblaje en adelante. Es por ello que incorporan términos adicionales en sus expresiones para identificar las relaciones de las fechas de inicio de los trabajos en la máquina, con las fechas de fin del trabajo en la máquina anterior de la línea principal, las fechas de fin del componente necesario en la máquina de abastecimiento correspondiente, la fecha de fin del trabajo anterior en la máquina, y en el caso de la expresión (16) las relaciones en el almacén de capacidad limitada entre la máquina y la máquina siguiente.

Dado que la última máquina M de la línea principal no tiene almacén posterior, son necesarias las expresiones (17) (18) y (19) para calcular los tiempos de inicio en esa máquina. Finalmente, (20) permite calcular la fecha de fin del último trabajo de la secuencia.

III. HEURÍSTICAS Y METAHEURÍSTICA PLANTEADAS

Pese a la existencia del modelo matemático descrito en el apartado anterior, en el apartado siguiente se muestra la dificultad de abordar instancias de tamaño medio o grande. Por ello, se han propuesto varias heurísticas y una metaheurística basada en Búsqueda por Entorno Variable (VNS) para resolver el problema. A continuación, se hace una descripción de las características principales de estos métodos.

A. Heurísticas

Se han evaluado tres reglas heurísticas adaptadas al problema que nos ocupa.

1) Heurística NEH

La primera regla que se ha elegido es la conocida como regla NEH [9], puesto que ha dado muy buenos resultados en diversas variantes de flowshop [3]. Específicamente, en [10] se corrobora su bondad para configuraciones flowshop con capacidad de almacenamiento limitada.

En lo que se refiere a su funcionamiento, la regla NEH se basa en la ordenación de los trabajos en función de un criterio determinado (en el artículo original se usa la regla de mayor tiempo de proceso). En el presente trabajo, la regla de ordenación de los trabajos considera tanto los tiempos en las estaciones de ensamblaje como los de las máquinas de fabricación de componentes. Así se han tenido en cuenta los siguientes criterios de ordenación:

- MAX_T/MIN_T : Valor máximo (o mínimo) para cada trabajo entre todos los tiempos (tanto de la línea principal como las de las máquinas de fabricación de componentes).
- SUM_T : Para cada trabajo, la suma de todos los tiempos (tanto de la línea principal como las de las máquinas de fabricación de componentes).
- MAX_T_P/MIN_T_P : Valor máximo (o mínimo) para cada trabajo entre todos los tiempos de la línea principal.
- SUM_T_P/SUM_T_C : Para cada trabajo, la suma de todos los tiempos en las estaciones de ensamblaje de fabricación de componentes (o en las máquinas de fabricación de componentes).
- MAX_T_C/MIN_T_C : Valor máximo (o mínimo) para cada trabajo entre todos los tiempos de proceso en las máquinas de fabricación de componentes.

2) Heurística PF

En [11] se propuso la regla conocida como PF o Profile Fitting dentro de un caso de fabricación cíclica en un flowshop con almacenes (asimilable a una línea no sincronizada de fabricación con mezcla de modelos). Al ser una regla que tiene en cuenta los bloqueos y esperas de las máquinas se ha considerado en este trabajo puesto que se le supone un buen rendimiento en sistemas con ensamblajes donde hay muchas posiciones que originan esperas.

El procedimiento comienza eligiendo como primer trabajo de la secuencia aquel que tiene la mayor suma de tiempos de proceso. En nuestro caso optaremos por elegir el mayor de las sumas de tiempos de proceso en todas las máquinas (tanto de componentes como principales). Una vez elegido el primer trabajo i^* , se calcula para el resto de trabajos candidatos ic el valor $\sum_{j=1}^M D_{[ic]j} - D_{[i^*]j} - p_{[ic]j}$, donde $D_{[ij]} = \max(C_{[ij]}, S_{[i-b(j+1)]j+1})$. El sumatorio se extiende a todas las máquinas (tanto las de la línea principal como las de abastecimiento) y se elige aquel trabajo ic en el que se obtenga la suma menor. Este trabajo ahora se convierte en i^* y se repite el proceso hasta que se completa la secuencia.

3) Heurística MM

La última regla que se usará en esta comparación es la definida en [12] para configuraciones flowshop con bloqueo. En primer lugar, se elige en primera posición de la secuencia aquel trabajo que tenga menor tiempo de proceso en la primera máquina de la línea principal. A continuación, se coloca en última posición aquel trabajo que tenga el tiempo de proceso menor en la última máquina de la línea principal. A continuación del trabajo en primera posición (i^*) se coloca un trabajo (i_c) con el menor valor de la siguiente expresión:

$$\alpha \cdot \sum_{j=1}^{M-1} |p_{i_c,j} - p_{i^*,j+1}| - (1 - \alpha) \cdot \sum_{j=1}^M p_{i_c,j} \quad (21)$$

Donde M son las máquinas de la línea principal. Una vez colocado el trabajo elegido, este pasa ahora a ser i^* y se repite el proceso hasta que no hay más trabajos por colocar. En este trabajo se han propuesto adicionalmente dos variantes, la primera es usar, en vez de los valores de tiempos de proceso en la línea principal, el mayor entre los tiempos de proceso del trabajo i en la máquina que ensambla componentes para la máquina j y la máquina j de la línea principal. La segunda es utilizar la suma de ambos tiempos de procesado en las dos máquinas (la de fabricación de componentes y la de la línea principal).

B. Búsqueda por entorno variable (VNS)

La metaheurística VNS o búsqueda por entorno variable fue planteada por primera vez en [13] y se trata de un procedimiento de búsqueda local que usa sucesivamente diversos tipos de movimiento (con sus entornos asociados N_k) para explorar eficientemente el espacio de soluciones. La estrategia de búsqueda se basa en realizar cambios sistemáticos de entorno dentro de la búsqueda local y aumentar el tamaño del entorno cuando la búsqueda no avanza mediante una gran perturbación denominada “agitación”. Pueden plantearse diversos tipos de movimientos de agitación del mismo modo que se manejan diferentes entornos para la búsqueda local [14].

Como elemento novedoso, en este trabajo planteamos una variante de VNS a la que se añade un mecanismo de reinicio basado en frecuencias. Inicialmente se parte de una solución inicial obtenida mediante la regla NEH, sin embargo, en el momento en que se han realizado todos los movimientos según las diferentes combinaciones de movimientos de agitación y búsqueda local, se reinicia una nueva solución mediante un procedimiento constructivo. Este procedimiento genera una secuencia basada en las posiciones relativas de los trabajos que se obtienen en aquellas soluciones que, en una iteración determinada, han mejorado la mejor solución encontrada.

En este trabajo, y usando una representación del problema basada en una permutación de los trabajos, se han utilizado los siguientes movimientos de agitación: realizar varias inserciones, realizar varios swaps entre dos posiciones aleatoriamente elegidas, elegir varios trabajos en posiciones consecutivas y reescribirlos en orden inverso, y elegir varios trabajos en posiciones no consecutivas y reescribirlos en orden inverso.

En cuanto a los movimientos de búsqueda local, se han usado: inserción de un trabajo entre otros, intercambio de posiciones de dos trabajos consecutivos e intercambio de posiciones de dos trabajos no consecutivos.

El criterio de parada se ha fijado en un número de iteraciones determinado, aunque se ha monitorizado el tiempo de cálculo de cara al estudio experimental que se detalla en el siguiente apartado.

IV. ESTUDIO EXPERIMENTAL

El estudio experimental se ha desarrollado sobre tres grupos de instancias: pequeñas, medianas y grandes. Todas ellas se han generado a partir del conocido conjunto de instancias de [15] adaptándolas para el caso que nos ocupa.

Con las instancias pequeñas (9 trabajos y cinco estaciones de trabajo) se ha realizado una enumeración completa de todas las soluciones y se han representado en la Fig. 4 para poder tener una visión del aspecto que puede tener el espacio de soluciones.

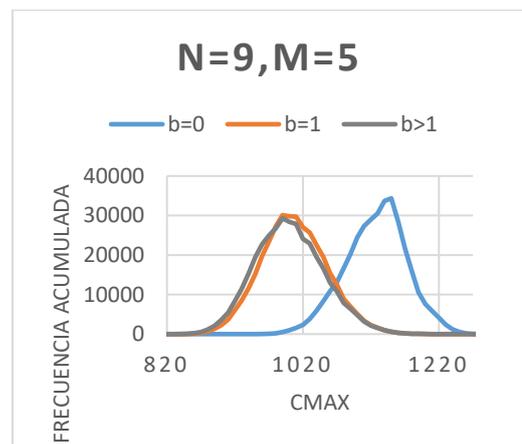


Fig. 3. Estudio de enumeración completa

Se ha observado experimentalmente una clara diferencia cuando el tamaño de los almacenes es nulo respecto a lo que sucede para tamaños de almacenes mayores. Siendo inexistente la influencia que tiene el tamaño de los almacenes intermedios para valores superiores a 1.

En el caso de las instancias medianas (instancias de 20 y 50 trabajos, y 5 y 10 estaciones de trabajo) se ha realizado un estudio experimental para evaluar hasta qué tamaño es posible resolver de manera óptima el problema. En la Tabla 1 se muestra el resultado de resolver 80 instancias diferentes con tamaños de almacén de 0 a 3.

Según la tabla, se puede concluir que el tamaño de los almacenes influye sobre la posibilidad de resolver óptimamente una instancia. Del mismo modo, el número de trabajos a secuenciar y el de máquinas también tienen influencia, siendo el primero más influyente que el segundo, puesto que, al aumentar el número de trabajo, provoca que, para un gran número de instancias, no se puedan ni siquiera calcular una relajación del modelo que permita la aplicación de las técnicas de ramificación implementadas en el optimizador

TABLA 1.



N	M	b	% Resuelto óptimamente	%GAP <0,05	0,05<%GAP <0,3	No solución
20	5	0	0%	20%	80%	0%
		1	20%	80%	0%	0%
		2	100%	0%	0%	0%
		3	80%	20%	0%	0%
	10	0	0%	100%	0%	0%
		1	0%	20%	80%	0%
		2	0%	60%	40%	0%
		3	0%	60%	40%	0%
		3	0%	60%	40%	0%
50	5	0	0%	0%	40%	60%
		1	0%	0%	20	80%
		2	0%	20%	80%	0%
		3	20%	60%	0%	20%
	10	0	0%	0%	80%	10%
		1	0%	0%	40%	60%
		2	0%	0%	40%	60%
		3	0%	0%	60%	40%
		3	0%	0%	60%	40%

Finalmente, se ha usado un conjunto de instancias grandes (200 y 500 trabajos, 5, 10 y 20 estaciones de trabajo y tamaños de almacén de 0 a 4) para evaluar el rendimiento de las heurísticas y la metaheurística VNS descrita. Se ha realizado un estudio de ajuste de parámetros en aquellas heurísticas que lo requieren de cara a definir el mejor criterio de ordenación en el caso de la regla NEH o el valor de α más adecuado en el caso de la regla MM. En la Fig. 4 se puede observar el gráfico de medias de las reglas (donde 1 es la regla MM, 2 la PF y 3 la NEH) en el que se observa la clara superioridad de la regla NEH.

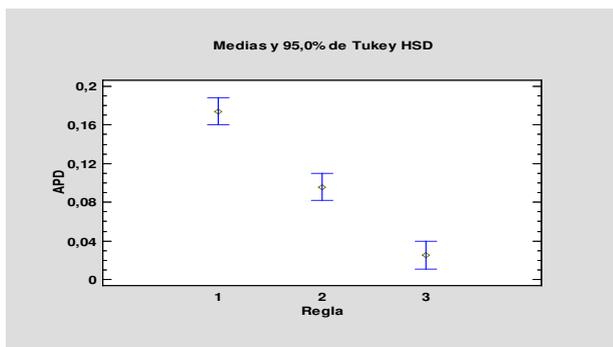


Fig. 4. Gráfico de medias

En el caso de VNS, su comportamiento es superior a la regla NEH, puesto que permite mejorar sus resultados en el 97,91% de las experiencias planteadas. En concreto, comparando las mismas instancias con diferentes tamaños de almacén, se ha observado que, el % en que VNS mejora a NEH aumenta según aumenta el tamaño de almacén (de una mejora del 0,52% respecto a la solución obtenida mediante NEH para los problemas con tamaño de almacén igual a cero hasta un 6,3% para problemas con tamaño de almacén igual a 4). Esto, indica que VNS es más efectiva respecto a NEH cuanto mayor es la capacidad de los almacenes intermedios.

Además, y basándonos ya en los resultados obtenidos por VNS, el estudio muestra que la influencia del tamaño del

almacén decrece conforme aumenta el tamaño de almacén, siendo casi despreciable a partir de tamaños superiores a 4.

V. CONCLUSIONES

En este trabajo se ha abordado el estudio de una configuración tipo flowshop con máquinas auxiliares de abastecimiento que presenta limitación de capacidad entre las diferentes máquinas del sistema. Este tipo de configuraciones son de gran interés porque permite representar tanto líneas de fabricación no sincronizadas como otros sistemas complejos en los que existen máquinas destinadas a la fabricación de componentes. Se ha propuesto un modelo matemático que permite obtener soluciones óptimas para instancias de hasta 20 trabajos. Además, se ha observado que los tamaños de los almacenes intermedios influyen sobre el problema, tanto en la topología del espacio de soluciones como en la capacidad de alcanzar buenas soluciones (o incluso óptimas) en un tiempo limitado.

REFERENCIAS

- [1] S.M. Johnson, "Optimal Two- and Three-Stage Production Schedules with Setup Times Included". *Naval Research Logistics Quarterly*, vol. 1, pp. 61–68, 1954
- [2] T. C. E. Cheng, J. Gupta y G. Wang, "A review of flowshop scheduling research with setup times". *Production and operations management*, vol. 9, pp. 262-282, 2000.
- [3] R. Ruiz y C. Maroto, "A comprehensive review and evaluation of permutation flowshop heuristics". *European Journal of Operational Research*, vol. 165, pp. 479-494, 2005
- [4] Q. Pan y R. Ruiz, "A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime". *Computers & Operations Research*, vol. 40, pp. 117-128, 2013.
- [5] C. Y. Lee, T.C.E. Cheng y B.M.T. Lin, "Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem". *Management Science*, vol. 39, pp. 616-625, 1993
- [6] C. N. Potts, S. V. Sevast'janov, V.A. Strusevich, L. N. Van Wassenhove, y C. M. Zwaneveld, "The Two-Stage Assembly Scheduling Problem: Complexity and Approximation". *Operations Research*, vol 43, pp. 346-355, 1995.
- [7] C. Andrés y J. Maheut, "Secuenciación con almacenes limitados. Una revisión de la literatura". *Dirección y Organización*, vol. 66, 2018.
- [8] J. Bautista-Valhondo y R. Alfaro-Pozo, "Mixed integer linear programming models for Flow Shop Scheduling with a demand plan of job types". *Central European Journal of Operations Research*, 2018. <https://doi.org/10.1007/s10100-018-0553-8>
- [9] M. Nawaz, E. E. Enscore e I. Ham, "A heuristic algorithm for the m-machine, n-job flowshop problem". *Omega*, vol. 11, pp.91-95, 1983.
- [10] R. Leisten, "Flowshop sequencing problems with limited buffer storage". *Int. Journal of Prod. Research*, vol. 28, pp. 2085-2100, 1990.
- [11] S. T. McCormick, M. I. Pinedo, S. Shenker, y B. Wolf, "Sequencing in an assembly line with blocking to minimize cycle time". *Operations Research*, vol. 37, pp. 925-936, 1989.
- [12] D. Ronconi, "A note on constructive heuristics for the flowshop problem with blocking". *Int. Journal of Prod. Economics*, vol. 87, pp. 39-48, 2004
- [13] N. Mladenović, P. Hansen, "Variable neighborhood search". *Computers & Operations Research*, vol. 24, pp. 1097-1100, 1997.
- [14] J. A. M. Pérez, N. Mladenović, B. M. Batista e I. J. G del Amo, "Variable Neighbourhood Search", in *Metaheuristic Procedures for Training Neural Networks*. *Op. Res. /Comp. Sc. Interfaces Series*, vol 36. E. Alba y R. Martí, Eds, Boston: Springer, 2006
- [15] E. Taillard, "Benchmarks for basic scheduling problems". *European Journal of Operational Research*, vol. 64, pp. 278-285, 1993.

Equilibrado robusto de una línea de motores mixtos con atributos temporales, espaciales y ergonómicos

Joaquín Bautista¹, Manuel Chica², Oscar Cordon², Sergio Damas²

¹IOC-ETSEIB-Nissan Chair, Universidad Politécnica de Catalunya, España

²Instituto Andaluz Interuniversitario DaSCI (Data Science and Computational Intelligence), Universidad de Granada, España
Emails: joaquin.bautista@upc.edu, manuelchica@ugr.es, ocordova@decsai.ugr.es, sdamas@ugr.es

Resumen— Partiendo de los modelos TSALBP-ergo (*Time and Space Assembly Line Balancing Problem with Ergonomic Risk*), proponemos 9 métricas para medir la robustez de un equilibrado de línea según sus atributos temporales, espaciales y contingentes. La versión robusta de TSALBP-ergo considera diversos planes de demanda e incluye funciones que miden los excesos de cargas temporales, espaciales y de riesgo ergonómico en las estaciones de la línea de montaje. Las métricas propuestas son útiles como funciones objetivo en problemas de optimización o se pueden emplear como indicadores del nivel de robustez de una línea. La nueva versión r-TSALBP-ergo pone a disposición del decisor nuevas soluciones de equilibrado más eficientes y robustas ante una demanda incierta.

Keywords—Líneas de ensamblado; Demanda incierta; Riesgo Ergonómico; Optimización robusta.

I. INTRODUCCIÓN

El diseño de una línea de montaje presenta tres vertientes como mínimo: (i) establecer el número de estaciones y decidir qué operaciones se realizarán en cada puesto de trabajo en función de la demanda, (ii) ofrecer una distribución en planta que contemple tanto el espacio requerido por el producto y sus componentes como los espacios adicionales para el desempeño correcto del trabajo de los operarios, y (iii) evitar que el reparto de cargas de trabajo entre estaciones genere lesiones físicas a los trabajadores a medio-largo plazo.

Dentro del sector de Automoción de los países de la OCDE, las líneas de montaje de modelos mixtos están capacitadas para ensamblar distintos tipos de producto, como es el caso de las plantas de motores que usan la misma instalación para fabricar distintas versiones de motores para vehículos SUVs, furgonetas y camiones.

El ensamblado de estos tipos de producto está constituido por tareas semejantes que presentan características comunes; no obstante, cada tipo de producto requiere piezas específicas, distintas formas de aplicar el trabajo, distintas herramientas y distintas habilidades y esfuerzo de los operarios. Obviamente, esta distinción se da tanto entre tipos de producto de la misma familia (v.g.- dos versiones de la familia SUVs) como entre productos de distintas familias (v.g. SUVs o camiones). En estos casos, el ensamblado de un motor requiere unas 400 tareas que pueden agregarse en unas 150 operaciones básicas ya sea por conveniencia o condicionantes de tipo tecnológico.

Brevemente, una línea de montaje es un sistema productivo constituido por un conjunto de estaciones de trabajo a las que se asigna de manera exclusiva una serie de tareas. Cada tarea requiere un tiempo de proceso (función de la actividad de los

operarios), un esfuerzo físico (según la categoría del riesgo ergonómico), y un espacio para el producto y sus componentes, la instrumentación y los recursos humanos. La definición de las tareas es el resultado de concebir la fabricación de un producto complejo como la superposición de las mismas, dando lugar a un problema denominado equilibrado de líneas de montaje (ALB: Assembly Line Balancing) [1], [2]. Las soluciones se basan en distribuir eficientemente las tareas entre estaciones, cumpliendo un conjunto de restricciones.

El problema de equilibrado de líneas en su versión simple denominado SALBP [3] (Simple Assembly Line Balancing Problem), consiste en repartir óptimamente las tareas entre las estaciones, respetando las restricciones de precedencia entre ellas, siguiendo dos criterios de optimización: (i) minimizar el número de estaciones de trabajo, dado un tiempo de ciclo de fabricación (SALBP-1), y (ii) maximizar la tasa de producción de la línea, dado un número fijo de estaciones de trabajo (SALBP-2).

TSALBP [4] (Time and Space Assembly Line Balancing Problem) es una extensión natural de SALBP más próxima a situaciones industriales realistas. En efecto, TSALBP incorpora adicionalmente atributos espaciales relacionados tanto con el espacio disponible para distribuir en planta las estaciones de trabajo como con el espacio requerido para ejecutar las tareas y para albergar los componentes de montaje del producto. TSALBP presenta 8 variantes que resultan de considerar las combinaciones de 3 criterios de optimización: m (número de estaciones), c (tiempo de ciclo) y A (área lineal disponible de las estaciones).

Por su parte, TSALBP-ergo [5] (Time and Space Assembly Line Balancing Problem with Ergonomic Risk) es una familia de problemas que incorpora a TSALBP el concepto de Riesgo Ergonómico tanto en la función objetivo como en restricciones adicionales en las estaciones de trabajo de la línea de montaje. Obviamente, TSALBP-ergo presenta 16 variantes al considerar las combinaciones de 4 criterios de optimización: los 3 criterios de TSALBP más R (riesgo ergonómico).

En este contexto, la flexibilidad en la fabricación es una propiedad importante de cara a que las grandes compañías puedan responder a cambios de su entorno económico. Entre dichos cambios está el de la variación de la demanda de sus productos. Esta variación puede ser tanto global, afectando al tiempo de ciclo de la línea, como parcial, afectando al mix de producción y a los atributos de las tareas (tiempos de proceso, áreas requeridas y riesgos ergonómicos).

La variación del mix de producción, representada a través de planes de producción diarios, obliga a reequilibrar la línea.



Esta alteración conduce normalmente a la reconfiguración de algunos o todos los puestos de trabajo, llegando a veces a ser necesaria nueva obra civil.

Para evitar cambios notables en la distribución en planta de la línea, se puede recurrir a la implantación de soluciones que sean lo más robustas posibles ante la variación de la demanda [6]-[8].

En este trabajo proponemos métricas para TSALBP-ergo, como extensión a [7], que permiten incorporar a los modelos el concepto de solución robusta [9] mediante restricciones y funciones objetivo. Para ello, las restricciones y las funciones objetivo contienen toda la información de un conjunto de referencia de planes de demanda (también llamados escenarios) que alteran los valores de los atributos temporales, espaciales y de contingencia de las tareas. Esto nos conduce a la definición de funciones de robustez temporal, espacial y de riesgo que miden las sobrecargas de tiempo, espacio y riesgo ergonómico en las estaciones, así como el grado de factibilidad del conjunto de planes de demanda.

El resto del texto sigue la siguiente estructura. La Sección II describe sucintamente la robustez en problemas ALB. En la Sección III describimos las condiciones de TSALBP-ergo ante una demanda incierta. En la Sección IV formalizamos y modelamos el problema objeto de estudio. En la Sección V presentamos algunos ejemplos ilustrativos. Por último, en la Sección VI recogemos nuestras conclusiones.

II. ROBUSTEZ EN LOS PROBLEMAS DE EQUILIBRADO DE LÍNEAS

Una forma de hallar soluciones robustas en ALBP consiste en buscar configuraciones que se ajusten a todos los escenarios posibles o, en su defecto, que ofrezcan un desempeño eficiente en todos ellos [10], [7], [11]. En [12] se presenta un modelo robusto para ALB y un programa dinámico que minimiza el tiempo de ciclo de la línea considerando el peor escenario entre los posibles. En [13], se consideran diversos escenarios que caracterizan la demanda inestable en una línea de modelos mixtos.

Otra forma de considerar la incertidumbre y la robustez en ALB es suponer que los tiempos de proceso de las tareas no son deterministas sino que están contenidos en intervalos de valores y se obtienen a partir de una función de distribución conocida. En esta vía se encuentra el trabajo [14], enfocado al SALBP-E, en el que se acotan los tiempos de proceso por intervalos y se busca un compromiso entre la minimización de la función objetivo y los valores de un ratio de estabilidad. En [15] se presentan dos modelos robustos de ALB que consideran la incertidumbre con intervalos de valores de los tiempos de proceso de las operaciones.

III. TSALBP-ERGO SUJETO A UNA DEMANDA INCIERTA

En nuestro ideario, las tareas en TSALBP-ergo poseen una serie de atributos elementales. Dichos atributos, por su distinta naturaleza, los agrupamos aquí en 3 categorías o clases: (i) temporales, (ii) espaciales y (iii) contingentes. Estos atributos elementales están vinculados a cada operación básica de forma individual.

Son atributos temporales de una tarea: el tiempo de proceso medido a actividad normal y el factor de actividad que se establece para equilibrar la línea. Por su parte, son atributos espaciales de una tarea: el área requerida por los operarios para

ejecutar cómodamente la operación, las dimensiones de los contenedores de piezas relacionadas con la operación, así como el espacio ocupado por herramientas y robots para llevar a cabo el montaje. Finalmente, son atributos contingentes de una tarea: el factor y la categoría de riesgo que supone al operario realizar una tarea, y el tiempo de exposición del operario a dicha tarea.

Cuando la línea fabrica un único tipo de producto, los datos que precisamos para resolver un problema de equilibrado son:

1. El conjunto de operaciones ($J: j = 1, \dots, |J|$) con sus atributos elementales: tiempo de proceso $t_j: j \in J$, área lineal requerida $a_j: j \in J$, categoría de riesgo $\chi_j: j \in J$, y riesgo ergonómico $R_j: j \in J$.
2. El conjunto de estaciones de trabajo ($K: k = 1, \dots, |K|$)
3. Las restricciones de precedencia, incompatibilidad entre tareas y las restricciones que afectan a los atributos de la línea: número de estaciones (m), tiempo de ciclo (c), área disponible en cada estación (A) y riesgo ergonómico permisible (R).

Cuando la línea es de modelos mixtos, los atributos elementales dependen de las tareas y productos, por tanto, teniendo en cuenta el conjunto de tipos de producto ($I: i = 1, \dots, |I|$), los atributos temporales, espaciales y contingentes adoptan respectivamente las formas siguientes: $t_{ji}, a_{ji}, R_{ji}: j \in J \wedge i \in I$.

Para simplificar, aquí supondremos $R_{ji} = \chi_j t_{ji}$ ($j \in J \wedge i \in I$), es decir: el riesgo R_{ji} depende del tiempo de proceso t_{ji} y de la categoría de riesgo χ_j de la tarea $j \in J$ (considerando solo un factor de riesgo del conjunto de factores Φ ($\varphi = 1, \dots, |\Phi|$))

Lógicamente, una línea de modelos mixtos debe atender a diversos planes de demanda de productos que simbolizamos mediante el conjunto de escenarios E . No obstante, es común realizar el equilibrado de la línea en función de un plan de demanda $\varepsilon \in E$, que definimos mediante el vector de demanda $\vec{d}_\varepsilon = (d_{1\varepsilon}, \dots, d_{|I|\varepsilon})$ o a través del vector mix de producción $\vec{\lambda}_\varepsilon = (\lambda_{1\varepsilon}, \dots, \lambda_{|I|\varepsilon})$ - donde $d_{i\varepsilon}$ es el número de unidades del tipo de producto $i \in I$ contenidas en el plan de demanda $\varepsilon \in E$, y $\lambda_{i\varepsilon}$ es la proporción del modelo $i \in I$ en el plan $\varepsilon \in E$. Aquí se cumple: $\vec{\lambda}_\varepsilon = \vec{d}_\varepsilon / D_\varepsilon$ con $D_\varepsilon = \sum_{i \in I} d_{i\varepsilon}$.

Normalmente, el equilibrado de la línea se efectúa con los valores promediados de los atributos de las tareas en función de un mix de producción. Por tanto, dado un mix de producción $\vec{\lambda}_\varepsilon$ ($\varepsilon \in E$), los tiempos de proceso, las áreas lineales y los riesgos ergonómicos de las tareas se determinan como en (1), (2) y (3), respectivamente:

$$\bar{t}_{j\varepsilon} = \sum_{i=1}^{|I|} t_{ji} \lambda_{i\varepsilon}, \forall j \in J \wedge \forall \varepsilon \in E \quad (1)$$

$$\bar{a}_{j\varepsilon} = \sum_{i=1}^{|I|} a_{ji} \lambda_{i\varepsilon}, \forall j \in J \wedge \forall \varepsilon \in E \quad (2)$$

$$\bar{R}_{j\varepsilon} = \sum_{i=1}^{|I|} R_{ji} \lambda_{i\varepsilon}, \forall j \in J \wedge \forall \varepsilon \in E \quad (3)$$

Así, dado un mix de producción de referencia $\vec{\lambda}_{\varepsilon_0}$ ($\varepsilon_0 \in E$), se dice que la línea de modelos mixtos está equilibrada cuando todas las tareas del conjunto J han sido repartidas coherentemente entre las estaciones de trabajo del conjunto K . Esto supone que a todo puesto de trabajo $k \in K$ se asigna en

exclusiva un grupo de tareas que se denomina carga de trabajo S_k , respetando, en el reparto, todas las restricciones que afectan a las estaciones. Por tanto, se cumple:

$$\bigcup_{k \in K} S_k = J \wedge (S_k \cap S_{k'} = \emptyset, \forall k \neq k') \quad (4)$$

$$\sum_{j \in S_k} \bar{t}_{j\epsilon_0} \leq c, \forall k \in K \quad (5)$$

$$\sum_{j \in S_k} \bar{a}_{j\epsilon_0} \leq A, \forall k \in K \quad (6)$$

$$\sum_{j \in S_k} \bar{R}_{j\epsilon_0} \leq R, \forall k \in K \quad (7)$$

La condición (4) representa la partición completa del conjunto de tareas J entre las estaciones del conjunto K , y las restricciones (5), (6) y (7) corresponden respectivamente a los límites que establecen el tiempo de ciclo c , el área lineal disponible A y el riesgo ergonómico permisible R , sobre los valores de los atributos que resultan tras la asignación de tareas a las estaciones de trabajo ($\forall k \in K$).

IV. INCORPORANDO ROBUSTEZ AL TSALBP-ERGO

Llamamos r-TSALB-ergo a la versión robusta de TSALB-ergo. Aquí estableceremos las restricciones del nuevo modelo para un número fijo de estaciones de trabajo, y propondremos un conjunto de métricas para la robustez de configuraciones de líneas de montaje (soluciones: ζ) en r-TSALB-ergo, frente al conjunto de planes de demanda (E).

A. Parámetros y variables

- I Conjunto de tipos de producto ($I: i = 1, \dots, |I|$)
- J, P_j Conjunto de tareas ($J: j = 1, \dots, |J|$) y Conjunto de tareas precedentes a la tarea $j \in J$ ($P_j \subseteq J$)
- K Conjunto de estaciones de trabajo ($K: k = 1, \dots, m$)
- E Conjunto de planes de demanda ($\epsilon \in E$)
- c, γ_c, Δ_c Tiempo de ciclo, flexibilidad temporal y tiempo de proceso adicional en toda estación $k \in K$. Se define ventana temporal así: $c + \Delta_c = (1 + \gamma_c)c$
- A, γ_A, Δ_A Área lineal disponible, flexibilidad espacial y área lineal adicional en toda estación $k \in K$. Se define ventana espacial así: $A + \Delta_A = (1 + \gamma_A)A$
- R, γ_R, Δ_R Riesgo ergonómico admisible en toda estación $k \in K$, flexibilidad contingente y riesgo adicional en cada estación. Se define ventana de contingencia o de riesgo así: $R + \Delta_R = (1 + \gamma_R)R$
- $\bar{t}_{j\epsilon}$ Tiempo de proceso promedio de la tarea $j \in J$ en función del mix de productos en el plan de demanda $\epsilon \in E$ – ecuación (1) -
- $\bar{a}_{j\epsilon}$ Área lineal promedio requerida por la tarea $j \in J$ en función del mix de productos en el plan de demanda $\epsilon \in E$ – ecuación (2) -
- $\chi_j, \bar{R}_{j\epsilon}$ Categoría de riesgo y riesgo ergonómico promedio asociado a la tarea $j \in J$ en función del mix de productos en el plan de demanda $\epsilon \in E$ – ecuación (3) – Supondremos $\bar{R}_{j\epsilon} = \chi_j \bar{t}_{j\epsilon} \quad \forall j \in J \wedge \forall \epsilon \in E$

- x_{jk} Variable binaria que vale 1 si la tarea $j \in J$ se asigna a la estación $k \in K$, y vale 0 en caso contrario.
- $y_{k\epsilon}^c$ Variable binaria que vale 1 si el tiempo de la carga de la estación $k \in K$, correspondiente al plan de demanda $\epsilon \in E$ y a la configuración de referencia ζ_0 de la línea, es mayor que el tiempo de ciclo – i.e. $t_\epsilon(S_k) > c$ – y vale 0 en caso contrario.
- $y_{k\epsilon}^A$ Variable binaria que vale 1 si el área lineal de la carga de la estación $k \in K$, para el plan de demanda $\epsilon \in E$ y la configuración de referencia ζ_0 de la línea, es mayor que el área lineal disponible – i.e. $a_\epsilon(S_k) > A$ – y vale 0 en caso contrario.
- $y_{k\epsilon}^R$ Variable binaria que vale 1 si el riesgo ergonómico de la carga de la estación $k \in K$, para el plan de demanda $\epsilon \in E$ y la configuración ζ_0 de la línea, es mayor que el riesgo ergonómico permisible – i.e. $R_\epsilon(S_k) > R$ – y vale 0 en caso contrario.
- Y^c Variable entera que representa el número de veces que $t_\epsilon(S_k) > c$ ($\forall \epsilon \in E, \forall k \in K$): $Y^c = \sum_{\forall \epsilon \forall k} y_{k\epsilon}^c$
- Y^A Variable entera que representa el número de veces que $a_\epsilon(S_k) > A$ ($\forall \epsilon \in E, \forall k \in K$): $Y^A = \sum_{\forall \epsilon \forall k} y_{k\epsilon}^A$
- Y^R Variable entera que representa el número de veces que $R_\epsilon(S_k) > R$ ($\forall \epsilon \in E, \forall k \in K$): $Y^R = \sum_{\forall \epsilon \forall k} y_{k\epsilon}^R$

B. Restricciones para el modelo r-TSALBP-ergo con número fijo de estaciones de trabajo

$$x_{jk} \in \{0,1\}, (\forall j \in J, k \in K) \quad (8)$$

$$y_{k\epsilon}^c \in \{0,1\}, (\forall k \in K, \epsilon \in E) \quad (9)$$

$$y_{k\epsilon}^A \in \{0,1\}, (\forall k \in K, \epsilon \in E) \quad (10)$$

$$y_{k\epsilon}^R \in \{0,1\}, (\forall k \in K, \epsilon \in E) \quad (11)$$

$$\sum_{k \in K} x_{jk} = 1, (\forall j \in J) \quad (12)$$

$$\sum_{j \in J} x_{jk} \geq 1, (\forall k \in K) \quad (13)$$

$$\sum_{k \in K} k(x_{ik} - x_{jk}) \leq 0, (\forall i \in P_j \subseteq J) \quad (14)$$

$$\sum_{j \in J} \bar{t}_{j\epsilon} x_{jk} \leq (c + \Delta_c y_{k\epsilon}^c), (\forall k \in K, \epsilon \in E) \quad (15)$$

$$\sum_{j \in J} \bar{a}_{j\epsilon} x_{jk} \leq (A + \Delta_A y_{k\epsilon}^A), (\forall k \in K, \epsilon \in E) \quad (16)$$

$$\sum_{j \in J} \bar{R}_{j\epsilon} x_{jk} \leq (R + \Delta_R y_{k\epsilon}^R), (\forall k \in K, \epsilon \in E) \quad (17)$$

Las condiciones (8)-(11) permiten forzar la integridad de las variables de decisión del modelo. Las igualdades (12) sirven para asignar cada tarea a una y solo una estación de trabajo. Las restricciones (13) obligan a que toda estación tenga al menos una tarea asignada. Las restricciones (14) aseguran la satisfacción de las precedencias entre tareas. Finalmente, las restricciones (15)-(17) limitan respectivamente el valor de los atributos temporales, espaciales y contingentes, en cada estación y en cada plan de demanda.



C. Métricas para la robustez en TSALBP-ergo

(mr.1) Proporción de planes del conjunto E satisfechos por la solución ζ_0 , según la limitación del tiempo de ciclo c.

$$r_c^1 = 1 - \frac{1}{|E|} \sum_{\varepsilon=1}^{|\mathcal{E}|} \max_{k \in K} y_{k\varepsilon}^c \quad (18)$$

(mr.2) Proporción de estaciones sin sobrecarga de trabajo. Es decir, proporción de estaciones de la línea con tiempo de carga menor o igual que el tiempo de ciclo c.

$$r_c^2 = 1 - \frac{1}{m} \sum_{k=1}^{|\mathcal{K}|} \max_{\varepsilon \in E} y_{k\varepsilon}^c \quad (19)$$

(mr.3) Proporción del tiempo de proceso adicional concedido (Δ_c) que no emplea la solución ζ_0 , Es igual a 1 cuando $r_c^1 = 1$ o $r_c^2 = 1$.

$$r_c^3 = 1 - \frac{1}{\Delta_c Y^c} \sum_{\forall \varepsilon, k} \max \left\{ 0, \sum_{\forall j} \bar{t}_{j\varepsilon} x_{jk} - c \right\} : Y^c > 0, \quad (20)$$

si $Y^c = 0 \Rightarrow r_c^3 = 1$

(mr.4) Proporción de planes del conjunto E satisfechos por la solución ζ_0 , según el área lineal disponible A.

$$r_A^1 = 1 - \frac{1}{|E|} \sum_{\varepsilon=1}^{|\mathcal{E}|} \max_{k \in K} y_{k\varepsilon}^A \quad (21)$$

(mr.5) Proporción de estaciones que requieren un área lineal menor o igual que el área lineal disponible A.

$$r_A^2 = 1 - \frac{1}{m} \sum_{k=1}^{|\mathcal{K}|} \max_{\varepsilon \in E} y_{k\varepsilon}^A \quad (22)$$

(mr.6) Proporción del área lineal adicional concedida (Δ_A) que no emplea la solución ζ_0 . Es igual a 1 cuando $r_A^1 = 1$ o $r_A^2 = 1$.

$$r_A^3 = 1 - \frac{1}{\Delta_A Y^A} \sum_{\forall \varepsilon, k} \max \left\{ 0, \sum_{\forall j} \bar{a}_{j\varepsilon} x_{jk} - A \right\} : Y^A > 0, \quad (23)$$

si $Y^A = 0 \Rightarrow r_A^3 = 1$

(mr.7) Proporción de planes del conjunto E satisfechos por la solución ζ_0 , según la limitación del riesgo R.

$$r_R^1 = 1 - \frac{1}{|E|} \sum_{\varepsilon=1}^{|\mathcal{E}|} \max_{k \in K} y_{k\varepsilon}^R \quad (24)$$

(mr.8) Proporción de estaciones sin sobre-riesgo ergonómico: proporción de estaciones cuyo riesgo ergonómico no excede al riesgo permisible R.

$$r_R^2 = 1 - \frac{1}{m} \sum_{k=1}^{|\mathcal{K}|} \max_{\varepsilon \in E} y_{k\varepsilon}^R \quad (25)$$

(mr.9) Proporción del riesgo ergonómico adicional concedido (Δ_R) que no emplea la solución ζ_0 . Es igual a 1 cuando $r_R^1 = 1$ o $r_R^2 = 1$.

$$r_R^3 = 1 - \frac{1}{\Delta_R Y^R} \sum_{\forall \varepsilon, k} \max \left\{ 0, \sum_{\forall j} \bar{R}_{j\varepsilon} x_{jk} - R \right\} : Y^R > 0, \quad (26)$$

si $Y^R = 0 \Rightarrow r_R^3 = 1$

V. EJEMPLOS ILUSTRATIVOS

A. Datos y condiciones

Adoptamos un caso de estudio de la línea de producción de motores mixtos de la fábrica de Barcelona de Nissan Motor Ibérica, evaluando la robustez de soluciones ante un conjunto de planes de demanda en un entorno industrial.

Los 7 planes seleccionados para este artículo ($\varepsilon \in E$) tienen una demanda diaria de 270 motores; los detalles se recogen en la Tabla I.

TABLA I. DEMANDAS DIARIAS POR TIPO DE MOTOR Y PLAN ($d_{i,\varepsilon}$) PARA LAS INSTANCIAS CATEGÓRICAS NISSAN-9ENG.I ($\varepsilon \in E$).

$i \in I$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$	$\varepsilon = 6$	$\varepsilon = 9$	$\varepsilon = 12$	$\varepsilon = 18$
p_1	30	30	10	50	70	24	60
p_2	30	30	10	50	70	23	60
p_3	30	30	10	50	70	23	60
p_4	30	45	60	30	15	45	30
p_5	30	45	60	30	15	45	30
p_6	30	23	30	15	8	28	8
p_7	30	23	30	15	8	28	8
p_8	30	22	30	15	7	27	7
p_9	30	22	30	15	7	27	7
SUVs	90	90	30	150	210	70	180
Vans	60	90	120	60	30	90	60
Truck	120	90	120	60	30	110	30
Total	270	270	270	270	270	270	270

En la línea se fabrican hasta 9 tipos de motores distintos (p_1, \dots, p_9) con diferentes destinos y características de montaje: los tres primeros tipos son para vehículos SUVs; los tipos p_4 y p_5 son para furgonetas (vans); y los cuatro últimos se destinan a camiones comerciales (trucks) de medio tonelaje. Para una demanda equilibrada (idéntica demanda para los 9 tipos) y un tiempo de ciclo de 3 minutos, la línea consta de 21 estaciones en serie con una longitud media de 4 metros cada una. El número de tareas elementales para fabricar un motor asciende a 380 y su agrupación da lugar a 140 tareas de montaje. En [5],[16] se pueden consultar los tiempos de proceso $\bar{t}_{j\varepsilon}$, las áreas lineales $\bar{a}_{j\varepsilon}$, las categorías de riesgo χ_j y los conjuntos de precedencias P_j de las 140 tareas de montaje.

B. Resultados

Aquí analizamos cuatro configuraciones de línea mediante la aplicación de las métricas de robustez (mr.1-mr.9):

- Dos configuraciones de línea, ζ_1 y ζ_2 , con 18 estaciones, 180s de ciclo, 5,5 metros de área lineal por estación, 500 ergo-s de riesgo permitido y con coeficientes de flexibilidad ($\gamma_c, \gamma_A, \gamma_R$) iguales al 5%. Las soluciones ζ_1 y ζ_2 se han obtenido mediante un IDEA adaptativo [7] y se muestran en las Tablas II y III.
- Dos configuraciones de línea, ζ_3 y ζ_4 , con 25 estaciones, 170s de ciclo, 3,5 metros de área lineal disponible por estación, 320 ergo-s de riesgo permitido y con coeficientes de flexibilidad ($\gamma_c, \gamma_A, \gamma_R$) iguales al 5%. Las soluciones ζ_3 y ζ_4 se muestran en las Tablas IV y V y se han obtenido mediante un algoritmo GRASP [16], cuyo desempeño ha sido contrastado con PLEM.

TABLA II. CONFIGURACIÓN ζ_1 (18, 5,5, 500, 5%) IDEA ADAPTATIVO

k	$j \in S_k$: carga de las estaciones										
1	1	7	8	9	11						
2	3	4	5	10	13	14					
3	6	12	16	17	19	20	21				
4	15	18	22	23	24	25	26	27	28		
5	2	29	30	31	32	34	36				
6	33	35	37	38	39	40	41	43			
7	42	44	45	46	47	48	49	59	60		
8	50	51	52	53	54	55	56	57	58		
9	61	62	63	64	65	66	67	70			
10	68	69	71	72	73	74	75				
11	76	77	78	79	80	81	82	83	84	86	
12	85	87	88	89	90	91	92	99			
13	93	94	95	96	98	100	101	103			
14	102	104	105	106	107	108	109	110	111	112	113
	114	115									
15	116	117	118	119	120	131					
16	121	122	123	128	132	134	135	136			
17	97	124	125	126	127	129	137	138	139		
18	130	133	140								

TABLA III. CONFIGURACIÓN ζ_2 (18, 5,5, 500, 5%) IDEA ADAPTATIVO

k	$j \in S_k$: carga de las estaciones										
1	1	3	7	10	11						
2	4	5	8	9	13	14	18				
3	6	12	15	16	17	20					
4	22	23	24	25	28	29	30	31	32	34	
5	2	33	35	36	37	38					
6	39	40	41	42	43	44	45				
7	46	47	48	49	52	53	54	55	56		
8	50	51	57	58	59	60	61	62			
9	63	64	65	66	67	68	69	71	72		
10	70	73	74	75	76	77	78				
11	21	80	81	82	83	84	85	86	87	88	89
12	19	26	90	91	92	98					
13	27	94	95	99	100	101	102	103	104	108	109
	110	111									
14	107	112	113	114	115	116	117	118			
15	105	106	119	120	121	122	123	124			
16	79	93	128	131	132	134	135	136			
17	96	97	125	126	127	129	137	138	139		
18	130	133	140								

En las Tablas VI-IX mostramos los resultados resaltables para las cuatro configuraciones. Debemos hacer notar que las cuatro configuraciones son fuertemente robustas en atributos espaciales: $y_{k\epsilon}^A = 0 (\forall k \in K, \epsilon \in E)$.

La configuración ζ_1 . (Tabla VI) es fuertemente robusta en tiempo y en cuanto a riesgo ergonómico solo hay sobrecarga en las estaciones 7 y 12, en los 7 planes, por debajo del 5% de flexibilidad concedida. La configuración ζ_2 (Tabla VII) es casi fuertemente robusta en tiempos; en riesgo solo hay exceso en la estación 7 por debajo del 5% de flexibilidad concedida.

Por su parte, las configuraciones ζ_3 y ζ_4 , aun siendo semejantes, tienen distintas propiedades (Tablas VIII y IX). En efecto, ζ_4 es fuertemente robusta en riesgo y ζ_3 casi también lo es. En cuanto al tiempo de carga, tanto ζ_3 como ζ_4 exceden levemente el tiempo de ciclo.

A modo de resumen, la Tabla X recoge los valores de las 9 métricas para las 4 configuraciones $\zeta_1 - \zeta_4 (\forall \epsilon \in E)$.

TABLA IV. CONFIGURACIÓN ζ_3 (25, 3,5, 320, 5%) GRASP_MAX

k	$j \in S_k$: carga de las estaciones										
1	1	10									
2	13	14	11	3	9	18					
3	17	15	21	20	19						
4	5	4	6	16							
5	26	27	25	23	22	7					
6	24	28	29	30	8						
7	31	32	36	33	35	37	34				
8	39	40	38	43	41	59					
9	42	44	45	46	48						
10	49	47	50	54	55	53					
11	56	52	51	60	57	64					
12	58	61	62	63	66	67					
13	70	65	68	71	72						
14	73	69	74	76							
15	79	75	77	78	81	80	82				
16	83	84	85	86	88	87	89				
17	90	91	99	92							
18	98	100	101	102	94	103					
19	106	104	108	105	113	109	111	115	114	112	107
20	12	2	116	118							
21	119	120	117	131	132	134	135				
22	121	136	93	95	122						
23	123	124	125	126	127	96	129				
24	128	137	130	133	97						
25	138	139	140	110							

TABLA V. CONFIGURACIÓN ζ_4 (25, 3,5, 320, 5%) GRASP_MED. SE MUESTRA LAS CARGA DE LAS ESTACIONES 9,10 Y 16-21. LAS CARGAS RESTANTES SON IDÉNTICAS A LAS DE LA CONFIGURACIÓN ζ_3 .

k	$j \in S_k$: carga de las estaciones										
9	42	44	45	46	47						
10	49	48	50	54	55	53					
16	83	84	85	86	88	87	91				
17	90	89	99	92							
18	98	100	101	106	103						
19	102	104	108	105	113	109	111	115	114	112	94
20	12	2	116	119							
21	107	118	120	117	131	132	134	135			

TABLA VI. TIEMPOS DE CARGA Y RIESGO DE CARGA PARA LAS ESTACIONES SATURADAS O SOBRECARGADAS DE LA CONFIGURACIÓN ζ_1 .

Tiempos de carga $t(S_k)$								
$k \setminus \epsilon$	1	2	3	6	9	12	8	
3	178,0	178,2	177,3	179,1	180,0	177,8	179,6	
12	180,0	179,6	179,5	179,8	179,9	179,7	179,6	
17	180,0	179,6	179,5	179,6	179,7	179,7	179,4	
max	180,0	179,6	179,5	179,8	180,0	179,7	179,6	
Δ_{max}	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
Riesgos de carga $R(S_k)$								
$k \setminus \epsilon$	1	2	3	6	9	12	18	
7	510,0	511,3	511,3	511,2	511,2	510,9	511,8	
12	520,0	518,8	518,4	519,3	519,3	519,1	518,9	
max	520,0	518,8	518,4	519,3	519,3	519,1	518,9	
Δ_{max}	20,00	18,83	18,43	19,31	19,31	19,11	18,92	



TABLA VII. TIEMPOS DE CARGA Y RIESGO DE CARGA PARA LAS ESTACIONES SATURADAS O SOBRECARGADAS DE LA CONFIGURACIÓN ζ_2 .

Tiempos de carga $t(S_k)$							
$k \setminus \varepsilon$	1	2	3	6	9	12	18
15	180,0	179,9	179,8	180,0	180,1	179,9	180,0
17	180,0	179,6	179,6	179,5	179,5	179,7	179,3
max	180,0	179,9	179,8	180,0	180,1	179,9	180,0
Δ_{max}	0,00	0,00	0,00	0,00	0,06	0,00	0,00
Riesgos de carga $R(S_k)$							
$k \setminus \varepsilon$	1	2	3	6	9	12	18
7	510,0	511,1	511,1	510,9	510,9	510,8	511,5
max	510,0	511,1	511,1	510,9	510,9	510,8	511,5
Δ_{max}	10,00	11,09	11,13	10,94	10,94	10,83	11,46

TABLA VIII. TIEMPOS DE CARGA Y RIESGO DE CARGA PARA LAS ESTACIONES SATURADAS O SOBRECARGADAS DE LA CONFIGURACIÓN ζ_3 .

Tiempos de carga $t(S_k)$							
$k \setminus \varepsilon$	1	2	3	6	9	12	18
25	170,0	170,8	171,5	170,0	169,4	170,8	170,1
max	170,0	170,8	171,5	170,0	169,4	170,8	170,1
Δ_{max}	0,00	0,78	1,45	0,03	0,00	0,82	0,08
Riesgos de carga $R(S_k)$							
$k \setminus \varepsilon$	1	2	3	6	9	12	18
20	320,0	320,1	319,9	320,2	320,2	320,0	320,3
max	320,0	320,1	319,9	320,2	320,2	320,0	320,3
Δ_{max}	0,00	0,09	0,00	0,20	0,20	0,01	0,32

TABLA IX. TIEMPOS DE CARGA Y RIESGO DE CARGA PARA LAS ESTACIONES SATURADAS O SOBRECARGADAS DE LA CONFIGURACIÓN ζ_4 .

Tiempos de carga $t(S_k)$							
$k \setminus \varepsilon$	1	2	3	6	9	12	18
21	170,0	170,7	170,9	170,7	170,5	170,5	171,0
25	170,0	170,8	171,5	170,0	169,4	170,8	170,1
max	170,0	170,8	171,5	170,7	170,5	170,8	171,0
Δ_{max}	0,00	0,78	1,45	0,66	0,54	0,82	0,97
Riesgos de carga $R(S_k)$							
$k \setminus \varepsilon$	1	2	3	6	9	12	18
20	320,0	318,9	318,8	319,0	319,0	319,3	318,5
max	320,0	318,9	318,8	319,0	319,0	319,3	318,5
Δ_{max}	0,00	0,00	0,00	0,00	0,00	0,00	0,00

TABLA X. VALORACIÓN SEGÚN MÉTRICAS DE ROBUSTEZ - $(mr. 1 - mr. 9)$ - DE LAS 4 CONFIGURACIONES DE LÍNEA $\zeta_1 - \zeta_4$ ($\forall \varepsilon \in E$).

	r_c^1	r_c^2	r_c^3	r_R^1	r_R^2	r_R^3	$r_A^{1,2,3}$
ζ_1	1,00	1,00	1,00	0,00	0,89	0,91	1,00
ζ_2	0,86	0,89	0,99	0,00	0,94	0,93	1,00
ζ_3	0,29	0,96	0,97	0,29	0,96	1,00	1,00
ζ_4	0,14	0,92	0,97	1,00	1,00	1,00	1,00

VI. CONCLUSIONES

Hemos definido 9 métricas de robustez para el problema TSALBP-ergo basándonos en trabajos previos para TSALBP. Nuestra propuesta, r-TSALBP-ergo, será útil en las siguientes circunstancias: (i) una línea de productos mixtos como sistema productivo, (ii) un ensamblado complejo con adiestramiento y especialización de los operarios y con tiempos de ciclo grandes (0.5'-15'), (iii) productos de grandes dimensiones (motores,

carrocerías, vehículos, etc.), (iv) lista de materiales compleja y con elevado número de componentes grandes y pesados, (v) estaciones largas (300cm-1000cm), (vi) valores de los atributos temporales, espaciales y de riesgo estables o con variabilidad controlada, y (vii) demanda variable de los productos.

En trabajos futuros, propondremos metaheurísticas y otros procedimientos híbridos, asistidos por la PLEM, para resolver diversas variantes del r-TSALBP-ergo.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad (Gobierno de España) con el proyecto TIN2014-57497-P (FHI-SELM2).

REFERENCIAS

- [1] Battaia, O., Dolgui, A., 2013. A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics* 142, 259–277.
- [2] Boysen, N., Fliedner, M., Scholl, A., 2008. Assembly line balancing: Which model to use when? *International Journal of Production Economics* 111, 509–528.
- [3] 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.
- [4] 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.
- [5] Bautista, J., Batalla-García, C., Alfaro-Pozo, R. 2016. Models for assembly line balancing by temporal, spatial and ergonomic risk attributes. *European Journal of Operational Research* 251, 814–829.
- [6] Chica, M., Cordon, O., Damas, S., Bautista, J., 2013. A robustness information and visualization model for time and space assembly line balancing under uncertain demand. *International Journal of Production Economics* 145, 761–772.
- [7] Chica, M., Bautista, J., Cordon, O., Damas, S. 2016. A multiobjective model and evolutionary algorithms for robust time and space assembly line balancing under uncertain demand. *Omega* 58, 55–68.
- [8] Chica, M., Bautista, J., de Armas, J. 2018. Benefits of robust multiobjective optimization for flexible automotive assembly line balancing. *Flexible Services and Manufacturing Journal*, 1-29. <https://doi.org/10.1007/s10696-018-9309-y>
- [9] Beyer, H., Sendhoff, B., 2007. Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering* 196, 3190–3218.
- [10] Xu, W., Xiao, T., 2011. Strategic robust mixed model assembly line balancing based on scenario planning. *Tsinghua Science & Technology* 16, 308–314.
- [11] Simaria, A.S., Zanella de Sá, M., Vilarinho, P.M., 2009. Meeting demand variation using flexible u-shaped assembly lines. *International Journal of Production Research* 47, 3937–3955.
- [12] Dolgui, A., Kovalev, S., 2012. Scenario based robust line balancing: Computational complexity. *Discrete Applied Mathematics* 160, 1955–1963.
- [13] Li, J., Gao, J., 2014. Balancing manual mixed-model assembly lines using overtime work in a demand variation environment. *International Journal of Production Research* 52, 3552–3567.
- [14] Gurevsky, E., Battaia, O., Dolgui, A., 2012. Balancing of simple assembly lines under variations of task processing times. *Annals of Operations Research* 201, 265–286.
- [15] Hazir, Ö., Dolgui, A., 2013. Assembly line balancing under uncertainty: Robust optimization models and exact solution method. *Computers & Industrial Engineering* 65, 261–267.
- [16] Bautista-Valhondo, J., Alfaro-Pozo, R. 2018. A Case Study at the Nissan Barcelona factory to minimize the ergonomic risk and its standard deviation in a mixed-model assembly line. *Progress in Artificial Intelligence*, 1-12. <https://doi.org/10.1007/s13748-018-0153-9>