IX Congreso de Ingeniería de Organización
Gijón, 8 y 9 de septiembre de 2005

Modelo de programación lógica de restricciones (CLP) para una línea de producción con posibilidad de resecuenciar considerando almacenes limitados [*]

Gerrit Färber, Anna M. Coves Moreno

Instituto de Organización y Control de Sistemas Industriales (IOC), Universidad Politécnica de Cataluña (UPC), Av. Diagonal 647, Planta 11, 08028 Barcelona, España. gerrit_faerber@gmx.de, anna.maria.coves@upc.es

Resumen

Este trabajo presenta un modelo CLP del problema de secuenciar piezas en una línea de producción, organizado como línea de flujo. Se tiene en cuenta la posibilidad de cambiar la secuencia de piezas para estaciones con acceso a un almacén intermedio o centralizado. El acceso al almacén además está restringido, debido al tamaño de las piezas.

Palabras clave: CLP; Non-permutation flowshop; Almacén limitado, intermedio y centralizado

1. Introducción

Cuando se producen variaciones del mismo producto básico en una misma línea de producción, adquiere relevancia el problema de secuenciar piezas en la línea. Estas variaciones de producto implican, en general, que el tiempo de proceso en una misma estación difiera, dependiendo de la pieza. En los últimos años aumenta cada vez más la necesidad de producir diferentes tipos de productos en una misma línea, el motivo es ofrecer una variación de productos más grande al cliente. Además el stock de productos acabados disminuye notablemente, con respeto a la fabricación por lotes, y con él sus gastos derivados. La gran mayoría de trabajos en esta área temática se limita a soluciones que determinan el orden de las piezas antes de que entren en la línea y lo mantienen sin cambiar hasta el final de la línea, conocido como línea de flujo con una permutación (*permutation flowshop*).

En el caso de más de tres estaciones y con el objetivo de minimizar el *makespan* una única permutación ya no es óptima. Potts *et al*. (1991) estudian el beneficio de usar una línea de flujo no-permutación. Existen además varios diseños de líneas que puedan permitir la resecuenciación, ya sea utilizando grandes almacenes (ASRS), desacoplando una parte del resto de la línea, Lee and Schaefer (1997), o bien almacenes con plazas escasas fuera de la línea, Lahmar *et al*. (2003), o con líneas híbridas o flexibles, Sawik (2000), o mediante la división y unión de líneas, Engström *et al*. (1996) o más raramente el cambio de los atributos de las piezas en vez de la posición en la secuencia, Rachakonda y Nagane (2000). La resecuenciación de piezas en la línea llega a ser más relevante cuando se presenta un tiempo o coste adicional, necesario, si en una estación la siguiente pieza es de otro modelo, conocido como *setup-time* y *setup-cost*, Bolat (1994).

[*] Este trabajo se deriva de la participación de sus autores en el proyecto de investigación financiado por el Ministerio de Ciencia y Tecnología con cofinanciación del Fondo de Desarrollo Regional con referencia DPI2004-03472, titulado "Diseño y equilibrio de líneas de montaje en entornos realistas".

El problema que planteamos en este trabajo considera una línea de flujo con la posibilidad de resecuenciar piezas entre estaciones consecutivas. Los almacenes están ubicados fuera de la línea y en un primer paso accesible desde una sola estación (caso del almacén intermedio). A continuación se utilizará un solo almacén, centralizado, accesible desde varias estaciones. En los dos casos se considera que una pieza, debido a su tamaño, quizás no pueda ocupar ciertas plazas del almacén ya sea intermedio o centralizado.

A continuación se presenta la descripción de la línea, seguido por la formalización del modelo con sus correspondientes parámetros y variables. El modelo se divide en tres casos: el caso sin restricciones, el caso restringido con almacén intermedio y el caso restringido utilizando un almacén centralizado. Después se presentan algunos resultados numéricos y finalmente las conclusiones junto con la discusión.

2. Descripción de la línea

El trabajo realizado se presenta bajo la modelización de la programación lógica de restricciones. Consideremos una línea de flujo en la cual las piezas (J_1, J_2, ..., J_j , ..., J_n) pasan consecutivamente por las estaciones (I_1, I_2, ..., I_i, ..., I_m), además después de determinadas estaciones se permite resecuenciar las piezas mediante un almacén (B_i), colocado fuera de la línea. Como muestra la Figura 1, los almacenes disponen de varias plazas ($B_{i,1}$, $B_{i,2}$, …) y para un diseño reducido, cada plaza está restringida por el tamaño de la pieza que pueda ser almacenada.

En un primer paso se colocan los almacenes entre estaciones consecutivas. En este caso se asigna un almacén a la estación precedente y solo puede accederse desde esta estación. Después, para un beneficio adicional, se utiliza un solo almacén centralizado, con acceso desde varias estaciones, mientras se mantienen las restricciones de tamaño de las plazas.

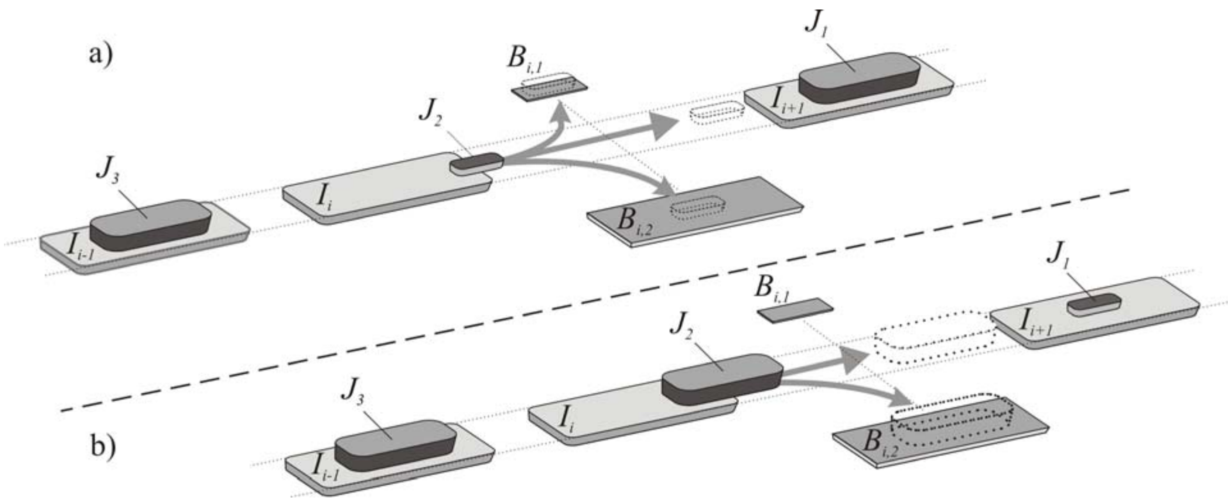

Figura 1. Esquema de la línea de flujo. Las piezas J pasan consecutivamente por las estaciones I. El almacén B_i permite almacenar una pieza temporalmente con el fin de recolocar la pieza en otra posición en la secuencia. a) La pieza J_2 puede pasar por cualquiera de las dos plazas $B_{i,1}$ o $B_{i,2}$ del almacén B_i. b) La pieza J_2 solo puede pasar por la plaza $B_{i,2}$, por tener un tamaño grande.

3. Modelo y resolución

El modelo CLP propuesto para el problema en cuestión incluye como función objetivo principalmente la minimización del *makespan*. Se presenta como minimizar el número de cambios de piezas y como incorporar *setup-cost* y *setup-time*, hecho que supondremos que ocurre cada vez que en una estación existe un cambio de modelo de la pieza para procesar.

En el caso de disponer del almacén intermedio, primero es necesario determinar las piezas que requieren ser extraídas temporalmente de la línea. Para restringir que no pueda haber dos piezas que estén guardadas en la misma plaza del almacén simultáneamente se dispone de la información de las piezas que están fuera de la línea, en una misma estación, en el momento en que se extrae una pieza específica. Una vez determinada la plaza del almacén de una estación a la cual se asigna una pieza especifica, se asegura que cada pieza que necesite ser extraída de la línea sea asignada a una plaza del almacén correspondiente, sin que el tamaño de la pieza supere el tamaño disponible.

En el caso de disponer del almacén centralizado, las restricciones además deben tener en cuenta que puede haber piezas de diferentes estaciones que están fuera de la línea en el momento de extraer alguna pieza de la línea. Las únicas variables continuas utilizadas en la formalización son el tiempo de comienzo de las piezas en cada estación, las demás variables necesarias son binarias y enteras.

3.1 Definición de parámetros y variables

Parámetros y variables comunes:

M	Número de estaciones	
i, h	Índice de estaciones	$i, h = 1, \ldots, M$
N	Número de piezas	
j, k	Índice de las piezas	$j, k = 1, \ldots, N$
$P_{i,j}$	Tiempo de proceso de pieza j en la estación i	

$P_{i,j}$ es el tiempo de proceso de la pieza j en la estación i, mientras $P_{i,[j]}$ es el tiempo de proceso de la pieza en la posición j en la estación i. Esta convención también es aplicable para las demás variables como $s_{i,[j]}$ y $c_{i,[j]}$.

$s_{i,j}$	Tiempo de inicio de la pieza j en la estación i	
$c_{i,j}$	Tiempo de terminación de la pieza j en la estación i	
$\pi_{i,j}$	Pieza j en la estación i	
π_i	Secuencia de piezas en la estación i	$\pi_i = \{\pi_{i,1}, \ldots, \pi_{i,N}\}$
$\lambda_{i,[j],[k]}$	Indica si la pieza en la posición k sucede o no a la pieza en la posición j en la estación i y también en la estación $i+1$.	$\lambda_{i,[j],[k]} \in \{0,1\}$
$\Lambda_{i,[j]}$	Indica si la pieza en la posición j requiere ser extraída o no después de la estación i con el propósito de resecuenciar.	$\Lambda_{i,[j]} \in \{0,1\}$ $i = 1, \ldots, M\text{-}1$

Parámetros y variables utilizadas para consideraciones de setup:

μ_j	Tipo de modelo de la pieza j	
$SC_{f,g,i}$	Setup-Cost que ocurre en la estación i, necesario para cambiar de modelo f a modelo g	
$ST_{f,g,i}$	Setup-Time que ocurre en la estación i, necesario para cambiar de modelo f a modelo g	

Parámetros y variables utilizadas para resecuenciar con almacenes intermedios:

D	Número máximo de plazas de los almacenes	
d	Índice de las plazas de un almacén	$d = 1, \ldots, D$
b_i	Un almacén, permitiendo resecuenciar, está colocado después de la estación i, si $b_i = 1$.	$b_i \in \{0,1\}$ $i = 1, \ldots, M\text{-}1$

ϕ_j	Tamaño físico de la pieza j	$\phi_j = 1, 2, \ldots$
$\Phi_{i,d}$	Un almacén con un máximo de D plazas, permitiendo resecuenciar, está colocado después de la estación i. El argumento de $\Phi_{i,d}$ determina el tamaño permitido de la pieza.	$\Phi_{i,d} = 0, 1, \ldots$
$\Delta_{i,[j],[k]}$	Indica si en el instante en que la pieza en la posición j es extraída de la línea después de la estación i, la pieza en la posición k está extraída o no temporalmente de la línea.	$\Delta_{i,[j],[k]} \in \{0,1\}$ $i = 1, \ldots, M\text{-}1$
$\psi_{i,d,[j]}$	Indica si la pieza en la posición j, después de salir de la estación i, está asignada o no a la plaza d del almacén respectivo.	$\psi_{i,d,[j]} \in \{0,1\}$ $i = 1, \ldots, M\text{-}1$
$\Psi_{i,[j]}$	El argumento de $\Psi_{i,[j]}$ indica la plaza del almacén al cual está asignada la pieza en la posición j después de la estación i.	$\Psi_{i,[j]} \in \{1, \ldots, D\}$ $i = 1, \ldots, M\text{-}1$

Parámetros y variables utilizadas para resecuenciar con almacenes centralizados:

b'_i	Indica si la pieza, después de salir de la estación i, puede pasar o no por el almacén centralizado con el fin de resecuenciarla.	$b'_i \in \{0,1\}$ $i = 1, \ldots, M\text{-}1$
Φ'_d	Un almacén con D plazas, está localizado como almacén centralizado, accesible desde varias estaciones. El argumento de Φ'_d determina el tamaño máximo de una pieza que pase por la plaza del almacén.	$\Phi'_d = 0, 1, \ldots$
$\Delta'_{(i-1)\cdot N+[j],\ (h-1)\cdot N+[k]}$	Indica si en el instante en que la pieza en la posición j es extraída de la línea después de la estación i, la pieza en la posición k, saliendo de la estación h, está temporalmente guardada o no en el almacén fuera de la línea.	$i,h = 1, \ldots, M\text{-}1$ $\Delta'_{(i-1)\cdot N+[j],\ (h-1)\cdot N+[k]} \in \{0,1\}$
$\psi'_{i,d,[j]}$	Indica si la pieza j, después de salir de la estación i, está asignada o no a la plaza d del almacén centralizado.	$\psi'_{i,d,[j]} \in \{0,1\}$ $i = 1, \ldots, M\text{-}1$
$\Psi'_{i,[j]}$	El argumento de $\Psi'_{i,[j]}$ indica la plaza del almacén centralizado al cual está asignada la pieza en la posición j después de salir de la estación i.	$\Psi'_{i,[j]} \in \{1, \ldots, D\}$ $i = 1, \ldots, M\text{-}1$

Pesos para la función objetivo:

α	Peso para el makespan	$\alpha = [0.0, \ldots, 1.0]$
β	Peso para el cambio de posición de una pieza	$\beta = [0.0, \ldots, 1.0]$
γ	Peso para el setup-cost	$\gamma = [0.0, \ldots, 1.0]$
δ	Peso para el setup-time	$\delta = [0.0, \ldots, 1.0]$

3.2 Resecuenciar sin restricciones (Non permutation flowshop)

Una línea de flujo clásica permite secuenciar las piezas de manera arbitraria para cada estación. En otras palabras, la secuencia de las piezas de la primera estación no necesariamente tiene que ser la misma en la segunda estación. Este problema se puede formular de la siguiente manera, utilizando programación lógica de restricciones:

$$\text{Minimizar} \qquad \left(s_{M,[N]} + P_{M,[N]} \right) \tag{1}$$

sujeto a

$$s_{i,[j]} + P_{i,[j]} \le s_{i+1,[j]} \qquad i = 1,\dots,M-1 ; \forall j \tag{2}$$

$$\begin{aligned} &\left(\left(s_{i,[j]} + P_{i,[j]} \le s_{i,[k]} \right) \wedge \left(\pi_{i,[j]} < \pi_{i,[k]} \right) \right) \\ &\quad \vee \left(\left(s_{i,[k]} + P_{i,[k]} \le s_{i,[j]} \right) \wedge \left(\pi_{i,[k]} < \pi_{i,[j]} \right) \right) \end{aligned} \qquad \forall i \,; \forall j,k \big|_{j \ne k} \tag{3}$$

$$\pi_{i,[k]} = j \quad \rightarrow \quad \pi_{i,j} = k \qquad \forall i,j,k \tag{4}$$

La función objetivo (1) minimiza el *makespan*, es decir el tiempo en que la última pieza termina ser procesada en la última estación. La restricción (2) especifica las precedencias de las estaciones, es decir después de haber terminado el procesado en la estación i la pieza tiene que trasladar a la estación i+1 (línea de flujo). La restricción (3) asegura que solamente una pieza puede ser procesada en una estación al mismo tiempo. Aparte de eso se determina $\pi_{i,[j]}$, indicando la posición de la pieza j en la estación i. Además, la restricción (4) convierte $\pi_{i,[k]}$ a $\pi_{i,j}$, indicando que en la estación i la pieza en la posición k es la pieza j.

3.2.1 Cambios de piezas innecesarios

Una de las características inherentes de líneas de flujo es que todas las piezas tienen que pasar por todas las estaciones, a pesar de que una pieza j tenga un tiempo de proceso de cero en la estación i ($P_{i,j} = 0$). En el caso en que se presenta un tiempo de proceso igual a cero, es mucho más probable que la solución encontrada contenga cambios de la secuencia innecesarios y para evitarlos, la función objetivo (1) se amplia a la función objetivo (5), utilizando el peso α para el *makespan* y el peso β para la suma de los cambios de piezas entre estaciones.

$$\text{Minimizar} \qquad \left(\alpha \cdot \left(s_{M,[N]} + P_{M,[N]} \right) + \beta \cdot \sum_{i=1}^{M} \sum_{j=1}^{N-1} \Lambda_{i,[j]} \right) \tag{5}$$

Además, para determinar si una pieza requiere ser extraída de la línea, con el fin de reinsertarla más tarde, se utilizan las dos restricciones (6) y (7). $\lambda_{i,[j],[k]}$ indica si la pieza en la posición k sucede a la pieza en la posición j en la estación i y en la estación i+1. En este caso $\lambda_{i,[j],[k]}$ es igual a 1.

$$\pi_{i+1,[\pi_{i,j}]} < \pi_{i+1,[\pi_{i,k}]} \quad \rightarrow \quad \lambda_{i,[j],[k]} = 1 \qquad i = 1,\dots,M-1 ; \forall j,k \big|_{j>k} \tag{6}$$

Después, $\Lambda_{i,[j]}$ determina si el número de piezas que suceden la pieza en la posición j en la estación i y en la estación i+1, indicadas por $\lambda_{i,[j],[k]}$, es igual al número de piezas que suceden la pieza en la posición j en la estación i. En este caso $\Lambda_{i,[j]}$ es igual a 1, indicando que la pieza j ocupa una plaza del almacén después de la estación i.

$$\sum_{k=1}^{N} \lambda_{i,[j],[k]} = N - j \quad \rightarrow \quad \Lambda_{i,[j]} = 1 \qquad i = 1,\dots,M-1 ; \forall j \tag{7}$$

3.2.2 Setup-cost

El problema de secuenciar piezas de diferentes modelos puede implicar que una estación requiera modificaciones de herramientas necesarias cuando hay un cambio de modelo. Esto debe repercudir en la función objetivo. Si el objetivo es minimizar el coste puede ocurrir que el *makespan* sea mayor.

$$\text{Minimizar} \qquad \left(\alpha \cdot \left(s_{M,[N]} + P_{M,[N]} \right) + \gamma \cdot \sum_{i=1}^{M} \sum_{j=1}^{N-1} SC_{\mu_{i,[j]},\mu_{i,[j+1]},i} \right) \tag{8}$$

Para incorporar el *setup-cost* a la función objetivo añadimos γ, un peso adicional para la suma del setup-cost $SC_{f,g,i}$.

3.2.3 Setup-time

Similar a la consideración del *setup-cost*, puede existir un tiempo adicional, cada vez que el tipo de modelo cambia en una estación. Al contrario al caso anterior, la aparición del *setup-time* influye directamente al *makespan*.

$$\text{Minimizar} \qquad \left(\alpha \cdot \left(s_{M,[N]} + P_{M,[N]} \right) + \delta \cdot \sum_{i=1}^{M} \sum_{j=1}^{N-1} ST_{\mu_{i,[j]},\mu_{i,[j+1]},i} \right) \tag{9}$$

$$\begin{aligned} &\left(\left(s_{i,[j]} + P_{i,[j]} + ST_{\mu_{i,[j]},\mu_{i,[k]},i} \le s_{i,[k]} \right) \wedge \left(\pi_{i,[j]} < \pi_{i,[k]} \right) \right) \\ &\quad \vee \left(\left(s_{i,[k]} + P_{i,[k]} + ST_{\mu_{i,[k]},\mu_{i,[j]},i} \le s_{i,[j]} \right) \wedge \left(\pi_{i,[k]} < \pi_{i,[j]} \right) \right) \end{aligned} \qquad \forall i\,; \forall j,k\big|_{j \ne k} \tag{10}$$

Para incorporar el *setup-time*, se define la función objetivo (9) como una suma ponderada del *makespan* y el *setup-time*. La restricción (3) debe sustituirse por la (10), incluyendo también el setup-time $ST_{f,g,i}$, que existe cada vez que una pieza de modelo *f* precede a una pieza de modelo *g* en la estación *i*.

3.3 Resecuenciar con restricciones

Este modelo considera la necesidad de determinar si la plaza del almacén es capaz de guardar la pieza que necesita ser extraída de la línea, considerando su tamaño físico. Este caso es más complejo que el modelo sin restricciones, visto en 3.2.

3.3.1 Almacén intermedio

El almacén intermedio está colocado entre dos estaciones consecutivas, conteniendo un número finito de plazas. Además, las plazas del almacén tienen diferentes tamaños y solo pueden almacenar piezas de tamaño menor o igual.

La función objetivo es el *makespan* y se mantiene sin cambios, como visto en 3.2. También se mantiene sin cambios las restricciones de la (2) a la (4), las precedencias de las estaciones y de las piezas. Para determinar si una pieza requiere ser extraída de la línea de producción con el fin de reinsertarla más tarde, se utilizan las dos restricciones (6) y (7).

$$\text{Minimizar} \qquad \left(s_{M,[N]} + P_{M,[N]} \right) \tag{1}$$

sujeto a

$$s_{i,[j]} + P_{i,[j]} \le s_{i+1,[j]} \qquad i = 1, \ldots, M-1; \forall j \tag{2}$$

$$\begin{array}{l}\left(\left(s_{i,[j]}+P_{i,[j]}\le s_{i,[k]}\right)\wedge\left(\pi_{i,[j]}<\pi_{i,[k]}\right)\right)\\ \quad\vee\left(\left(s_{i,[k]}+P_{i,[k]}\le s_{i,[j]}\right)\wedge\left(\pi_{i,[k]}<\pi_{i,[j]}\right)\right)\end{array}\qquad \forall i\,;\forall j,k\big|_{j\neq k} \tag{3}$$

$$\pi_{i,[k]}=j \quad\rightarrow\quad \pi_{i,j}=k \qquad \forall i,j,k \tag{4}$$

$$\pi_{i+1,[\pi_{i,j}]}<\pi_{i+1,[\pi_{i,k}]} \quad\rightarrow\quad \lambda_{i,[j],[k]}=1 \qquad i=1,\dots,M-1;\forall j,k\big|_{j>k} \tag{6}$$

$$\sum_{k=1}^{N}\lambda_{i,[j],[k]}=N-j \quad\rightarrow\quad \Lambda_{i,[j]}=1 \qquad i=1,\dots,M-1;\forall j \tag{7}$$

Una vez se han determinado las piezas que requieren ser extraídas de la línea, es necesario asegurar que en cada plaza disponible de los almacenes sólo puede ser asignada una pieza simultáneamente. Además, la pieza asignada no puede exceder al tamaño de la plaza del almacén. Se utilizan las restricciones (11) a la (15) para incluir este enfoque.

$\Delta_{i,[j],[k]}$ es igual a 1 en el caso en que la pieza en la posición k en la estación i está temporalmente fuera de la línea en el momento en que se extrae la pieza j de la línea.

$$\begin{array}{l}\left(\Lambda_{i,[j]}=1\right)\wedge\left(\Lambda_{i,[k]}=1\right)\\ \quad\wedge\left(c_{i,[j]}\ge c_{i,[k]}\right)\wedge\left(c_{i,[j]}<s_{i+1,[k]}\right)\end{array} \quad\rightarrow\quad \Delta_{i,[j],[k]}=1 \qquad i=1,\dots,M-1;\forall j,k\big|_{j\neq k} \tag{11}$$

$\psi_{i,d,[j]}$ asigna la pieza en la posición j en la estación i a la plaza d del correspondiente almacén si el tamaño de la pieza (ϕ_j) no excede el tamaño de la plaza ($\Phi_{i,d}$).

$$0\le\psi_{i,d,[j]}\cdot\left(\Phi_{i,d}-\phi_{[j]}\right) \qquad i=1,\dots,M-1;\forall d,j \tag{12}$$

Para asegurar que dos piezas no están asignadas a la misma plaza del mismo almacén simultáneamente, $\psi_{i,d,[j]}$ no puede ser 1 si la pieza en la posición k está asignada a la plaza d del almacén después de la estación i, indicado por $\psi_{i,d,[k]}$ y $\Delta_{i,[j],[k]}$.

$$\Delta_{i,[j],[k]}=1 \quad\rightarrow\quad \left(\psi_{i,d,[j]}\neq\psi_{i,d,[k]}\right)\vee\left(\psi_{i,d,[j]}=0\right) \qquad \begin{array}{l} i=1,\dots,M-1;\forall d\\ \forall j,k\big|_{j\neq k}\end{array} \tag{13}$$

La restricción (14) asegura que todas las piezas, que necesitan ser resecuenciadas, están asignadas a una plaza del correspondiente almacén.

$$\sum_{d=1}^{D}\psi_{i,d,[j]}=\Lambda_{i,[j]} \qquad i=1,\dots,M-1;\forall d,j \tag{14}$$

Finalmente, resumiendo la asignación de las piezas a los almacenes, el argumento de $\Psi_{i,[j]}$ indica la plaza del almacén al cual la pieza en la posición j está asignada después de salir de la estación i.

$$\Psi_{i,[j]}=\sum_{d=1}^{D}\left(d\cdot\psi_{i,d,[j]}\right) \qquad i=1,\dots,M-1;\forall d,j \tag{15}$$

3.3.2 Almacén centralizado

El almacén centralizado es accesible desde varias estaciones. El beneficio de la centralización está en la reducción del espacio que ocupa el almacén. Evidentemente, dos piezas no pueden ocupar la misma plaza simultáneamente.

Tanto la función objetivo, como las restricciones de las precedencias de las estaciones y de las precedencias de las piezas se mantienen, respeto al caso del almacén intermedio, de la (1) a la (4), (6) y (7). Debido al hecho de que el almacén centralizado está accesible desde varias estaciones, las restricciones de asignación a las plazas del almacén (11) a la (15) requieren modificaciones y deberán ser sustituidas por las restricciones de la (16) a la (20).

Δ' es igual a 1 en el caso en que la pieza en la posición k después de la estación h está temporalmente fuera de la línea en el momento en que se extrae la pieza j después de la estación i de la línea.

$$\begin{array}{c} \left(\Lambda_{i,[j]}=1\right)\wedge\left(\Lambda_{h,[k]}=1\right) \\ \wedge\left(c_{i,[j]}\geq c_{h,[k]}\right)\wedge\left(c_{i,[j]}<s_{h+1,[k]}\right) \end{array} \rightarrow \Delta'_{(i-1)\cdot N+[j],(h-1)\cdot N+[k]}=1 \qquad \begin{array}{l} i,h=1,\ldots,M-1\big|_{b'_1=1} \\ \forall j,k\big|_{j\neq k} \end{array} \tag{16}$$

$\psi'_{i,d,[j]}$ asigna la pieza en la posición j saliendo de la estación i a la plaza d almacén centralizado si el tamaño de la pieza (ϕ_j) no excede el tamaño de la plaza del almacén (Φ'_d).

$$0\leq\psi'_{i,d,[j]}\cdot\left(\Phi'_d-\phi_{[j]}\right) \qquad \begin{array}{l} i=1,\ldots,M-1 \\ \forall d,j \end{array} \tag{17}$$

$\psi'_{i,d,[j]}$ indica si la pieza en posición j, después de estación i, tiene que pasar o no por la plaza d del almacén centralizado. Para asegurar que dos piezas no están asignadas a la misma plaza simultáneamente, $\psi'_{i,d,[j]}$ no puede ser 1 si la pieza en posición k, después de estación h, está asignada a la plaza d del almacén centralizado, indicado por $\psi'_{i,d,[k]}$ y $\Delta'_{(i-1)\cdot N+[j],(h-1)\cdot N+[k]}$.

$$\Delta'_{(i-1)\cdot N+[j],(h-1)\cdot N+[k]}=1 \rightarrow \left(\psi'_{i,d,[j]}\neq\psi'_{h,d,[k]}\right)\vee\left(\psi'_{i,d,[j]}=0\right) \qquad \begin{array}{l} i,h=1,\ldots,M-1 \\ \forall d\,;\forall j,k\big|_{j\neq k} \end{array} \tag{18}$$

La restricción (19) asegura que todas las piezas, que necesitan ser resecuenciadas, están asignadas a una plaza del almacén centralizado.

$$\sum_{d=1}^{D}\psi'_{i,d,[j]}=\Lambda_{i,[j]} \qquad \begin{array}{l} i=1,\ldots,M-1 \\ \forall d,j \end{array} \tag{19}$$

Finalmente, resumiendo la asignación de las piezas al almacén, el argumento de $\Psi'_{i,[j]}$ indica la plaza del almacén al cual la pieza en la posición j está asignada después de salir de la estación i.

$$\Psi'_{i,[j]}=\sum_{d=1}^{D}\left(d\cdot\psi'_{i,d,[j]}\right) \qquad \begin{array}{l} i=1,\ldots,M-1 \\ \forall d,j \end{array} \tag{20}$$

4. Ejemplo numérico

Como ejemplo numérico se considera el caso de cuatro piezas que deben ser procesadas por una planta de tres estaciones. En el ejemplo no se considera *setup-time* ni *setup-cost*. Considerando consideraciones de *setup* y teniéndolo en cuenta en la función objetivo, es más probable que haya algunos cambios más. El tiempo de proceso, el tamaño físico de cada una de las piezas y el tamaño físico de las plazas del almacén intermedio se presenta en las tablas 1 hasta 3. Se colocan dos almacenes intermedios con una sola plaza después de las estaciones 2 y 3.

Tabla 1. Tiempo de proceso.

Piezas	1	2	3
$P_{1,j}$	2	8	5
$P_{2,j}$	8	4	0
$P_{3,j}$	9	1	4
$P_{4,j}$	3	6	6

Tabla 3. Tamaño físico de las plazas del almacén.

Plaza	1
$\Phi_{1,d}$	0
$\Phi_{2,d}$	2
$\Phi_{3,d}$	1

Tabla 2. Tamaño físico de las piezas.

Piezas	1	2	3
ϕ_j	2	1	3

Se ha resuelto el ejemplo utilizando la formulación de CLP presentada, realizada en OPL Studio versión 3.7. El grafo de Gant se presenta en la figura 2. En el caso en que ningún almacén es instalado en la planta, el makespan es 33. En el caso el segundo caso se presenta un cambio de posición de la pieza 1 con tamaño físico 2 después de la estación 2. El tamaño físico de la plaza del almacén permite extraer la pieza temporalmente. Con este cambio de secuencia el makespan es reducido a 31. En el caso del almacén centralizado se obtiene la misma solución presentado en la figura 3.

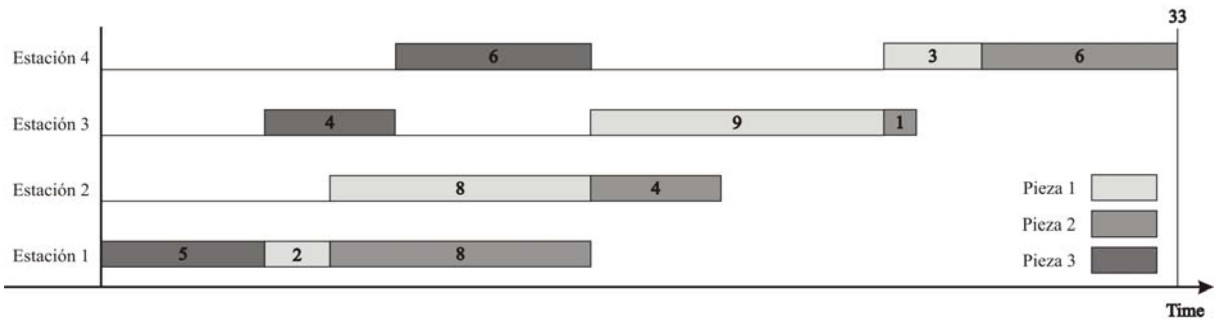

Figura 2. Grafo de Gant sin almacén, resultando en un secuencia de permutación (Makespan = 33).

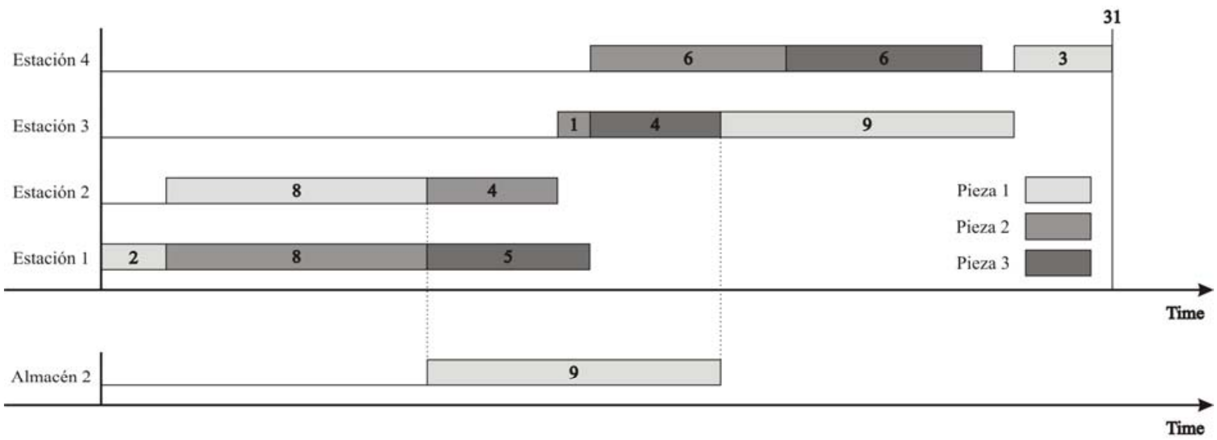

Figura 3. Grafo de Gant para dos almacenes intermedios, accesible desde las estaciones 2 y 3 (Makespan = 31).

5. Conclusiones y discusión

El modelo CLP presentado considera una línea de flujo con la posibilidad de resecuenciar piezas entre estaciones consecutivas utilizando un almacén fuera de la línea. Para minimizar el espacio de almacenes el diseño considera que cada plaza está restringida por el tamaño de la pieza que pueda ser almacenada temporalmente. En un primer caso se han colocado los almacenes entre estaciones consecutivas. Después, para un beneficio adicional del espacio dedicado a los almacenes, se ha utilizado un solo almacén centralizado, con acceso desde varias estaciones, mientras se han mantenido las restricciones de tamaño.

Para pequeñas líneas el modelo CLP se ha validado mediante OPL Studio versión 3.7 en un tiempo razonable, tanto para el caso del almacén intermedio como para el almacén centralizado. Para problemas de mayores dimensiones se ha diseñado un algoritmo genético (Genetic Algorithm), la formulación y valoración del cual está fuera del alcance de este artículo.

Referencias

Bolat, A. (1994). Sequencing jobs on an automobile assembly line: objectives and procedures. *International Journal of Production Research*, Vol. 32, No. 5, pp. 1219–1236.

Engström, T.; Jonsson, D.; Johansson, B. (1996). Alternatives to line assembly:Some swedish examples. *International Journal of Industrial Ergonomics*, Vol. 17, No. 3, pp. 235–245.

Lahmar, M.; Ergan, H.; Benjaafar, S. (2003). Resequencing and feature assignment on an automated assembly line. *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 1, pp 89–102.

Lee, H.; Schaefer, S. (1997). Sequencing methods for automated storage and retrieval systems with dedicated storage. *Computers and Industrial Engineering*, Vol. 32, No. 2, pp. 351–362.

Potts, C.; Shmoys, D.; Williamson, D. (1991). Permutation vs. non-permutation flow shop schedules. *Operations Research Letters*, Vol. 10, No. 5, pp. 281–284.

Rachakonda P; Nagane S. (2000). Simulation Study of Paint Batching Problem in Automobile Industry, http://sweb.uky.edu/~pkrach0/Projects/MFS605Project.pdf, consultado 14.07.2004

Sawik, T. (2000). Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers. *Mathematical and Computer Modelling*, Vol. 31, No. 13, pp. 39–52.