

## Modelo MILP para batching y scheduling en plantas batch multiproducto y multisitio.

Sergio Ackermann<sup>1</sup>, Yanina Fumero<sup>1</sup>, Jorge M. Montagna<sup>1</sup>

<sup>1</sup> Instituto de Desarrollo y Diseño, CONICET - UTN, Avellaneda 3657, S3002GJC Santa Fe, Argentina

{sackermann, yfumero, mmontagna}@santafe-conicet.gov.ar

**Resumen.** El problema de scheduling, importante en plantas de producción por “lotes”, usualmente ha sido resuelto de manera simplificada, teniendo como datos el número y tamaño de lotes (o batches) necesarios para satisfacer la demanda. Más recientemente, han sido desarrollados modelos que resuelven simultáneamente el problema de batching, o sea determinar el número y tamaño de los lotes, y scheduling para el caso de una planta. En este trabajo se presenta un modelo mixto entero lineal (MILP) para la resolución simultánea del batching y scheduling en el caso de instalaciones batch multisitio, en las cuales cada planta tiene varias etapas, con unidades que pueden ser distintas operando en paralelo y fuera de fase en cada una de las etapas. Se presenta un ejemplo para mostrar la eficiencia del enfoque propuesto.

**Palabras Claves:** Instalaciones multisitio, plantas batch, scheduling, batching.

### 1 Introducción

En la industria de procesos, los sistemas de producción por “lotes” generan mayor flexibilidad en el manejo de las instalaciones; por ejemplo varios productos pueden ser elaborados usando el mismo equipamiento. En estas plantas, los productos compiten por el uso de los recursos requeridos para su fabricación y, en consecuencia, el scheduling de la producción juega un rol importante en este tipo de instalaciones.

En las últimas dos décadas, varios enfoques basados en programación matemática, como así también diferentes estrategias de resolución, han sido propuestos para el problema de scheduling de la producción de plantas batch ([1], [2]). En ambientes de producción con topología secuencial, la mayoría de los modelos presentados en la literatura consideran un problema de scheduling “simplificado” donde el número y tamaños de los lotes (o batches) necesarios para satisfacer las órdenes de los clientes son conocidos. Es decir, el denominado problema de “batching” es resuelto en una primera etapa y su solución es utilizada como parámetro para resolver el modelo de scheduling, el cual consiste entonces en determinar la asignación, secuenciamiento y tiempo de los lotes que se deben procesar. Prasad y Maravelias [3] han presentado el

adfa, p. 1, 2011.

© Springer-Verlag Berlin Heidelberg 2011

primer enfoque que considera la optimización simultánea de ambas decisiones, y desde entonces algunos enfoques en esta área, con ciertas limitaciones y características, han sido presentados ([4], [5]). Más recientemente, Fumero et al. [6] han desarrollado un modelo de programación lineal mixta-entera (MILP) para optimizar simultáneamente las decisiones de batching y scheduling para una planta batch multiproducto multietapa que opera a través de campañas de producción mixtas.

En el caso más general de ambientes de producción que cuentan con múltiples plantas de producción multiproducto ubicadas en distintos sitios, resolver el problema integrado de batching y scheduling de la producción en cada planta individual no lleva, en general, a la solución óptima, ya que decidir en qué planta procesar cada orden es una parte importante del problema, por lo que es necesario plantear el problema de optimización sobre toda la instalación.

Kopanos y Puigjaner [7] desarrollaron un modelo MILP para optimizar simultáneamente las decisiones de batching y scheduling para el caso de un esquema de producción con múltiples plantas, donde cada planta está compuesta de una única etapa con diferentes unidades operando en paralelo.

Cuando las plantas cuentan con varias etapas y unidades en paralelo con diferentes tamaños y tiempos de procesamiento, la optimización simultánea resulta aún más significativa. En este trabajo se presenta un modelo MILP para la optimización simultánea del batching y scheduling de la producción en el caso de múltiples instalaciones batch multiproducto multietapa con diferentes unidades operando en paralelo y fuera de fase en cada etapa. El problema consiste en determinar el número y tamaños de lotes de cada producto que deben ser procesados para satisfacer cierta demanda, decidir en qué planta de la instalación se procesará cada lote (distintos lotes de un mismo producto pueden ser procesados en plantas diferentes), y hallar el secuenciamiento y los tiempos iniciales y finales de cada lote en las unidades correspondientes a la planta seleccionada para procesarlo, de modo que se minimice el tiempo total de procesamiento (makespan).

El trabajo es organizado de la siguiente manera. En la sección 2 se presenta la descripción del problema y se introduce la notación utilizada en la formulación matemática del modelo. El modelo MILP propuesto para resolver el problema es descrito en la sección 3, y un ejemplo ilustrativo para mostrar la eficiencia del enfoque propuesto es desarrollado en la sección siguiente. Por último, en la sección 5, se exponen las conclusiones del trabajo.

## 2 Descripción del problema

El problema que se presenta en este trabajo considera un conjunto  $N$  de plantas batch multiproducto, las cuales cuentan con el mismo conjunto  $J$  de etapas de procesamiento. Sea  $K_n$  el conjunto de todas las unidades de la planta  $n \in N$ , y  $K_{n,j}$  las unidades en la etapa  $j \in J$  de la planta  $n$ .

Para cada planta, cada etapa de la misma cuenta con unidades en paralelo, las cuales pueden ser de diferentes tamaños, operando fuera de fase.

Se debe procesar un conjunto  $I$  de productos. La demanda  $D_i$ ,  $i \in I$  es un dato del problema y puede ser satisfecha por uno o más lotes (o batches), por lo que se intro-

duce el índice  $b$  para indicar los lotes requeridos para completar la demanda del correspondiente producto.

Sea  $V_{k,n}$  el tamaño de la unidad  $k$  de la planta  $n$ . Son, además, datos del problema: el tiempo de procesamiento de cada lote del producto  $i$  en la unidad  $k$  de la planta  $n$ , indicado con  $t_{i,k,n}$ , el factor de tamaño  $SF_{i,j}$  que denota la capacidad requerida en la etapa  $j$  para producir una unidad del producto final  $i$  y el coeficiente  $\alpha_{i,k,n}$  que indica el porcentaje de utilización mínima en cada unidad de cada planta para procesar un lote del producto  $i$ . El tiempo de changeover, que indicamos con  $c_{i,i',k,n}$ , es el tiempo de transición requerido en la unidad  $k$  de la planta  $n$  para pasar de procesar un lote del producto  $i$  a uno del producto  $i'$ .

En este modelo se supone que no hay tanques de almacenamiento intermedio. Por lo tanto, teniendo en cuenta que cada lote de producto debe pasar por todas las etapas y que una vez decidido el tamaño del mismo no se puede dividir o mezclar con otro, se adopta la política de transferencia ZW (zero wait), esto es, luego de ser procesado en la etapa  $j$  el lote  $b$  debe ser inmediatamente transferido a la etapa siguiente  $j + 1$ . El tiempo de transferencia de un lote entre unidades se considera muy pequeño en comparación con los tiempos de procesamiento, y por lo tanto está incluido en dichos tiempos.

Es importante notar que tanto el número como el tamaño de los lotes de los diferentes productos a procesar son variables del problema. Se deben proponer cotas adecuadas para las mismas de modo de facilitar la solución del problema. Considerando la demanda del producto  $i$ , las dimensiones diferentes de las unidades en paralelo, el porcentaje de utilización mínima de las mismas y los factores de tamaño, podemos calcular los números mínimo y máximo de lotes de cada producto que se requieren para satisfacer la demanda.

El mínimo número de lotes de producto  $i$  está dado por

$$NBC_i^{min} = \left\lceil \frac{D_i}{B_i^{max}} \right\rceil \quad \forall i \in I \quad (1)$$

donde  $B_i^{max}$  es el máximo tamaño posible de un lote de producto  $i$ , determinado por la siguiente expresión:

$$B_i^{max} = \max_{n \in N} \left\{ \min_{j \in J} \left\{ \max_{k \in K_{n,j}} \left\{ \frac{V_{k,n}}{SF_{i,j}} \right\} \right\} \right\} \quad \forall i \in I \quad (2)$$

De la misma manera, el máximo número de lotes de producto  $i$  está dado por

$$NBC_i^{max} = \left\lfloor \alpha_{i,k,n} \frac{D_i}{B_i^{min}} \right\rfloor \quad \forall i \in I \quad (3)$$

donde  $B_i^{min}$  es el mínimo tamaño posible de un lote de producto  $i$ , dado por la siguiente expresión:

$$B_i^{min} = \min_{n \in N} \left\{ \max_{j \in J} \left\{ \min_{k \in K_{n,j}} \left\{ \frac{V_{k,n}}{SF_{i,j}} \right\} \right\} \right\} \quad \forall i \in I \quad (4)$$

Para las decisiones de scheduling del problema, se adopta una representación basada en slots de tiempo. En cada unidad se propone un número fijo de slots de tiempo a los cuales se puede asignar uno o ningún lote. Los slots cuentan con un tiempo de inicio y uno de finalización desconocidos. Si ningún lote es asignado a un slot, su tiempo final será igual al inicial, es decir el slot es de longitud 0; mientras que si se asigna un

lote de producto, el tiempo final del slot es calculado a partir de su tiempo inicial más el tiempo de procesamiento del lote asignado. En la formulación propuesta, a fin de reducir la complejidad combinatoria propia del problema abordado, se han considerado reglas válidas para la manipulación de los slots (Fumero et al. [6]). Para cada planta, se asume que cada lote procesado en la misma sea asignado al mismo slot en cada etapa de la planta. Más aún, el slot seleccionado para procesar un lote en una cierta unidad de una etapa de la planta no podrá ser utilizado en ninguna otra unidad de la misma etapa de esa planta.

A la hora de proponer el número de slots, se debe tener en cuenta que puede ocurrir el caso extremo en que todas las órdenes se procesen en una misma planta; por lo que el número de slots propuestos en cada planta es igual al número máximo de lotes en que se pueden dividir las ordenes si se procesan en forma íntegra en dicha planta.

El máximo número de lotes de producto  $i$  si estos se procesan todos en la planta  $n$  está dado por

$$NBC_{i,n}^{max} = \left\lfloor \alpha_{i,k,n} \frac{D_i}{B_{i,n}^{min}} \right\rfloor \quad \forall i \in I \quad \forall n \in N \quad (5)$$

donde  $B_{i,n}^{min}$  es el tamaño mínimo de un lote de producto  $i$  procesado en la planta  $n$  y está dado por la siguiente expresión:

$$B_{i,n}^{min} = \max_{j \in J} \left\{ \min_{k \in |K_{n,j}|} \left\{ \frac{V_{k,n}}{SF_{i,j}} \right\} \right\} \quad \forall i \in I \quad \forall n \in N \quad (6)$$

El número de slots propuestos en cada unidad de la planta  $n$  es entonces

$$L_n = \sum_{i \in I} NBC_{i,n}^{max} \quad \forall n \in N \quad (7)$$

Se determina además, mediante la siguiente expresión, el máximo tamaño posible para un lote de producto  $i$  en la planta  $n$

$$B_{i,n}^{max} = \min_{j \in J} \left\{ \max_{k \in K_{n,j}} \left\{ \frac{V_{k,n}}{SF_{i,j}} \right\} \right\} \quad \forall i \in I \quad \forall n \in N \quad (8)$$

Los tamaños mínimo ( $B_{i,n}^{min}$ ) y máximo ( $B_{i,n}^{max}$ ) para cada lote de producto  $i$  procesado en la planta  $n$  son datos necesarios en las restricciones de dimensionamiento de lotes introducidas en la siguiente sección.

### 3 Formulación del modelo

El modelo MILP que es propuesto resuelve simultáneamente el problema de batching y scheduling para instalaciones batch multisitio, donde las plantas cuentan con varias etapas y unidades no idénticas trabajando en paralelo y fuera de fase.

En la primera subsección son presentadas las ecuaciones de dimensionamiento de los lotes de acuerdo a las demandas y capacidades de las unidades en las plantas, y en las siguientes subsecciones son presentadas las ecuaciones de asignación y secuenciamiento de los lotes, las restricciones de tiempos de procesamiento de los lotes, la inclusión de la política de transferencia ZW y la función objetivo, en ese orden.

### 3.1 Restricciones de dimensionamiento de lotes

Se propone el conjunto genérico  $IB_i$  de lotes asociados al producto  $i$ , donde  $|IB_i| = NBC_i^{max}$ .

Se define la variable binaria

$$x_{i,b} = \begin{cases} 1 & \text{si el lote } b \text{ de producto } i \text{ se selecciona para satisfacer la demanda} \\ 0 & \text{en caso contrario} \end{cases}$$

Luego,

$$NBC_i^{min} \leq \sum_{b \in IB_i} x_{i,b} \leq NBC_i^{max} \quad \forall i \in I \quad (9)$$

Los lotes se seleccionan en forma creciente

$$x_{i,b+1} \leq x_{i,b} \quad \forall i \in I, \quad \forall b, b+1 \in IB_i \quad (10)$$

Se introduce además la variable binaria

$$y_{i,b,n} = \begin{cases} 1 & \text{si el lote } b \text{ de producto } i \text{ se procesa en la planta } n \\ 0 & \text{en caso contrario} \end{cases}$$

La siguiente ecuación asegura que cada lote seleccionado para satisfacer la demanda se procesa en una y solo una planta

$$\sum_{n \in N} y_{i,b,n} = x_{i,b} \quad \forall i \in I, \quad \forall b \in IB_i \quad (11)$$

Sea la variable no negativa  $B_{i,b}$  el tamaño del lote  $b$  del producto  $i$ . Las dos ecuaciones siguientes aseguran que los tamaños de los lotes estén dentro de lo admisible en las plantas y, además, imponen que el tamaño del lote sea cero si éste no fue seleccionado para satisfacer la demanda

$$B_{i,b} \leq \sum_{n \in N} B_{i,n}^{max} y_{i,b,n} \quad \forall i \in I, \quad \forall b \in IB_i \quad (12)$$

$$B_{i,b} \geq \sum_{n \in N} B_{i,n}^{min} y_{i,b,n} \quad \forall i \in I, \quad \forall b \in IB_i \quad (13)$$

La demanda debe ser satisfecha, luego

$$D_i = \sum_{b \in IB_i} B_{i,b} \quad \forall i \in I \quad (14)$$

Además, para evitar soluciones alternativas, el tamaño de los lotes seleccionados se genera en forma creciente

$$B_{i,b+1} \leq B_{i,b} \quad \forall i \in I, \quad \forall b, b+1 \in IB_i \quad (15)$$

### 3.2 Restricciones de asignación y secuenciamiento

Se introducen las variables binarias

$$z_{i,b,l,n} = \begin{cases} 1 & \text{si el lote } b \text{ de producto } i \text{ se asigna al slot } l \text{ de la planta } n \\ 0 & \text{en caso contrario} \end{cases}$$

$$X_{k,l,n} = \begin{cases} 1 & \text{si el slot } l \text{ de la unidad } k \text{ en la planta } n \text{ procesa algún lote} \\ 0 & \text{en caso contrario} \end{cases}$$

Si el lote  $b$  de producto  $i$  no se procesa en la planta  $n$ , debe ser  $z_{i,b,l,n} = 0$  para  $l = 1, 2, \dots, L_n$ ; esto es

$$z_{i,b,l,n} \leq y_{i,b,n} \quad \forall i \in I, \quad \forall b \in IB_i, \quad 1 \leq l \leq L_n \quad \forall n \in N \quad (16)$$

Cada lote procesado en una planta es asignado al mismo slot a lo largo de todas las etapas, por lo que es necesario plantear una relación lote-slot, que está dada en la siguiente ecuación.

$$\sum_{1 \leq l \leq L_n} z_{i,b,l,n} = y_{i,b,n} \quad \forall i \in I, \quad \forall b \in IB_i, \quad \forall n \in N \quad (17)$$

En cada planta, cada slot puede ser usado para procesar a lo sumo un lote. Más aún, si el slot  $l$  de una planta  $n$  es utilizado para procesar un batch sobre una unidad de la etapa  $j$ , dicho slot no se puede utilizar en las unidades restantes de esa etapa. Las dos ecuaciones siguientes imponen lo antes mencionado

$$\sum_{b \in IB_i} z_{i,b,l,n} \leq 1 \quad \forall i \in I, \quad 1 \leq l \leq L_n, \quad \forall n \in N \quad (18)$$

$$\sum_{k \in K_{n,j}} X_{k,l,n} \leq 1 \quad \forall j \in J, \quad 1 \leq l \leq L_n, \quad \forall n \in N \quad (19)$$

Los slots son utilizados en forma creciente, para evitar soluciones alternativas

$$\sum_{k \in K_{n,j}} X_{k,l+1,n} \leq \sum_{k \in K_{n,j}} X_{k,l,n} \quad \forall j \in J, \quad 1 \leq l \leq L_n - 1, \quad \forall n \in N \quad (20)$$

El número de lotes seleccionados para satisfacer la demanda y procesados en la planta  $n$  coincide con el número de slots usados en las unidades de cada etapa de esa planta

$$\sum_{k \in K_{n,j}} \sum_{1 \leq l \leq L_n} X_{k,l,n} = \sum_{i \in I} \sum_{b \in IB_i} y_{i,b,n} \quad \forall j \in J, \quad \forall n \in N \quad (21)$$

Con el fin de relacionar cada lote  $b$  con un slot  $l$  en una unidad específica  $k$ , se introduce la variable binaria

$$Y_{i,b,k,l,n} = \begin{cases} 1 & \text{si } z_{i,b,l,n} = X_{k,l,n} = 1 \\ 0 & \text{en caso contrario} \end{cases}$$

Las ecuaciones siguientes permiten determinar el valor de esa variable

$$\sum_{k \in K_{n,j}} Y_{i,b,k,l,n} = z_{i,b,l,n} \quad \forall i \in I, \quad \forall b \in IB_i, \quad \forall j \in J, \quad 1 \leq l \leq L_n - 1, \quad \forall n \in N \quad (22)$$

$$\sum_{b \in \cup_{i \in I} IB_i} Y_{i,b,k,l,n} = X_{k,l,n} \quad \forall i \in I, \quad \forall j \in J, \quad k \in K_{n,j}, \quad 1 \leq l \leq L_n - 1, \quad \forall n \in N \quad (23)$$

$$Y_{i,b,k,l,n} \geq X_{k,l,n} + z_{i,b,l,n} - 1 \quad \forall i \in I, \quad \forall b \in IB_i, \quad \forall j \in J, \quad k \in K_{n,j}, \quad 1 \leq l \leq L_n - 1, \quad \forall n \in N \quad (24)$$

### 3.3 Restricciones temporales

Las variables no negativas  $TI_{k,l,n}$  y  $TF_{k,l,n}$  representan los tiempos inicial y final de procesamiento en los slots propuestos en cada unidad de cada planta.

De este modo, considerando los tiempos de transición ya definidos  $c_{i,i',k,n}$ , los tiempos de procesamiento deben satisfacer la ecuación

$$TF_{k,l,n} = TI_{k,l,n} + \sum_{i \in I} \sum_{i' \in I} \sum_{b \in IB_i} \sum_{\substack{b' \in IB_{i'} \\ b' \neq b}} (t_{i,k,n} + c_{i,i',k,n}) Y_{i,b,k,l,n} Y_{i',b',k,l',n} \quad \forall j \in J, \forall k \in K_{n,j}, 1 \leq l < l' \leq L_n, \forall n \in N \quad (25)$$

donde  $l'$  es el primer slot efectivamente usado luego del slot  $l$  en la misma unidad.

Para evitar la no linealidad se introduce la variable no negativa

$$YY_{i,i',b,l,b',l',k,n} = \begin{cases} 1 & \text{si } Y_{i,b,k,l,n} = Y_{i',b',k,l',n} = 1 \\ 0 & \text{en caso contrario} \end{cases}$$

Las siguientes tres ecuaciones aseguran esos valores para la variable definida

$$YY_{i,i',b,l,b',l',k,n} \geq Y_{i,b,k,l,n} + Y_{i',b',k,l',n} - 1 \quad \forall i, i' \in I, \quad \forall b \in IB_i, \quad \forall b' \in IB_{i'}, \quad \forall j \in J, \forall k \in K_{n,j}, 1 \leq l < l' \leq L_n, \forall n \in N \quad (26)$$

$$YY_{i,i',b,l,b',l',k,n} \leq Y_{i,b,k,l,n} \quad \forall i, i' \in I, \quad \forall b \in IB_i, \quad \forall b' \in IB_{i'}, \quad \forall j \in J, \forall k \in K_{n,j}, 1 \leq l < l' \leq L_n, \forall n \in N \quad (27)$$

$$YY_{i,i',b,l,b',l',k,n} \leq Y_{i',b',k,l',n} \quad \forall i, i' \in I, \quad \forall b \in IB_i, \quad \forall b' \in IB_{i'}, \quad \forall j \in J, \forall k \in K_{n,j}, 1 \leq l < l' \leq L_n, \forall n \in N \quad (28)$$

Con la variable antes introducida y una representación Big-M, es posible escribir la ecuación correspondiente al tiempo de procesamiento como

$$TF_{k,l,n} - TI_{k,l,n} - \sum_{i \in I} \sum_{b \in IB_i} \left( t_{i,k,n} Y_{i,b,k,l,n} + \sum_{\substack{i' \in I \\ b' \in IB_{i'} \\ b' \neq b}} c_{i,i',k,n} YY_{i,i',b,l,b',l',k,n} \right) \geq M_1 \left( X_{k,l',n} - 1 - \sum_{l < \bar{l} < l'} X_{k,\bar{l},n} \right) \quad \forall j \in J, \forall k \in K_{n,j}, 1 \leq l < l' \leq L_n, \forall n \in N \quad (29)$$

$$-TF_{k,l,n} - TI_{k,l,n} + \sum_{i \in I} \sum_{b \in IB_i} \left( t_{i,k,n} Y_{i,b,k,l,n} + \sum_{\substack{i' \in I \\ b' \in IB_{i'} \\ b' \neq b}} c_{i,i',k,n} YY_{i,i',b,l,b',l',k,n} \right) \geq M_1 \left( X_{k,l',n} - 1 - \sum_{l < \bar{l} < l'} X_{k,\bar{l},n} \right) \quad \forall j \in J, \forall k \in K_{n,j}, 1 \leq l < l' \leq L_n, \forall n \in N \quad (30)$$

con  $M_1$  lo suficientemente grande como para que la restricción sea redundante si  $l'$  no se usa.

Notemos que las restricciones anteriores son válidas para  $l < L_n$ . Cuando  $l = L_n$  la restricción resulta

$$TF_{k,l,n} = TI_{k,l,n} + \sum_{i \in I} \sum_{b \in IB_i} t_{i,k,n} Y_{i,b,k,l,n} \quad \forall j \in J, \forall k \in K_{n,j}, l = L_n, \forall n \in N \quad (31)$$

Para evitar solapamiento de los slots, se introduce la restricción

$$TF_{k,l,n} \leq TI_{k,l+1,n} \quad \forall j \in J, \forall k \in K_{n,j}, \quad 1 \leq l, l+1 < L_n, \quad \forall n \in N \quad (32)$$

Imponemos además que si el slot  $l+1$  no se usa en la unidad  $k$ , su tiempo inicial es igual al tiempo final del slot  $l$

$$TF_{k,l,n} - TI_{k,l+1,n} \geq -M_2 X_{k,l+1,n} \quad \forall j \in J, \forall k \in K_{n,j}, \\ 1 \leq l, l+1 < L_n, \quad \forall n \in N \quad (33)$$

con  $M_2$  suficientemente grande como para que la ecuación sea redundante cuando el slot  $l+1$  se usa para procesar algún lote.

### 3.4 Política de transferencia ZW

Si  $l'$  es el primer slot efectivamente usado en la unidad  $k$  después del slot  $l$ , debe ser

$$TF_{k,l,n} - \sum_{i \in I} \sum_{i' \in I} \sum_{b \in B_i} \sum_{b' \in B_{i'}} c_{i,i',k,n} Y_{i,i',b,b',l',k,n} = TI_{k',l,n} \quad \forall j, j+1 \in J, \\ \forall k \in K_{n,j}; k' \in K_{n,j+1}, k: X_{k,l,n} = 1, k': X_{k',l,n} = 1, \\ 1 \leq l < L_n, \quad \forall n \in N \quad (34)$$

La ecuación anterior se debe satisfacer sólo cuando un batch es asignado a esas unidades ( $k$  y  $k'$ ). Será necesaria entonces la siguiente representación Big-M

$$TF_{k,l,n} - \sum_{i \in I} \sum_{i' \in I} \sum_{b \in B_i} \sum_{b' \in B_{i'}} c_{i,i',k,n} Y_{i,i',b,b',l',k,n} - TI_{k',l,n} \\ \geq M_3 \left( X_{k,l,n} + X_{k',l,n} - 2 - \sum_{l < \bar{l} < l'} X_{k,\bar{l},n} \right) \\ \forall j, j+1 \in J, \quad \forall k \in K_{n,j}, \quad \forall k' \in K_{n,j+1}, \quad 1 \leq l < L_n, \quad \forall n \in N \quad (35)$$

$$-TF_{k,l,n} + \sum_{i \in I} \sum_{i' \in I} \sum_{b \in B_i} \sum_{b' \in B_{i'}} c_{i,i',k,n} Y_{i,i',b,b',l',k,n} + TI_{k',l,n} \\ \geq M_3 \left( X_{k,l,n} + X_{k',l,n} - 2 - \sum_{l < \bar{l} < l'} X_{k,\bar{l},n} \right) \\ \forall j, j+1 \in J, \quad \forall k \in K_{n,j}, \quad \forall k' \in K_{n,j+1}, \quad 1 \leq l < L_n, \quad \forall n \in N \quad (36)$$

En caso de que  $l$  sea el último slot propuesto en la planta, es decir, si  $l = L_n$ , las restricciones resultan simplemente

$$TF_{k,l,n} - TI_{k',l,n} \geq M(X_{k,l,n} + X_{k',l,n} - 2) \quad \forall j, j+1 \in J, \\ \forall k \in K_{n,j}, \quad \forall k' \in K_{n,j+1}, l = L_n, \quad \forall n \in N \quad (37)$$

$$-TF_{k,l,n} + TI_{k',l,n} \geq M(X_{k,l,n} + X_{k',l,n} - 2) \quad \forall j, j+1 \in J, \\ \forall k \in K_{n,j}, \quad \forall k' \in K_{n,j+1}, l = L_n, \quad \forall n \in N \quad (38)$$

### 3.5 Función objetivo

Se introduce la variable continua  $MK$  para plantear la función objetivo.

Como la meta es minimizar el makespan, la restricción

$$MK \geq TF_{k,l,n} \quad \forall j \in J, \quad \forall k \in K_{n,j}, \quad 1 \leq l \leq L_n, \quad \forall n \in N \quad (39)$$

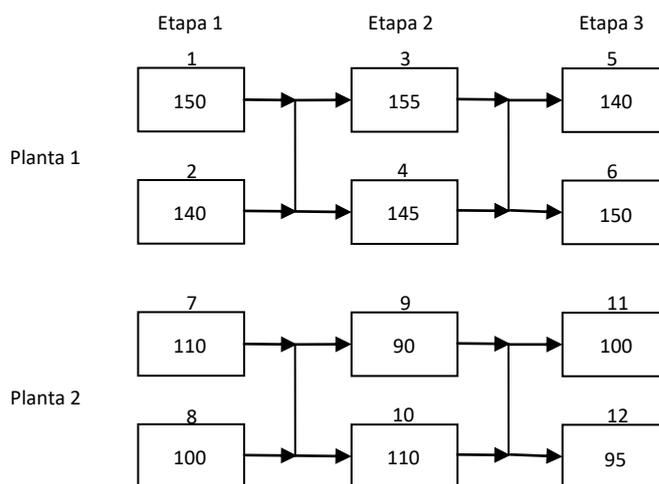
asegura que  $MK$  sea mayor o igual que el tiempo final de procesamiento de cualquier lote; con lo cual, la función objetivo es

$$\text{Minimizar } MK \quad (40)$$

sujeto a las restricciones 9-24, 26-33 y 35-39.

## 4 Ejemplo

En este ejemplo se consideran dos plantas de tres etapas cada una, con unidades de capacidades conocidas y distintas, trabajando en paralelo y fuera de fase, como se muestra en la Figura 1.



**Fig. 1.** Estructura de las plantas, numeración y capacidad de las unidades.

Las unidades correspondientes a cada planta, en cada etapa, quedan dadas por los siguientes conjuntos:  $K_{1,1} = \{1,2\}$ ,  $K_{1,2} = \{3,4\}$ ,  $K_{1,3} = \{5,6\}$ ,  $K_{2,1} = \{7,8\}$ ,  $K_{2,2} = \{9,10\}$ ,  $K_{2,3} = \{11,12\}$ . Los productos  $p_1$ ,  $p_2$ ,  $p_3$ ,  $p_4$  y  $p_5$  deben ser procesados a lo largo de las tres etapas en cualquiera de las dos plantas. Las cantidades requeridas de cada producto son:  $D_{p_1} = 250 \text{ kg}$ ,  $D_{p_2} = 180 \text{ kg}$ ,  $D_{p_3} = 230 \text{ kg}$ ,  $D_{p_4} = 190 \text{ kg}$  y  $D_{p_5} = 120 \text{ kg}$ . En este ejemplo los factores de tamaño se consideran iguales a uno para todos los productos y todas las etapas mientras que la ocupación mínima de las unidades es del 50% para todas ellas y todos los productos. Considerando lo antes

mencionado y los tamaños de las unidades, los mínimos tamaños posibles para cada producto son:

$$B_{pi}^{min} = 0.5 \min\{\max\{140, 145, 140\}, \max\{100, 90, 95\}\} = 0.5 \min\{145, 100\} = 50kg$$

para  $i = 1, 2, 3, 4, 5$ . Luego, el máximo número de lotes de cada producto está dado por

$$NBC_{p1}^{max} = \left\lfloor \frac{250}{50} \right\rfloor = 5, \quad NBC_{p2}^{max} = \left\lfloor \frac{180}{50} \right\rfloor = 3, \quad NBC_{p3}^{max} = \left\lfloor \frac{230}{50} \right\rfloor = 4,$$

$$NBC_{p4}^{max} = \left\lfloor \frac{190}{50} \right\rfloor = 3, \quad NBC_{p5}^{max} = \left\lfloor \frac{120}{50} \right\rfloor = 2.$$

Por lo tanto los lotes propuestos por el modelo son  $\{b_1, b_2, b_3, b_4, b_5\}$ ,  $\{b_6, b_7, b_8\}$ ,  $\{b_9, b_{10}, b_{11}, b_{12}\}$ ,  $\{b_{13}, b_{14}, b_{15}\}$  y  $\{b_{16}, b_{17}\}$  para los productos p1, p2, p3, p4 y p5, respectivamente.

La Tabla 1 presenta los tiempos de procesamiento, en horas, para los distintos productos, en cada unidad de cada planta.

**Tabla 1.** Tiempos de procesamiento de los productos en cada unidad

| Producto | Planta 1 |   |         |   |         |   | Planta 2 |    |         |    |         |    |
|----------|----------|---|---------|---|---------|---|----------|----|---------|----|---------|----|
|          | Etapa 1  |   | Etapa 2 |   | Etapa 3 |   | Etapa 1  |    | Etapa 2 |    | Etapa 3 |    |
|          | 1        | 2 | 3       | 4 | 5       | 6 | 7        | 8  | 9       | 10 | 11      | 12 |
| p1       | 6        | 5 | 8       | 9 | 6       | 8 | 10       | 6  | 12      | 10 | 8       | 9  |
| p2       | 7        | 9 | 6       | 7 | 8       | 8 | 10       | 9  | 10      | 11 | 10      | 8  |
| p3       | 9        | 8 | 8       | 9 | 11      | 9 | 8        | 8  | 9       | 8  | 7       | 8  |
| p4       | 8        | 9 | 9       | 8 | 10      | 9 | 9        | 9  | 7       | 8  | 8       | 10 |
| p5       | 9        | 8 | 8       | 9 | 7       | 8 | 12       | 11 | 8       | 9  | 10      | 9  |

Fueron considerados además tiempos de changeover dependientes de la secuencia de productos y de las unidades utilizadas, que no se incluyen por falta de espacio.

El modelo descripto fue implementado y resuelto en GAMS, mediante el resolvidor CPLEX – versión 12.6.3, en un procesador Intel Core i5, 2.66 GHz, 12 GB RAM.

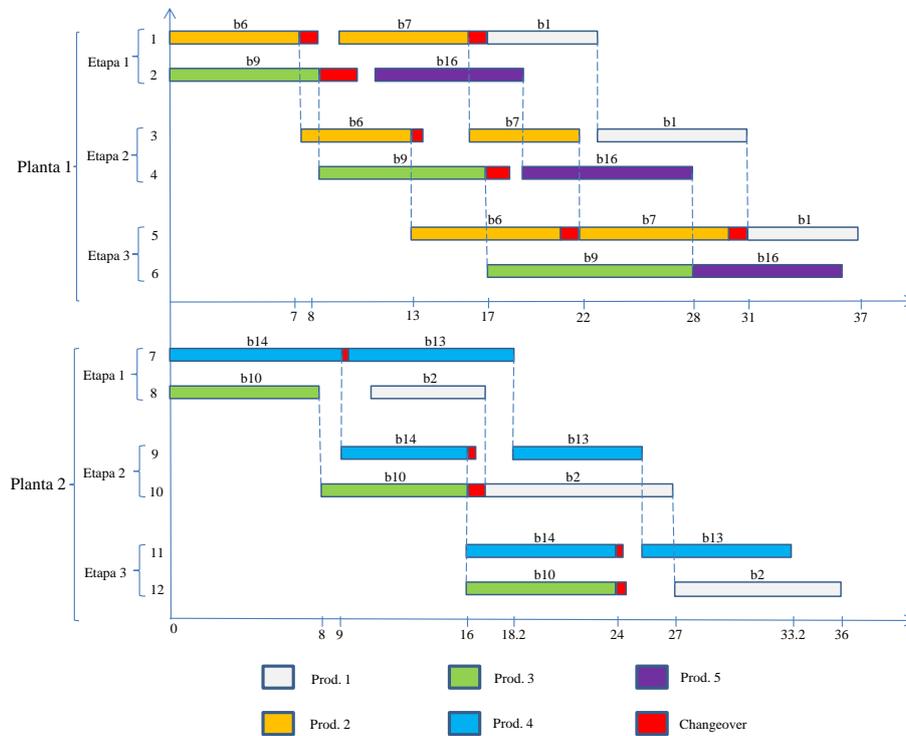
El problema generado tiene 107350 restricciones, 36120 variables continuas y 270 variables binarias. La resolución del mismo demandó un tiempo de CPU de 5679 segundos.

El valor de la función objetivo, es decir el makespan del problema es igual a 37 horas.

En la Tabla 2 se muestran los lotes seleccionados en la solución óptima para satisfacer la demanda de cada producto y la planta en la que dichos lotes se procesan, mientras que en la Figura 2 se muestra la asignación, secuenciamiento y tiempos iniciales y finales de procesamiento de los mismos en las unidades de ambas plantas.

**Tabla 2.** Lotes seleccionados, planta donde se procesa cada lote y tamaño en kg. del mismo.

|    |     | $b_1$ | $b_2$ | $b_6$ | $b_7$ | $b_9$ | $b_{10}$ | $b_{15}$ | $b_{16}$ | $b_{19}$ |
|----|-----|-------|-------|-------|-------|-------|----------|----------|----------|----------|
| p1 | n=1 | 150   |       |       |       |       |          |          |          |          |
|    | n=2 |       | 100   |       |       |       |          |          |          |          |
| p2 | n=1 |       |       | 90    | 90    |       |          |          |          |          |
|    | n=2 |       |       |       |       |       |          |          |          |          |
| p3 | n=1 |       |       |       |       | 130   |          |          |          |          |
|    | n=2 |       |       |       |       |       | 100      |          |          |          |
| p4 | n=1 |       |       |       |       |       |          | 95       | 95       |          |
|    | n=2 |       |       |       |       |       |          |          |          |          |
| p5 | n=1 |       |       |       |       |       |          |          |          | 120      |
|    | n=2 |       |       |       |       |       |          |          |          |          |



**Fig. 2.** Asignación y secuencia de los lotes en las unidades de ambas plantas

## 5 Conclusiones

En este trabajo fue presentado un modelo MILP, para la resolución de los problemas de batching y scheduling en instalaciones batch multiproducto y multisitio, donde el objetivo es minimizar el makespan para procesar un conjunto de productos cuya demanda es un dato conocido.

El modelo considera múltiples plantas batch, las cuales cuentan con la misma cantidad de etapas. Los productos pueden ser procesados en cualquiera de las plantas y todos ellos deben pasar a través de todas las etapas. En cada etapa las plantas disponen de unidades que pueden ser de distinta dimensión, trabajando en paralelo y fuera de fase. Además, se consideran tiempos de changeover dependientes de la secuencia de procesamiento de los productos y distintos en cada unidad. Se adopta la política de transferencia zero-wait (ZW), necesaria en los casos donde no se cuenta con disponibilidad de tanques de almacenamiento intermedio.

Los problemas de batching y scheduling son resueltos simultáneamente, lo que permite tener en consideración la relación que existe entre ambos tipos de decisiones, especialmente cuando las instalaciones cuentan con unidades diferentes trabajando en paralelo. Diferentes lotes de un mismo producto pueden ser procesados en plantas distintas aprovechando de mejor manera los recursos disponibles.

El ejemplo presentado muestra la capacidad del modelo para planear la operación de las plantas, determinando los tamaños de lotes, la asignación de los lotes a las plantas, su secuencia y los tiempos de operación.

Como trabajo futuro se pretende considerar distintas alternativas para la producción de los lotes de modo de ajustar su producción no solo a una operación eficiente de la planta sino a compromisos de satisfacción de requerimientos de los clientes.

## Referencias

1. Méndez, C. A.; Cerdá, J.; Grossmann, I. E.; Harjunkoski, I.; Fahl, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng.* 2006, 30 (6-7), 913-946.
2. Harjunkoski, I.; Maravelias, C. T.; Bongers, P.; Castro, P. M.; Engell, S.; Grossmann, I. E.; Hooker, J.; Méndez, C.; Sand, G.; Wassick, J. Scope for industrial applications of production scheduling models and solution methods. *Comput. Chem. Eng.* 2014, 62, 161-193.
3. Prasad, P.; Maravelias, C. T. Batch selection, assignment and sequencing in multi-stage multi-product processes. *Comput. Chem. Eng.* 2008, 32, 1106-1119.
4. Sundaramoorthy, A.; Maravelias, C. Simultaneous batching and scheduling in multistage multiproduct process. *Ind. Eng. Chem. Res.* 2008, 47, 1546-1555.
5. Sundaramoorthy, A.; Maravelias, C. Modeling of storage in batching and scheduling of multistage process. *Ind. Eng. Chem. Res.* 2008, 47, 6648-6660.
6. Fumero, Y.; Corsano, G.; Montagna, J. M. Simultaneous batching and scheduling of batch plants that operate in a campaign-mode, considering nonidentical parallel units and sequence-dependent changeovers. *Ind. Chem. Eng.* 2014, 53, 17059-17074.
7. Kopanos, G. M.; Puigjaner, L. Multi-site scheduling/batching and production planning for batch process industries. *Comput. Aided Chem. Eng.* 2009, 27, 2109-2114.