Universidad Tecnológica Nacional
DICIEMBRE 2020 / Año 18- Nº 39

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

Metodología Matemática-Algorítmica de Programación de Operaciones Aplicada a un Caso de Estudio de Escala Industrial

Algorithmic-Mathematical Scheduling Methodology Applied to an Industrial Size Test Case

Presentación: 25/01/2020

Aprobación: 02/11/2020

Santiago Zuffiaurre
Universidad Tecnológica Nacional, Facultad Regional Santa Fe - Lavaisse 610, (3000) Santa Fe - Argentina
santiago.zuffiaurre@gmail.com

Pablo A. Marchetti
Instituto de Desarrollo Tecnológico para la Industria Química (INTEC), Universidad Nacional del Litoral (UNL) - Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) - Güemes 3450, (3000) Santa Fe - Argentina
Universidad Tecnológica Nacional, Facultad Regional Santa Fe - Lavaisse 610, (3000) Santa Fe - Argentina
pmarchet@intec.unl.edu.ar

Resumen

En este trabajo se presenta una metodología para la programación de operaciones de procesos “batch” en instalaciones multiproducto multietapa. El modelo matemático empleado es de tipo mixto-entero lineal (MILP) y utiliza una representación de ranuras de tiempo (“time slots”). El modelo se complementa con un algoritmo iterativo, basado en la resolución de una secuencia de subproblemas, que permite identificar y fijar la programación de la etapa cuello de botella en cada paso. La metodología propuesta apunta a obtener soluciones de buena calidad para problemas de escala industrial en tiempos de cómputo razonables. Fue aplicada a un caso de estudio real de la industria farmacéutica, que involucra la programación de 30 productos en una planta de 6 etapas y 17 equipos. Si bien no garantiza la optimalidad de la solución hallada, a diferencia de otros aportes de tipo heurístico provee una cota inferior rigurosa que permite medir la calidad de la solución.

Palabras claves: Programación de operaciones, Planta multiproducto, Optimización.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

Abstract

This paper presents a batch process scheduling methodology for multiproduct multistage facilities. A mixed-integer linear programming (MILP) mathematical model based on a time-slot representation is used. The model is complemented with an iterative algorithm that solves a sequence of subproblems, through which the bottleneck stage is identified and fixed at each step. The proposed methodology seeks to attain good quality solutions for industrial size problems in reasonable computational times. It has been applied to a real case study from the pharmaceutical industry, comprising the scheduling of 30 products in a plant with 17 units and 6 processing stages. Even though the proposed method does not guarantee the optimality of the best solution found, in contrast to other heuristic approaches it provides a rigorous lower bound from which its quality can be determined.

Keywords: Scheduling, Multiproduct facility, Optimization.

Introducción

Entre las actividades de planificación de la producción, la programación de operaciones en piso de planta (o "scheduling") es fundamental para garantizar la competitividad de las organizaciones. De hecho, la rentabilidad de una planta industrial está estrechamente vinculada a la eficiente utilización de los recursos de producción disponibles. Esta actividad se caracteriza por un horizonte temporal de corto plazo, realizándose con una frecuencia semanal o quincenal.

En las últimas décadas se han realizado importantes esfuerzos desde las disciplinas de Ingeniería de Procesos e Investigación de Operaciones para el desarrollo de algoritmos y métodos de optimización para el problema de "scheduling". Dado que permite garantizar la optimalidad de la solución cuando el problema planteado es resuelto satisfactoriamente, la formulación de modelos matemáticos de tipo mixto-entero lineal (MILP) es una de las metodologías más utilizadas. Las formulaciones de este tipo de problemas se pueden clasificar, según su representación de la variable tiempo, en aquellas de tiempo discreto y de tiempo continuo (Méndez et al., 2006).

Las representaciones de tiempo discreto se caracterizan por dividir el horizonte de programación en un número finito de intervalos de duración predeterminada, y restringir los tiempos de inicio o finalización de tareas a los extremos de estos períodos. Generan representaciones matemáticas más simples que otras formulaciones, al costo de perder exactitud si la grilla de tiempo no tiene la granularidad suficiente, y disminuir su desempeño computacional cuando el número de intervalos de tiempo aumenta (Kondili et al., 1993; Rodrigues et al., 2000).

Por otra parte, las representaciones de tiempo continuo asocian el tiempo a variables continuas, lo que permite modelar los instantes exactos en los que suceden los eventos. Sin embargo, estas formulaciones muchas veces requieren el uso de restricciones más complejas e incluso restricciones de tipo "big-M". Las metodologías para procesos secuenciales utilizan representaciones continuas de la variable tiempo tales como: ranuras de tiempo o "time-slots" (Pinto y Grossmann, 1995; Lim y Karimi, 2003) y variantes del concepto de precedencia, incluyendo precedencia inmediata (Cerdá et al., 1997; Gupta y Karimi, 2003) y general (Méndez et al., 2001; Marchetti y Cerdá, 2009; Marchetti et al., 2012).

Harjunkoski et al. (2014) presentan una extensa revisión donde analizan las fortalezas y

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

debilidades de los diferentes modelos de programación de operaciones colocando el foco en la implementación de aplicaciones industriales concretas. Dichos autores sostienen que las herramientas de optimización actuales pueden aplicarse satisfactoriamente para aumentar la eficiencia de la producción en piso de planta.

En este trabajo se desarrolla una metodología matemática-algorítmica para la programación de operaciones de una planta "batch" multiproducto multietapa, utilizando una representación basada en ranuras de tiempo ("time-slots"). El modelo se complementa con un algoritmo iterativo, basado en la resolución de una secuencia de subproblemas, que permite identificar y fijar la programación de la etapa cuello de botella en cada paso. Una característica fundamental del método propuesto consiste en su capacidad de resolver problemas reales de escala industrial, y en la consideración de una cota rigurosa que permite evaluar la calidad de la solución obtenida. La metodología fue aplicada a un caso de estudio real de la industria farmacéutica, introducido y estudiado previamente por Castro et al. (2009) y Kopanos et al. (2010).

La estructura del trabajo es la siguiente: en la sección Metodología se formaliza el problema abordado, se describe la formulación matemática basada en ranuras de tiempo utilizada, y se presenta en forma resumida la metodología iterativa propuesta en sus dos niveles. Luego, en la sección Resultados se introduce el caso de estudio tomado como base para este trabajo, y se presenta la solución obtenida junto con el requerimiento computacional requerido para su obtención. En la sección Discusión se analizan los resultados y presentan comparaciones y, para finalizar, se presentan las Conclusiones obtenidas.

Metodología

Descripción del problema

Dados: (i) una planta multiproducto multietapa de producción por lotes (planta "batch") con numerosos equipos $j \in J_l$ operando en paralelo en cada etapa $l \in L$ del proceso, (ii) los lotes o "batches" $i \in I$ a ser procesados y la secuencia de etapas $l \in L_i \subseteq L$ requeridas para cada lote, (iii) el subconjunto de equipos $J_{il} \subseteq J_l$ disponibles para cada tarea (i, l), (iv) los tiempos de procesamiento pt_{ij} del "batch" i en el equipo j, (v) los tiempos de "setup" (o alistamiento) $\tau_{i'ij}$ dependientes de la secuencia (es decir, el tiempo de limpieza o preparación del equipo j cuando el lote i' se procesa inmediatamente antes del lote i), y (vi) la duración H del horizonte de tiempo disponible para la programación. El objetivo del problema es encontrar un programa de operaciones factible, que contemple la asignación, secuenciación y temporización detallada de las tareas, completando todos los lotes dentro del horizonte temporal especificado y, simultáneamente, minimizando el tiempo de finalización máximo o "makespan."

En este trabajo no se consideran fechas de entrega para los lotes y se asume que existe disponibilidad ilimitada o suficiente de otros recursos de fabricación (por ejemplo: tanques de almacenamiento para productos intermedios y finales, mano de obra, servicios, etc.)

Formulación de ranuras de tiempo para la programación de operaciones multietapa

El foco de estudio del presente trabajo es un caso real y frecuente de la industria, aquel de una planta multiproducto en el que la cantidad de productos es ampliamente mayor a la cantidad de equipos en cada etapa. Se eligió representarlo mediante una formulación de ranuras de tiempo debido a su capacidad para manejar de manera eficiente la secuenciación

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

de tareas consecutivas en un equipo dado.

El modelo de ranuras de tiempo ("time slots") especifica un número fijo de períodos de tiempo para cada equipo, durante los cuales los lotes pueden ser procesados. Como formulación de base se tomó aquella propuesta por Pinto y Grossmann (1995), con modificaciones para representar los tiempos de transición dependientes de la secuencia. Esta representación considera intervalos de tiempo con duración variable, de modo que los instantes de comienzo y finalización de las tareas son modelados con variables continuas. A continuación se describe la formulación.

La decisión de asignar un determinado lote *i* a una ranura *k* asociada a un equipo *j* de una etapa *l*, se representa mediante la variable binaria W_{ijkl}, la cual tomará el valor uno cuando se realiza la asignación; o cero en caso contrario. La ecuación (1) indica que cada lote debe ser asignado a una ranura de alguno de los equipos, para cada una de las etapas l que le correspondan. La ecuación (2) limita la cantidad de lotes que se procesan por ranura a 1. Seguidamente, la ecuación (3) asegura que las ranuras sean utilizadas en orden, evitando que se dejen ranuras sin utilizar entre otras asignadas (esto evita soluciones simétricas).

$$\sum_{j\in J_{il}}\sum_{k\in K_j} W_{ijkl} = 1 \qquad \forall i \in I, l \in L_i \tag{1}$$

$$\sum_{i\in I_j} W_{ijkl} \le 1 \qquad \forall l \in L, j \in J_l, k \in K_j \tag{2}$$

$$\sum_{i\in I_j} W_{ijkl} \ge \sum_{i\in I_j} W_{ij(k+1)l} \qquad \forall l \in L, j \in J_l, k \in K_j : k < k_j^{last} \tag{3}$$

Las ecuaciones (4) y (5) sincronizan los tiempos de comienzo para el lote y la ranura asociados (ST_{il}^1 y ST_{jk}^2, respectivamente) cuando W_{ijkl}= 1. Dado que se utilizan restricciones de tipo "big-M", esta sincronización entre los tiempos de comienzo de "batches" y ranuras añade dificultad a la resolución del problema. En cambio, la relación entre los tiempos de comienzo mencionados y los tiempos de finalización (CT_{il}^1 y CT_{jk}^2) se obtiene en forma directa considerando el tiempo de procesamiento pt_{ij} asignado al lote o ranura, respectivamente, en las ecuaciones (6) y (7).

$$-M\left(1-W_{ijkl}\right) \le ST_{il}^1 - ST_{jk}^2 \qquad \forall l \in L, i \in I_l, j \in J_{il}, k \in K_j \tag{4}$$

$$M\left(1-W_{ijkl}\right) \ge ST_{il}^1 - ST_{jk}^2 \qquad \forall l \in L, i \in I_l, j \in J_{il}, k \in K_j \tag{5}$$

$$CT_{il}^1 = ST_{il}^1 + \sum_{j\in J_{il}}\sum_{k\in K_j} W_{ijkl}\, pt_{ij} \qquad \forall l \in L, l \in I_l \tag{6}$$

$$CT_{jk}^2 = ST_{jk}^2 + \sum_{i\in I_{jl}} W_{ijkl}\, pt_{ij} \qquad \forall l \in L, j \in J_l, k \in K_j \tag{7}$$

Los tiempos de transición entre "batches" consecutivos procesados en el mismo equipo se introducen en la ecuación (8), donde CO_{jk} es una variable continua positiva que representa el tiempo de alistamiento efectivamente necesario entre los lotes asignados a las ranuras (*k*-1) y *k*. Su valor se determina en la ecuación (9), como la diferencia entre el máximo tiempo de alistamiento posible para el lote *i* en el equipo *j* $\left(\sigma_{ij}^{\max}\right)$, y la magnitud que corresponde restar según el lote precedente *i'*. Nótese que el lado izquierdo de la ecuación (9) sólo tendrá

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

un valor positivo cuando W_{ijkl}=1, en cuyo caso el "changeover" efectivamente requerido será el límite inferior de la variable CO_{jk}.

$$CT^2_{j(k-1)} + CO_{jk} \leq ST^2_{jk} \qquad \forall l \in L, j \in J_l, k \in K_j : k > 1 \tag{8}$$

$$CO_{jk} \geq \sigma^{\max}_{ij} W_{ijkl} - \sum_{i' \in I_j : i' \neq i} \left(\sigma^{\max}_{ij} - \tau_{i'ij} \right) W_{i'j(k-1)l} \quad \forall i \in I, l \in L_i, j \in J_{il}, k \in J_k : k > 1 \tag{9}$$

La ecuación (10) determina que el tiempo de comienzo de un "batch" en una etapa debe ser mayor o igual al tiempo de finalización en la etapa anterior. Finalmente, las ecuaciones (11) y (12) relacionan los tiempos de finalización de los lotes y los equipos como límite inferior del "makespan," representado por la variable continua *MK*, la cual se minimiza en la función objetivo.

$$CT^1_{il} \leq ST^1_{i(l+1)} \qquad \forall i \in I, l \in L_i : l < l^{last} \tag{10}$$

$$CT^1_{il} \leq MK \qquad \forall i \in I, l = l^{last} \tag{11}$$

$$CT^2_{jk} \leq MK \qquad \forall j \in J_l, l = l^{last}, k = k^{last}_j \tag{12}$$

A fin de obtener una región factible más compacta y mejorar la velocidad de convergencia del resolvedor en la búsqueda de la solución óptima, se incorporan al modelo algunas restricciones de ajuste ("tightening constraints"). Se describen a continuación dos de estas restricciones, basadas en parámetros adicionales que representan el tiempo de comienzo más temprano (EST_{il}) y el tiempo remanente mínimo (MRT_{il}) para la tarea (*i, l*). Estos parámetros son calculados inicialmente mediante la ecuación (13). Como se verá más adelante, a medida que avanza en las iteraciones el algoritmo propuesto actualiza los valores de estos parámetros para incluir información de las variables de decisión fijadas. Las ecuaciones (14) y (15) utilizan los parámetros EST_{il} y MRT_{il} para establecer límites inferiores y superiores para el tiempo de comienzo y finalización, respectivamente, de cada ranura (*j, k*). En el caso de la ecuación (15), permite estimar un tiempo máximo luego del cual un lote no puede ser finalizado a fin de asegurar el cumplimiento del "makespan." Otras ecuaciones de ajuste relevantes son aquellas diseñadas para proveer de cotas inferiores a los tiempos de transición, para lo cual se establecieron parámetros cuyo valor representa el mínimo tiempo de alistamiento para cada "batch."

$$EST_{il} = \sum_{l' \in L_i : l' < l} \min_{j \in J_{il'}} \left[pt_{ij} \right], \quad MRT_{il} = \sum_{l' \in L_i : l' > l} \min_{j \in J_{il'}} \left[pt_{ij} \right], \qquad \forall i \in I, l \in L_i \tag{13}$$

$$ST^2_{jk} \geq \sum_{i \in I_j} EST_{il} W_{ijkl} \quad \forall l \in L, j \in J_l, k \in K_j \tag{14}$$

$$CT^2_{jk} \leq MK - \sum_{i \in I_j} MRT_{il} W_{ijkl} \quad \forall l \in L, j \in J_l, k \in K_j \tag{15}$$

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

Algoritmo iterativo propuesto basado en la identificación de cuellos de botella

La aplicación a casos reales de la formulación matemática presentada se ve limitada a problemas de tamaño mediano/pequeño, ya que sólo allí el tiempo de cómputo será razonable. Para abordar la complejidad resultante de la combinatoria en problemas más grandes, se propone una metodología iterativa basada en los cuellos de botella. Comprende una iteración principal (*main iteration*) y otra interna (*inner iteration*).

En términos generales, la iteración principal selecciona en cada paso una etapa cuello de botella o crítica, fijando las decisiones 0-1 para dicha etapa. El proceso de selección consiste en resolver subproblemas (iteración interna) para cada etapa pendiente a fin de identificar aquella en la que se produce un cuello de botella, obteniendo una solución óptima o de buena calidad para dicha etapa. En base a esta solución se fijan las variables de decisión W_{ijkl} correspondientes (en un vector solución $\mathbf{x}$). Luego, el proceso se repite incorporando la información de las etapas previamente fijadas, de manera que en sucesivas iteraciones la cantidad de decisiones pendientes sea menor.

A continuación, se presenta en pseudocódigo el algoritmo principal utilizado para resolver el problema:

MAIN ITERATION
$z^{CUT} = 0$
$z = 0$
$L^{\text{pending}} = \text{SORT}(L)$
While $L^{\text{pending}} \neq$ empty
    Update parameters EST_{il}, MRT_{il}
    $l^{sel} =$ none
    For $l \in L^{\text{pending}}$
        $(z_l, z_l^{LB}, \mathbf{x}_l) =$INNER ITERATION$(l, z^{\text{CUT}})$
        If *solver* not interrupted by z^{CUT}
            If $l^{sel} =$ none or $z_l > z$
                $l^{sel} = l$
                $z = z_l$
            End If
            $z^{CUT} = \max(z^{CUT}, z_l^{LB})$
        End If
    End For
    Fix allocation in $\mathbf{x}$ based on $\mathbf{x}_{l^{sel}}$
    Remove l^{sel} from $L^{pending}$
End While
Return $(z, \mathbf{x})$

El algoritmo comienza fijando valores iniciales para z y z^{CUT}, parámetros que representan el mayor valor mínimo de la función objetivo y un límite inferior para interrumpir la búsqueda, respectivamente. También, se inicializa la lista de etapas aún no fijadas $L^{pending}$ con todas las etapas, preordenadas decrecientemente en base a una estimación del tiempo de procesamiento requerido. Seguidamente, mientras la lista mencionada no esté vacía se

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

repite la iteración principal. Esta consiste en los siguientes pasos: Se actualizan las estimaciones de los parámetros EST_{il} (*earliest start time*) y MRT_{il} (*minimum remaining time*). Se resetea la variable l^{sel}, que especifica la siguiente etapa a fijar. Luego, para cada l pendiente, se resuelve la iteración interna (*inner iteration*) y procesa su resultado. Si se encuentra una solución factible del subproblema donde la función objetivo (FO) es menor que z^{CUT}, el "solver" es interrumpido y se continúa analizando otra etapa. Caso contrario, y cuando no hubiera ninguna etapa seleccionada o el valor de la FO obtenido en la iteración interna z_l fuera mayor que z, se define la etapa considerada como la siguiente a fijar, y se asigna $z=z_l$. Seguidamente, se asigna a z^{CUT} el valor máximo entre el valor de corte actual y el "lower bound" obtenido para la solución de la etapa l considerada. Una vez inspeccionadas todas las etapas pendientes, l^{sel} será la siguiente etapa a fijar, es decir aquella con el mayor mínimo valor de la FO ("makespan"). La solución de esta etapa se incorpora el vector solución $\mathbf{x}$, y la etapa l^{sel} es quitada del conjunto $L^{pending}$. Mediante este proceso, en cada iteración del ciclo While se fijan las decisiones correspondientes a las variables binarias de una de las etapas, construyendo gradualmente una solución completa del problema.

Por su parte, la iteración interna resuelve una secuencia de modelos que representan el subproblema obtenido al considerar una única etapa l, y tiene por objeto encontrar una solución de la mejor calidad posible para dicha etapa. Para esto, se exploran diferentes configuraciones para el número de ranuras en los equipos de l. En esta iteración se resuelven en forma alternada un subproblema aproximado, para determinar un número candidato de ranuras posible, y un subproblema exacto, para encontrar la mejor solución para dicho número de ranuras. En los subproblemas aproximados se considera el último "slot" de cada equipo (k_j^{last}) como una ranura "comodín." Este tipo de ranuras admitirá la asignación de múltiples lotes, no considerándose las decisiones de secuenciación ni tiempos de comienzo y finalización asociados. Por lo tanto, para la ranura k_j^{last} la ecuación (2) se omite y las demás restricciones se modifican consistentemente. Como no considera todas las decisiones de la etapa l, la solución del subproblema aproximado será incompleta. Sin embargo, el número de ranuras seleccionadas en dicha solución se utiliza como configuración de ranuras candidata para el subproblema exacto posterior. En este último, todas las ranuras de la etapa l serán utilizadas, por lo tanto la ecuación (2) se convierte en igualdad y la restricción (3) se omite. En la siguiente iteración interna, la última configuración de ranuras candidata se descarta mediante la introducción de restricciones de corte de tipo entero. A medida que se avanza en las iteraciones se registra la mejor solución encontrada para un subproblema exacto, que constituye la mejor solución factible para el subproblema de la etapa l.

En resumen, la estrategia planteada permite analizar diferentes configuraciones para el número de ranuras de los equipos en l, encontrando una solución exacta $\mathbf{x}_l$ y manteniendo información rigurosa de las cotas durante el proceso. La variable z_l es la función objetivo obtenida por la iteración interna al resolver el subproblema de la etapa l, mientras que z_l^{LB} constituye la mejor solución posible (o *lower bound*) de dicho problema. El vector $\mathbf{x}_l$ representa las variables de decisión, e incluye las variables W_{ijkl} de asignación de lotes a ranuras. El proceso termina cuando la diferencia entre las cotas es suficientemente pequeña $\left(z_l - z_l^{LB} < \varepsilon\right)$ o cuando se alcanza un número máximo de iteraciones. Si algún subproblema exacto obtiene una solución entera menor al parámetro z^{CUT} la iteración interna termina anticipadamente. En este caso, la etapa l no será considerada crítica (es decir, cuello de botella) porque hay otra etapa cuya mejor solución hallada es mayor.

Utilizando la información de las variables de asignación fijadas es posible actualizar las estimaciones de EST_{il} y MRT_{il} en la iteración principal. En las ecuaciones (16) y (17), cuando

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

$l \notin L^{pending}$ los índices j^s, i^p e i^n representan el equipo seleccionado, el "batch" previo y el "batch" siguiente en dicho equipo, respectivamente, para el lote i en la etapa l. Los parámetros con índices (l - 1) o (l + 1) no definidos toman el valor cero, y $\overline{pt}_{il}$ se define en la ecuación (18). Nótese que la ecuación (13) es la estimación inicial de estos parámetros cuando ninguna de las etapas ha sido fijada.

$$EST_{il} = \begin{cases} EST_{i(l-1)} + \overline{pt}_{i(l-1)} & \forall i \in I, l \in L^{pending} \\ \max\left(EST_{i(l-1)} + \overline{pt}_{i(l-1)}, EST_{i^p l} + pt_{i^p j^s} + \tau_{i^p i j^s}\right) & \forall i \in I, l \notin L^{pending} \end{cases} \quad (16)$$

$$MRT_{il} = \begin{cases} MRT_{i(l+1)} + \overline{pt}_{i(l+1)} & \forall i \in I, l \in L^{pending} \\ \max\left(MRT_{i(l+1)} + \overline{pt}_{i(l+1)}, MRT_{i^n l} + pt_{i^n j^s} + \tau_{i i^n j^s}\right) & \forall i \in I, l \notin L^{pending} \end{cases} \quad (17)$$

$$\overline{pt}_{il} = \begin{cases} \min\limits_{j \in J_{il}} \left[pt_{ij} \right] & \forall i \in I, l \in L^{pending} \\ pt_{ij^s} & \forall i \in I, l \notin L^{pending} \end{cases} \quad (18)$$

Resultados

La metodología propuesta se aplicó a un caso de estudio real de la industria farmacéutica, estudiado previamente por Castro et al. (2009) y Kopanos et al. (2010). El ejemplo considerado involucra la producción de 30 "batches". La planta de producción tiene 6 etapas y 17 equipos en total, pertenecientes los dos primeros a la etapa inicial y los restantes en grupos de tres a las etapas sucesivas. Algunos productos no requieren su procesamiento en la etapa 3, resultando un total de 162 tareas de procesamiento a ser programadas. El problema involucra tiempos de "changeover" dependientes de la secuencia de procesamiento, que incluso en algunas etapas son superiores a los tiempos de procesamiento de las tareas. Los datos detallados del problema pueden encontrarse en el material suplementario de Kopanos et al. (2010). A diferencia del problema original, en este estudio no se consideran los "due dates" introducidos por dichos autores.

El algoritmo y formulación matemática se implementaron con la herramienta de modelado GAMS 25.0, utilizando para hallar la solución de los modelos MILP el resolvedor o "solver" GUROBI 7.5. Este resolvedor fue seleccionado por su mejor desempeño y porque dispone del parámetro *bestobjstop*, el cual permite detener la búsqueda cuando se ha encontrado una solución de cierta calidad (z^{CUT}). Se utilizó una PC genérica con procesador Intel Core i7 3.2 GHz y 16 GB de RAM, y se consideró un tiempo límite de 5 minutos para la resolución de los subproblemas de la iteración interna. Al ejecutar el algoritmo principal, fueron seleccionadas sucesivamente las etapas E_4, E_6, E_2, E_3, E_5 y E_1, hasta que el conjunto $L^{pending}$ estuvo vacío. En cuanto al tamaño del modelo, éste presentó variaciones en las distintas iteraciones del algoritmo entre (1556, 716, 8051) y (423, 581, 2360) para el número de variables binarias, variables continuas y restricciones, respectivamente. En general, los subproblemas de las primeras iteraciones son los de mayor tamaño. La Figura 1 muestra el diagrama Gantt de la mejor solución obtenida. El tiempo de computación total de la metodología propuesta para

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

alcanzar esta solución fue de aproximadamente 2 horas y 50 min.

Discusión

La solución obtenida y reportada en la Figura 1 presenta un "makespan" de 26,31 horas, obteniéndose con la metodología propuesta una cota inferior rigurosa de 21,94 h. Estos valores permiten determinar que el problema planteado aún admite una solución un 16,6% mejor, lo cual es el indicador de calidad de la solución ("relative gap"). Cabe mencionar que esta solución es menor a aquella reportada por Kopanos et al. (2010), con un valor de "makespan" de 26,56 h. Sin embargo, ambas soluciones no son directamente comparables porque Kopanos et al. (2010) incorporan fechas de entrega para cada lote, característica que requiere modificaciones del método que serán abordadas a futuro.

El caso de estudio planteado es de una dificultad considerable debido a la explosión combinatoria que generan las múltiples alternativas de asignación de tareas a equipos y de secuenciación de tareas asignadas a un mismo equipo. Para ilustrar esta dificultad, se resolvió una formulación más cercana al problema completo, utilizando también ranuras de tiempo. En dicha formulación se incluyeron todas las variables de decisión y ecuaciones para todas las etapas y equipos pero, para simplificar la elección del número de ranuras por equipo, se seleccionó la configuración de "slots" que aparece en la Figura 1. Por lo tanto, aún considerando todas las etapas, al prefijar el número de ranuras por equipo no se está considerando el espacio de soluciones completo del problema planteado. Este modelo con todas las etapas requirió 4066 variables binarias, 502 variables continuas, y 16479 restricciones, y permitió obtener como mejor solución un "makespan" de 33,57 h luego de 5 h de tiempo de CPU (con gap relativo 30%). En contraste, la metodología matemática-algorítmica propuesta permitió obtener una solución de mejor calidad debido al proceso de construcción incremental de la solución planteado, basado en resolver una secuencia de subproblemas enfocados cada uno en una etapa.

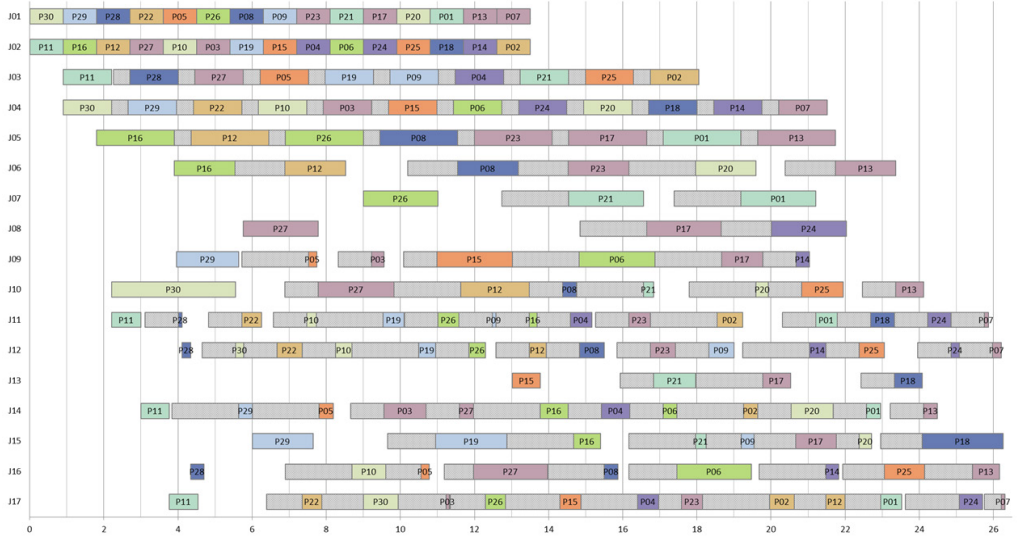

Figura 1. Diagrama de Gantt de la mejor solución obtenida para un caso de estudio real de la industria farmacéutica.

Revista Tecnología y Ciencia
DOI: https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

Conclusiones

En el presente trabajo se desarrolló una metodología matemática-algorítmica para la programación de operaciones de procesos "batch" en instalaciones multiproducto multietapa. El método está basado en una formulación matemática mixta-entera lineal (MILP) que utiliza el concepto de ranura de tiempo o "time slot", combinado con un algoritmo iterativo en el que se resuelve por partes dicho modelo, identificando la etapa cuello de botella en cada paso. La metodología fue aplicada a un caso de estudio real de la industria farmacéutica, obteniendo un programa eficiente de producción para 30 productos, en una planta de 6 etapas y 17 equipos, programando un total de 162 tareas de procesamiento. Si bien la metodología no garantiza la optimalidad de la solución hallada, lo que la diferencia de otros aportes en esta temática es que provee de una cota inferior con la que se puede medir la calidad de la solución.

Como trabajo futuro se prevé revisar el algoritmo para reducir el tiempo de cómputo mediante una selección más eficiente de los subproblemas a analizar. Otro punto a considerar es la incorporación de fechas de entrega para los productos, que no resulta trivial dado que puede introducir la posibilidad de infactibilidades en algún paso de la solución algorítmica. Como ventaja, la reducción de la región factible resultante podría disminuir el tiempo de resolución.

Agradecimientos

Los autores agradecen el financiamiento recibido desde la Universidad Tecnológica Nacional a través del proyecto PID UTN SIUTIFE0004932TC. Además, el primer autor agradece la beca de investigación y desarrollo otorgada por la Universidad Tecnológica Nacional, a través de la Secretaría de Ciencia, Tecnología y Posgrado.

Revista Tecnología y Ciencia
DOI:https://doi.org/10.33414/rtyc.39.151-161.2020 - ISSN 1666-6933

Referencias

Castro, P. M., Harjunkoski, I., y Grossmann, I. E. (2009). Optimal short-term scheduling of large-scale multistage batch plants. *Industrial & Engineering Chemistry Research*, 48, 11002-11016.

Cerdá, J., Henning, G. P., Grossmann, I. E. (1997). A mixed-integer linear programming model for short-term scheduling of single-stage multiproduct batch plants with parallel lines. *Industrial & Engineering Chemistry Research*, 36 (5), 1695-1707.

Gupta, S., Karimi, I. A. (2003). An improved MILP formulation for scheduling multiproduct, multistage batch plants. *Industrial & Engineering Chemistry Research*, 42, 2365-2380.

Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C., Sand, G., y Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. *Computers and Chemical Engineering*, 62, 161-193.

Kondili, E., Pantelides, C. C., Sargent, R. W. H. (1993). A General algorithm for short-term scheduling of batch operations – I. MILP formulation. *Computers and Chemical Engineering*, 17, 211-227.

Kopanos, G. M., Méndez, C. A., y Puigjaner, L. (2010). MIP-based decomposition strategies for large-scale scheduling problems in multiproduct multistage batch plants: A benchmark scheduling problem of the pharmaceutical industry. *European Journal of Operational Research*, 207(2), 644-655.

Lim, M., Karimi, I. A. (2003). Resource-constrained scheduling of parallel production lines using asynchronous slots. *Industrial & Engineering Chemistry Research*, 42, 6832-6842.

Marchetti, P. A., Cerdá, J. (2009). A continuous-time tightened formulation for single-stage batch scheduling with sequence dependent changeovers. *Industrial & Engineering Chemistry Research*, 48, 483-498.

Marchetti, P. A., Méndez, C. A., Cerdá, J. (2012). Simultaneous lot sizing and scheduling of multistage batch processes handling multiple orders per product. *Industrial & Engineering Chemistry Research*, 51 (16), 5762-5780.

Méndez, C. A., Henning, G. P., Cerdá, J. (2001). An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities. *Computers and Chemical Engineering*, 25, 701-711.

Méndez, C. A., Cerdá, J., Grossmann, I. E., Harjunkoski, I., y Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers and Chemical Engineering*, 30, 913-946.

Pinto, J. M., Grossmann, I. E. (1995). A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Industrial & Engineering Chemistry Research*, 34, 3037-3051.

Rodrigues, M. T. M., Latre, L. G., Rodrigues, L. C. A. (2000). Short-term planning and scheduling in multipurpose batch chemical plants: a multi-level approach. *Computers and Chemical Engineering*, 24, 2247-2258.