

# Programación de Operaciones Eficiente de un Caso de Estudio Real de la Industria Farmacéutica mediante una Metodología de Descomposición Matemática-Algorítmica

Josías A. Stürtz<sup>1</sup> and Pablo A. Marchetti<sup>1,2</sup>

<sup>1</sup> Universidad Tecnológica Nacional, Facultad Regional Santa Fe  
Lavaisse 610, 3000 Santa Fe, Argentina

<sup>2</sup> Instituto de Desarrollo Tecnológico para la Industria Química (INTEC)  
UNL-CONICET, Güemes 3450, 3000 Santa Fe, Argentina  
pmarchet@intec.unl.edu.ar

**Resumen** En este trabajo se presenta una metodología matemática-algorítmica para la programación de operaciones de instalaciones discontinuas multiproducto multietapa, desarrollada apuntando a resolver problemas de tamaño industrial. La propuesta se basa en la construcción iterativa de una solución mediante la resolución de una secuencia de subproblemas correspondientes a cada etapa, identificando y fijando las decisiones críticas en cada paso, y manteniendo información rigurosa de las cotas. La metodología se aplica a un caso de estudio real obteniendo soluciones de muy buena calidad en tiempos de cómputo competitivos.

**Keywords:** Plantas discontinuas multiproducto, Programación de operaciones, Optimización

## 1. Introducción

A pesar de innumerables esfuerzos en la formulación y solución de problemas de “scheduling” aún existe una importante brecha entre la teoría y la práctica, particularmente caracterizada por la dificultad intrínseca de resolver problemas de escala industrial. Harjunkoski y colab. [1] sostienen que las herramientas de optimización actuales pueden aplicarse satisfactoriamente para aumentar la eficiencia de la producción en piso de planta, aunque observan que existe mucho potencial de mejora. En este trabajo se presenta una metodología de programación de operaciones para plantas batch multiproducto multietapa considerando equipos en paralelo, almacenamiento ilimitado, y tiempos de transición dependientes de la secuencia, la cual es aplicada a un caso de estudio de escala industrial [2].

## 2. Metodología Matemática-Algorítmica Propuesta

La metodología propuesta consiste en un algoritmo iterativo que resuelve una secuencia de modelos matemáticos de tipo mixto entero lineal (MILP). Cada

uno de estos modelos corresponde a un subproblema del problema general, que considera una de las etapas y un subconjunto de las tareas previas y posteriores, requeridas en las etapas aguas arriba y aguas abajo del proceso. Para modelar los subproblemas se desarrolló una nueva versión del modelo de tiempo continuo basado en “ranuras” de tiempo de Pinto y Grossmann [3], en la cual es posible adicionar en un dado equipo una ranura o slot “comodín”. Es decir, un slot multivaluado al cual puede asignarse un número arbitrario de lotes.

El algoritmo se estructura con una iteración principal y una iteración interna. La iteración principal, que considera todas las etapas del proceso, construye gradualmente una solución del problema completo fijando una a una las soluciones de cada etapa. En cada iteración principal, el conjunto de etapas pendientes (no fijadas) es analizado, resolviendo una secuencia de subproblemas (iteración interna) para cada etapa. Se busca así identificar la etapa crítica, es decir aquella que produce mayor deterioro al valor de la función objetivo (makespan). Una vez que esta etapa es identificada, las decisiones de asignación y secuenciación obtenidas en el subproblema correspondiente son fijadas, y el conjunto de etapas pendientes se reduce. La iteración principal finaliza cuando todas las etapas han sido fijadas y se obtiene una solución completa del problema (schedule).

La iteración interna, por su parte, se utiliza para resolver los subproblemas correspondientes a cada etapa. En esta iteración se resuelven, en forma alternada, un modelo aproximado y un modelo exacto. El subproblema aproximado se utiliza para obtener una configuración de ranuras candidata para la etapa, incorporando ranuras multivaluadas en todos los equipos. Posteriormente, el subproblema exacto fija la configuración de ranuras obtenida previamente para hallar una solución factible para la etapa. En base a la calidad de esta solución y los parámetros del algoritmo se decide terminar la iteración interna o repetir el proceso. En caso de continuar, la configuración de ranuras recientemente evaluada será descartada en los subproblemas aproximados siguientes mediante la incorporación de restricciones de tipo entero. La iteración interna retorna la mejor solución factible encontrada e informa en forma precisa las cotas (solución entera MIP y mejor solución posible) a la iteración principal.

Los parámetros considerados para la iteración interna en una etapa dada son los siguientes: (a) el número de ranuras  $R$  a considerar para los equipos de las etapas anteriores (aguas arriba) y posteriores (aguas abajo) del proceso, (b) un tiempo límite  $T$  para la resolución de cada subproblema, (c) un tiempo límite para la iteración interna, y (d) un número máximo de iteraciones admitidas. La terminación de la iteración interna ocurre luego de obtener una solución del subproblema exacto cuya mejor solución posible no puede mejorarse, o bien al excederse los límites fijados en (c) o (d).

### 3. Resultados y Discusión

El método propuesto fue aplicado a un caso de estudio real de la industria farmacéutica (ver [2]) que involucra una instalación “batch” multiproducto con 17 equipos operando en paralelo distribuidos en 6 etapas. Tomando como base el

problema de 30 baches, se evaluaron instancias incluyendo los primeros 10, 15, 20, 25 y 30 lotes. Se consideraron distintas alternativas para los parámetros (a) con valores  $R=2$ ,  $R=3$  y (b) con valores  $T=300$ , 600, 1200 s. Además, se fijó un tiempo máximo de 1 h y un límite de 10 repeticiones para la iteración interna.

La metodología fue implementada en GAMS, utilizándose el “solver” GUROBI 7.5 para la resolución de los subproblemas de la iteración interna. A fin de comparar el desempeño computacional y la calidad de la solución obtenida (valor de la función objetivo y cota de integralidad), se resolvieron además dos formulaciones completas del problema (es decir, considerando todas las etapas) fijando en este caso un tiempo límite de 5 h. Por un lado, se utilizó un modelo con un número de “slots” fijo equivalente a aquel de la mejor solución obtenida con la metodología propuesta (denominado FULL FIXED). Por otro lado, se resolvió un modelo que considera el número máximo de “slots” posible en cada equipo (FULL). Todas las evaluaciones computacionales se realizaron en un equipo PC genérico, con procesador Intel Core i7 3.2 GHz y 16 GB de RAM.

La Figura 1 muestra cómo al aumentar la cantidad de lotes la estrategia propuesta permitió obtener soluciones de mejor calidad y cotas inferiores similares a los problemas completos (FULL y FULL FIXED), a pesar de resolver únicamente subproblemas. Los tamaños de los modelos y tiempos de CPU para el problema de 30 lotes se comparan en la Tabla 1, donde se observa que la metodología propuesta considera problemas más pequeños, en particular con respecto al número de variables binarias. Finalmente, el diagrama de Gantt de la mejor solución obtenida para este problema se presenta en la Figura 2.

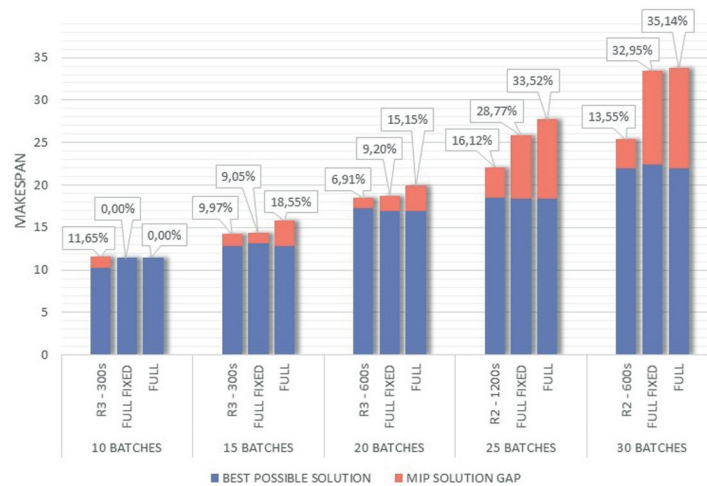


Figura 1: Comparación de las mejores soluciones obtenidas por la metodología propuesta con aquellas de modelos completos con número de ranuras fijo (FULL FIXED) o máximo (FULL).

Tabla 1: Comparación de tamaños de modelo y tiempos de CPU para 30 batches.

	Binary vars	Cont. Vars	Equations	Time (s)
$R=2, T=600s$ min	120	395	2935	5221
max	1556	775	19583	
FULL FIXED	4090	775	19847	18000
FULL	10264	1384	50190	18000

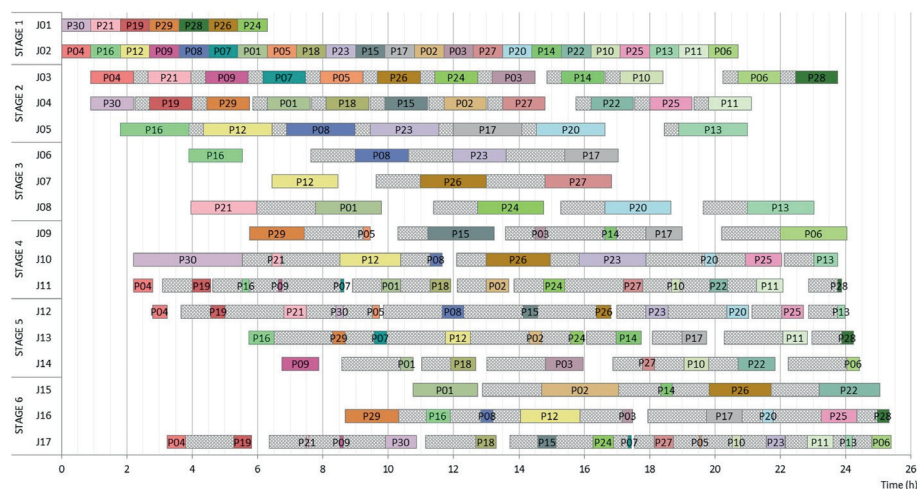


Figura 2: Diagrama de Gantt de la mejor solución obtenida (makespan: 25.38 h) para el problema de 30 lotes, con parámetros  $R=2$  y  $T=600$  s.

## 4. Conclusiones

Se presentó una metodología matemática-algorítmica basada en la resolución de una secuencia de subproblemas que permitió no sólo obtener soluciones de buena calidad en tiempos competitivos para problemas de escala industrial sino también proveer información rigurosa de las cotas de integralidad asociadas.

## Referencias

1. Harjunkoski, I., Maravelias, C.T., Bongers, P., et al.: Scope for industrial applications of production scheduling models and solution methods. *Comput. Chem. Eng.* 62, 161-193 (2014). doi:10.1016/j.compchemeng.2013.12.001
2. Kopanos, G.M., Méndez, C.A., Puigjaner, L.: MIP-based decomposition strategies for large-scale scheduling problems in multiproduct multistage batch plants: A benchmark scheduling problem of the pharmaceutical industry. *Eur. J. Oper. Res.* 207, 644-655 (2010). doi:10.1016/j.ejor.2010.06.002
3. Pinto, J.M., Grossmann, I.E.: A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Ind. Eng. Chem. Res.* 34, 3037-3051 (1995). doi:10.1021/ie00048a015