

***Scheduling* reactivo de procesos *make-and-pack* operando en modo campaña: enfoque CP novedoso**

Lautaro Marcolini¹, Gabriela P. Henning^{1,2}, Franco M. Novara¹

¹Facultad de Ingeniería Química, UNL, Santiago del Estero 2829, 3000 Santa Fe, Argentina

²INTEC (UNL, CONICET), Güemes 3450, 3000 Santa Fe, Argentina

marcolinilautaro@hotmail.com, ghenning@intec.unl.edu.ar,
fnovara@fiq.unl.edu.ar

Resumen. Se presenta un modelo matemático de *Constraint Programming* (CP) para resolver el problema de *scheduling* de corto plazo en plantas industriales de tipo *batch*, multiproducto, multietapa; las cuales operan con procesos *make-and-pack*. Se trata de un enfoque novedoso, capaz de considerar, entre otros aspectos, la operación en modo campañas y tiempos de *changeover* dependientes de la secuencia y/o del equipo productivo involucrado. El mismo permitió resolver 20 instancias, de tamaño creciente en cuanto al número de lotes a agendar, correspondiente a un problema ampliamente abordado en la bibliografía (proceso de producción de helados). Los resultados obtenidos indican un buen desempeño (veloz y eficaz). Asimismo, se trata de una formulación cuyo tamaño no se incrementa exponencialmente con el crecimiento en el número de lotes considerados.

1. Introducción

En la actualidad coexisten innumerables tipos de plantas productivas, con una amplia diversidad de procesos asociados. La dinámica de los ambientes industriales es cada vez más compleja, pues los clientes esperan de manera concurrente mayor calidad, variedad de productos, menores tiempos de entrega y precios, etc. Asimismo, la creciente globalización incrementa la competencia global entre productores.

Ante esta situación, es importante y necesario, para el éxito de cualquier empresa productiva, contar con planes de trabajo que permitan concretar las tareas necesarias a fin de atender a sus clientes, en una manera eficiente y confiable. En este trabajo resulta de particular interés uno de esos planes: el de producción de corto plazo. Por ser aquél que guía el trabajo diario de una planta, su (re)generación no solo demanda rapidez, sino también el considerar una gran cantidad de detalles operativos a fin de garantizar su factibilidad. Detalles que, debido a la diversidad de escenarios, mencionada anteriormente, varían de planta en planta [1].

Por su parte, la informática industrial ha ganado terreno de manera creciente, particularmente, durante las últimas tres décadas, desarrollando diferentes herramientas y metodologías que soportan el proceso de generación de tales planes; e intentado dar respuesta al demandante entorno. No obstante, muchos desafíos permanecen aún vigentes, abarcando desde la limitada capacidad de dichas metodologías para representar

los entornos productivos, lo cual impacta en la factibilidad de los planes resultantes, hasta el pequeño-mediano tamaño de los problemas que son capaces de abordar [2]. Un aspecto no menor, y cada vez más relevante, es la facilidad con la que tales metodologías/herramientas pueden ser reconfiguradas, a fin de acompañar los cambios de procesos, sin perder su eficiencia.

En pos de abordar ambas situaciones mencionadas se desarrolla la propuesta de este trabajo, en el cual se presenta el desarrollo de un modelo de Programación Matemática (*Constraint Programming* – CP) que permite generar tales *schedules* para plantas *batch* multiproducto, multietapa, las cuales trabajan con procesos de tipo *make-and-pack*, operando en modo campaña. Se trata de una formulación de alta expresividad, que resulta una adaptación de una formulación más general desarrollada por [3]. En el Sección 2 se describe el problema tratado, así como las suposiciones realizadas. En el siguiente apartado, se presenta el modelo CP generado. En la Sección 4 se discuten los resultados computacionales vinculados a numerosas instancias de un caso de estudio. Finalmente, se presentan conclusiones y trabajos futuros en la Sección 5.

2. Descripción del problema

Dado un conjunto de *batches*, cada uno de los cuales corresponde a un cierto producto, el problema de *scheduling* consiste en establecer las tareas de procesamiento que conformarán la agenda de producción. Cada tarea se corresponde con la asignación de una unidad productiva al procesamiento de un lote, estableciéndose el tiempo de inicio y finalización. Asimismo, las tareas a desarrollarse en una cierta unidad, deben ser secuenciadas temporalmente entre sí.

En particular, el proceso *make-and-pack* abordado en este trabajo, se compone de tres etapas: elaboración, fermentación/maduración y empaque. Mientras la primera y tercera operan de manera continua, la segunda etapa lo hace de forma intermitente. De esta manera se constituye un proceso semi-continuo.

En la primera etapa se dispone de equipos no idénticos operando en paralelo, con velocidades de producción dependientes del producto elaborado; pudiendo existir asignaciones prohibidas. En este caso, por razones tecnológicas o políticas de operación, ciertas unidades podrían no ser capaces de procesar determinados ítems. Asimismo, entre dos tareas consecutivas, en un mismo equipo, de dos lotes que corresponden a productos diferentes, podría ser necesaria la realización de actividades de *changeover* (limpieza), cuya duración puede depender de la secuencia de producción y/o de la unidad involucrada.

Los equipos de esta etapa no tienen capacidad de almacenamiento, por lo que su producción debe ser trasferida y almacenada, de manera continuada y sin demora, a alguno de los tanques que pertenecen a la siguiente etapa. Estos recipientes, donde se realiza la fermentación/envejecimiento, podrían no estar conectados con todas las unidades de elaboración.

En la segunda etapa, también pueden existir asignaciones prohibidas, por diversas razones. Por ejemplo, podría disponerse de unidades dedicadas a una cierta familia de producto (política de operación predefinida) a fin de evitar contaminaciones cruzadas

entre ítems difíciles de limpiar; o por la limitada capacidad de un tanque frente al tamaño de lote de un determinado ítem; o por las conexiones que cada tanque tenga con las líneas de envasado, que podrían no ser capaces de envasar ciertos productos. Esta última causa es un aspecto relevante, pues no todos los equipos de almacenamiento están conectados con las líneas de envasado aguas abajo, las cuales, a su vez, no son capaces de procesar todos los productos.

Respecto de los tiempos de fermentación, los mismos son considerados determinísticos (no sujetos a variabilidad) y dependientes de cada producto. Asimismo, cada lote puede esperar, una vez cumplimentada dicha tarea, un lapso máximo de tiempo a fin de evitar deterioros. Finalmente, cada tanque quedará listo para ser limpiado y albergar al siguiente *batch* una vez que el mismo se haya vaciado, lo cual ocurre a la velocidad con la que el producto es envasado.

Finalmente, en las líneas de *packing*, que constituyen la tercera etapa, se lleva a cabo el empaque de los productos finales. De manera similar a la etapa *make*, los equipos podrían no ser capaces de fabricar todos los ítems, como se anticipó, por lo que suelen presentarse asignaciones prohibidas. También es posible que se requiera una operación de limpieza entre pares de productos específicos, cuyo tiempo depende de la secuencia (tiempos de *changeover*).

Los equipos de esta etapa suelen ser los más costosos del proceso y, por la cantidad disponible, aquéllos que constituyen la etapa cuello de botella de la planta. En ocasiones, son los únicos que trabajan de manera continua (los de la etapa *make* no siempre lo hacen). Asimismo, la velocidad de trabajo es elevada y se torna relevante asegurar su abastecimiento continuado, reduciendo al mínimo, además, los tiempos improductivos resultantes de los *changeovers*. Para este fin, y en esta etapa, se establece una operación en modo campaña, la cual puede o no estar limitada tanto en el número máximo como mínimo de lotes que la constituyen; pero que generalmente implica realizar corridas largas de envasado.

Finalmente, una vez concluida todas las tareas del plan de trabajo, las líneas de *packing* deben ser sometidas a tareas de limpieza.

3. Enfoque CP propuesto

Se desarrolló un modelo CP que captura el problema de *scheduling* en procesos *make-and-pack*, cuya etapa de envasado opera en modo campaña. El mismo fue construido a partir de aquél presentado en [3] para ambientes *batch*, multietapa, multiproducto. Dicha formulación, también fue utilizada en [4] para considerar escenarios en los cuales los tiempos de procesamiento exhiben una variabilidad no despreciable (estocásticos), en [5] para atacar problemas de gran tamaño mediante la utilización de una metodología de descomposición MILP-CP y en [6] para resolver el problema de *rescheduling* que emerge por la falla de un equipo o ante el arribo de nuevas órdenes.

El modelo presentado en esta sección fue implementado en el lenguaje OPL soportado por el paquete IBM ILOG CP *Optimizer* y en el entorno de desarrollo CPLEX *Optimization Studio* 12.5.1 [7], igual que en los trabajos referidos anteriormente. Tal como se expresa en [3], dentro de las ventajas que la programación con restricciones posee para abordar problemas de *scheduling* se pueden mencionar: la i) posibilidad de

incluir variables y restricciones de manera incremental; ii) rapidez para obtener buenas soluciones iniciales; iii) detección inmediata de especificaciones inviables de problemas (problemas infactibles), iv) capacidad para manejar gran cantidad de variables, v) flexibilidad en el modelado, debido a la posibilidad de utilizar expresiones no lineales, vi) la obtención de soluciones óptimas y subóptimas en tiempos de CPU reducidos, entre otras. Asimismo, permite la introducción de tiempos de *changeover* y secuencias prohibidas de procesamiento sin incrementar el tamaño del problema en términos de variables, así tampoco ocasionar una evidente pérdida de desempeño.

3.1. Nomenclatura

Conjuntos

$P/p, p'$: productos demandados en el horizonte de planeación.

$B/b, b'$: lotes a agendar en el horizonte de planeación.

B_p : lotes del producto p .

C_p/c : campañas de procesamiento asociadas al producto p .

$S/s, s'$: etapas del proceso.

$Sa/-$: etapas de almacenamiento, sub conjunto de S .

$Sp/-$: etapas de procesamiento, sub conjunto de S .

$U/u, u'$: unidades.

U_s/us : unidades pertenecientes a la etapa s .

Parámetros

$maxTime$: tiempo máximo de permanencia para cada lote en las unidades de almacenamiento.

$cleaningTime$: tiempo de limpieza de las unidades de procesamiento después de concluir todas las actividades.

$processingTime_{p,u}$: tiempo de procesamiento de un lote del producto p en la unidad u . En las unidades pertenecientes a la etapa de almacenamiento corresponde al tiempo de envejecimiento.

$changeOverTimes_{p,p',u}$: tiempos de *changeover* en la unidad u para pares de productos p y p' que se procesan consecutivamente.

Variables

$task_{b,u}$: variable de intervalo que representa la ejecución del lote b , en la unidad u que no pertenece a la etapa de envasado.

$packTask_{b,u,c}$: variable de intervalo que representa la ejecución del lote b , en la unidad u de la etapa de envasado, asociada a la campaña c .

$stageTask_{b,s}$: variable de intervalo que representa la ejecución del lote b , en la etapa s .

$taskSeq_u$: variable de secuencia que captura un ordenamiento de las variables de intervalo $task_{b,u}$ en la unidad u .

$campaign_{p,us,c}$: variable de intervalo que representa la campaña c del producto p en la unidad u de la etapa de envasado.

$campaignSeq_u$: variable de secuencia que representa un ordenamiento de las variables de intervalo $campaign_{p,us,c}$ en la etapa de envasado.

Mk: Makespan.

3.2. Restricciones

Las restricciones (1) y (2) establecen que todo *batch* debe ser procesado en una única unidad en cada una de las etapas. Mientras (1) lo hace para las etapas que no operan en modo campaña, (2) lo hace para la etapa de envasado que sí opera bajo dicho modo.

$$\begin{aligned} & \text{Alternative}(\text{stageTask}_{b,s}, \text{all}(u \in Us) \text{task}_{b,u}), \\ & \forall b \in B, \forall s \in S, s \neq \text{card}(S) \end{aligned} \quad (1)$$

$$\begin{aligned} & \text{Alternative}(\text{stageTask}_{b,s}, \text{all}(u \in Us, c \in C_p) \text{task}_{b,u,c}), \\ & \forall b \in B, \forall s \in S, s = \text{card}(S) \end{aligned} \quad (2)$$

Por su parte, las restricciones (3) y (4) determinan las relaciones de precedencia para tareas asociadas a un mismo lote en las etapas de procesamiento *make* y de almacenamiento, respectivamente. Ambas tareas deben comenzar de manera conjunta, al iniciarse la elaboración pues, como se mencionó, la primera etapa es de tipo continua y no posee capacidad de almacenamiento.

$$\begin{aligned} & \text{startAtStart}(\text{task}_{b,u}, \text{task}_{b,u'}), \\ & \forall b \in B, \forall u \in U_s, \forall u' \in U_{s'}, s \in Sp, s' \in Sa, s' = s + 1 \end{aligned} \quad (3)$$

$$\begin{aligned} & \text{startAtStart}(\text{stageTask}_{b,s}, \text{stageTask}_{b,s'}), \\ & \forall b \in B, \forall s \in Sp, s' \in Sa, s' = s + 1 \end{aligned} \quad (4)$$

De manera análoga, las restricciones (5) y (6) establecen que los tanques serán de-ocupados una vez finalizada la tarea de envasado, la cual consume el producto almacenado.

$$\begin{aligned} & \text{endAtEnd}(\text{task}_{b,u}, \text{packTask}_{b,u',c}), \\ & \forall b \in B, \forall u \in U_s, \forall u' \in U_{s'}, \forall s \in Sa, s' \in Sp, s' = \text{card}(S) \end{aligned} \quad (5)$$

$$\begin{aligned} & \text{EndAtEnd}(\text{stageTask}_{b,s}, \text{stageTask}_{b,s'}) \\ & \forall b \in B, \forall s \in Sa, s' \in Sp, s' = \text{card}(S) \end{aligned} \quad (6)$$

La restricción (7) impone la duración de cada tarea de procesamiento (*make* y *pack*).

$$\begin{aligned} & \text{sizeOf}(\text{task}_{b,u}) = \text{processingTime}_{p,u} \cdot \text{presenceOf}(\text{task}_{b,u}), \\ & \forall p \in P, \forall b \in B_p, \forall u \in U_s, \forall s \in Sp \end{aligned} \quad (7)$$

Por su parte, cada tarea de almacenamiento (fermentación/maduración) tiene una duración mínima de acuerdo a la receta de manufactura, así como un período máximo de permanencia, una vez completado el envejecimiento, que dependen de cada producto. Las restricciones (8) y (9) establecen el rango de duración resultante.

$$\text{sizeOf}(\text{task}_{b,u}) \geq \text{processingTime}_{p,u} \cdot \text{presenceOf}(\text{task}_{b,u}), \quad (8)$$

$$\forall p \in P, \forall b \in B_p, \forall u \in U_s, \forall s \in Sa$$

$$\text{sizeOf}(\text{task}_{b,u}) \leq \text{maxTime} \cdot \text{presenceOf}(\text{task}_{b,u}), \quad (9)$$

$$\forall p \in P, \forall b \in B_p, \forall u \in U_s, \forall s \in Sa$$

La restricción (10) captura las campañas de producción en la etapa de envasado. A diferencia de formulaciones anteriores [3], en este caso solo se opera bajo este modo en esta etapa, y se establece la cantidad máxima de campañas para cada producto durante el horizonte de planificación. Para esto se recurre a una variable de intervalo denominada *campaign* que, mediante (10), se extiende temporalmente abarcando el procesamiento de todos los lotes de un determinado producto en una determinada unidad. El constructor *span* sincroniza el inicio de la variable $\text{campaign}_{p,u,c}$ con la variable de intervalo que representa el procesamiento, en la unidad u , del lote más temprano del producto p ; y hace lo propio con la finalización del más tardío.

$$\text{span}(\text{campaign}_{p,u,s,c}, \text{all}(b \in B_p) \text{packTask}_{b,u,c}), \quad (10)$$

$$\forall p \in P, \forall c \in C_p, \forall u \in U_{s=\text{card}(S)}$$

La restricción (11) impide que dos campañas realizadas en una misma unidad se ejecuten con algún solapamiento. Así se evita que otro producto p' , distinto de p , se envase en u durante el lapso de ejecución de $\text{campaign}_{p,u,c}$.

$$\text{spnoOverlap}(\text{campaignSeq}_u), \quad \forall u \in U_s, \forall u \in U_{s=\text{card}(S)} \quad (11)$$

Por su parte, la restricción (12) vincula las tareas correspondientes a los lotes de un determinado producto en un cierto equipo de envasado, con la campaña de ese producto en dicho equipo. Si esta última existe, al menos una tarea debe ser parte de ella.

$$\text{presenceOf}(\text{packTask}_{b,u,c}) \geq \text{presenceOf}(\text{campaign}_{p,u,c}), \quad (12)$$

$$\forall p \in P, \forall c \in C_p, \forall b \in B_p, \forall u \in U_s, \forall u \in U_{s=\text{card}(S)}$$

Mediante (13) se establece el tiempo de *changeover* entre tareas consecutivas en una determinada unidad u . El constructor *noOverlap*, además, impide que dichas tareas se solapen temporalmente.

$$\text{noOverlap}(\text{taskSeq}_u, \text{changeOverTimes}_{p,p',u}), \quad \forall u \in U, \forall p, p' \in P \quad (13)$$

Las restricciones (14) y (15) ayudan a mejorar el desempeño computacional de la formulación, rompiendo simetrías. Estas colaboran con las decisiones tomadas por el modelo referidas a la secuenciación de las tareas de un mismo producto en la etapa de almacenamiento: establecen que luego del ingreso del lote b , el lote del mismo producto que ingresará en dicha etapa será el $b+1$.

$$\text{startBeforeStart}(\text{task}_{b,u}, \text{task}_{b',u}), \quad (14)$$

$$\forall u \in U_s, \forall s \in Sa, \forall b, b' \in B_p, b' = b + 1, b \neq \text{card}(b)$$

$$\begin{aligned} & \text{startBeforeStart}(\text{stageTask}_{b,s}, \text{stageTask}_{b',s}), \\ & \forall s \in Sa, \forall b, b' \in B_p, b' = b + 1, b \neq \text{card}(b) \end{aligned} \quad (15)$$

La expresión (16) establece el *Makespan*, adoptado como medida de desempeño a minimizar, que representa el tiempo que transcurre desde el comienzo de la primera tarea ejecutada en el horizonte de planeación hasta la culminación de la última tarea realizada. En este caso, se adiciona el tiempo que corresponde a la limpieza final de la línea (*cleaningTime*).

$$Mk = \max(\text{endOf}(\text{stageTask}_{b,s}) + \text{cleaningTime}, \forall b \in B, \forall s \in S) \quad (16)$$

4. Casos de estudio

El modelo CP propuesto se aplicó a un caso de estudio introducido inicialmente en [8]. Se trata de un proceso de producción de helados (Figura 1), compuesto de tres etapas como las ya descritas. El problema consiste en la planificación de la manufactura de ocho tipos de productos diferentes (A-H).

Las materias primas ingresan a la primera etapa, que consta de una única unidad de procesamiento (UP1), en donde se transforman en productos intermedios o mezclas (mix A-H). Es importante destacar que, entre dos mezclas distintas, que se elaboran en dicha unidad, es necesario realizar tareas de limpieza (*changeover*).

La producción de esta etapa es transferida, a una velocidad de 4,5 ton/h, a los tanques de almacenamiento de la siguiente etapa (*storage*). En ésta última se dispone de seis unidades (UA1-UA6), que son las encargadas de estacionar la mezcla. Este estacionamiento debe llevarse a cabo durante un período de tiempo mínimo y no exceder, a su vez, una estadía máxima, propia de cada producto y determinada por su receta de manufactura. Asimismo, para esta etapa se presentan tiempos de *changeover* despreciables que son ignorados.

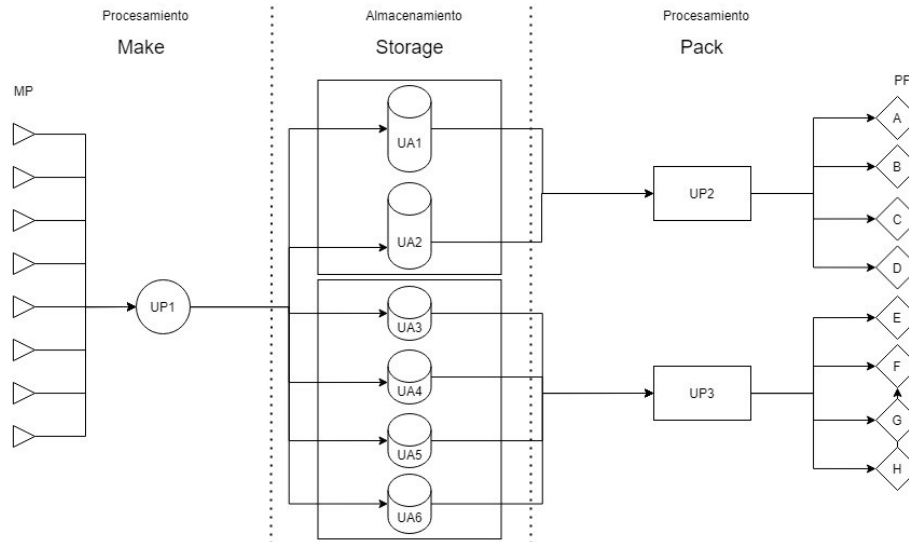


Fig. 1. Representación de proceso bajo estudio.

Una vez envejecidas, las mezclas son envasadas, obteniéndose así los productos finales (A-H). Para esta tarea se dispone de dos unidades dedicadas, pertenecientes a la tercera etapa (*pack*) denominadas UP2 y UP3. En estas líneas, al igual que en la de elaboración, se deben realizar tareas de *changeover* entre el procesamiento de lotes consecutivos correspondientes a productos diferentes. Asimismo, se establece que debe realizarse solo una campaña de cada producto en el horizonte de planificación.

Las mezclas de los productos A-D solo pueden ser enviadas a los tanques de almacenamiento UA1 y UA2 (capacidad de 8 ton cada uno) y desde allí a la unidad de envasado UP2. Por su parte, las de los productos E-H, solo pueden ser enviadas a los tanques de almacenamiento UA3-UA6 (capacidad de 4 ton cada uno), de los cuales se alimenta la unidad de envasado UP3.

Este problema fue abordado por [9], mediante el desarrollo de un modelo de programación matemática lineal mixto-entero (MILP). Posteriormente, [10] modificaron dicho aporte, dando lugar a uno denominado MIP-R, cuyas principales características son las exclusiones de las variables de decisión correspondientes a la asignación de los lotes a las unidades pertenecientes a las etapas *make* y *pack*, de las variables de secuenciación en las unidades de la etapa *pack*, así como las restricciones en las cuales estas variables participan. Esta simplificación es posible en virtud de las características particulares del proceso considerado, a saber: una sola línea continua en la primera etapa, tanques y líneas de *packing* dedicadas para cada producto (flexibilidad nula) y secuencias de productos conocidas “a priori” en la etapa de *packing*.

En el mismo trabajo, se propuso otro enfoque de solución, denominado MIP-BasB, para mejorar la eficiencia computacional del modelo MIP original, con el propósito de resolver problemas de *scheduling* que involucran muchos productos, múltiples lotes y horizontes de planificación más largos. La propuesta se basa en reducir el espacio de búsqueda mediante una estrategia de descomposición que opera en dos etapas. En la primera de ellas se trabaja en forma iterativa, resolviendo el problema para pequeños

conjuntos de *batches*, lo que resulta en tamaños de modelos menores. Nuevamente, esta propuesta carece de generalidad y sólo puede ser aplicada a casos semejantes al aquí descrito.

Resultados

Se aplicó el modelo propuesto al caso de estudio detallado anteriormente para ilustrar y resolver el problema de *scheduling*. Todos los cálculos se realizaron en una computadora con 16 GB RAM y procesador AMD Ryzen 3 3200G.

La metodología propuesta se validó mediante la resolución de veinte instancias referidas al caso de estudio descrito (P.01 a P.20), las cuales fueron introducidas y resueltas anteriormente por [10].

Partiendo desde la instancia P.01 y hasta la P.20, se incrementa la demanda total de cada producto de manera progresiva, trayendo aparejado un aumento en el número de lotes, desde 70 en la primera instancia a 180 *batches* en la última.

La ejecución del modelo permitió confeccionar las diferentes agendas de producción. En la Figura 2 se representa la programación correspondiente al escenario P.01 mediante un diagrama Gantt.

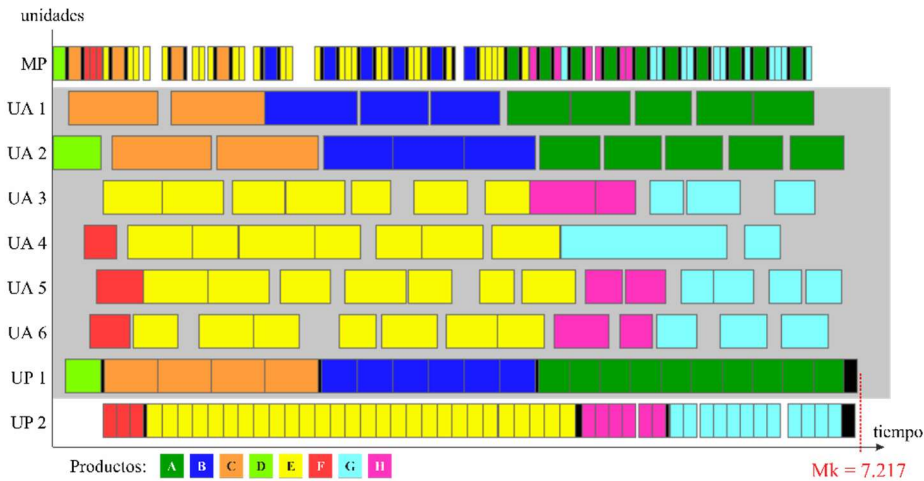


Fig. 2. Agenda resultante para el caso P.01.

Cualitativamente, en la Figura 2 se observa que, debido a estar trabajando la mayor parte del tiempo, la unidad de procesamiento cuello de botella es la UP2, perteneciente a la etapa de *pack*. De manera similar, la UP3 tiene una alta carga de trabajo. En ambos, y como resultado de la operación en modo campaña, los tiempos de *changeover* son mínimos. En contraposición, la UP1 (primera etapa) presenta una gran cantidad de *changeovers*, lo que se corresponde con tiempos de procesamiento cortos, la presencia de una sola unidad y la necesidad de mantener abastecidas a las unidades de las etapas siguientes. Estas observaciones generales se mantienen en todas las instancias estudiadas (P.01 a P.20).

Por razones de espacio no se presenta el análisis completo de la evolución del valor de la función objetivo durante el tiempo de solución para cada instancia. No obstante, es posible comentar algunas conclusiones:

- La primera solución (agenda) se encontró en unos pocos segundos de CPU para el 65% de los casos. Tan solo en uno de ellos el tiempo se extendió más allá de los 20 segundos (P.19).
- En la mayoría de las instancias el valor de la función objetivo, correspondiente a la primera solución hallada, resultó muy bueno. Exceptuando cuatro instancias, la primera solución estuvo, como máximo, solo un 5% por encima de la mejor solución encontrada.
- Se alcanzaron soluciones de buena calidad, en términos del valor de la función objetivo, para la mayoría de los problemas abordados, dentro de los 900 segundos de tiempo de CPU. En las instancias P.04, P.06, P.07, P.11, P.19 y P.20 se obtuvieron mejores soluciones cuando el tiempo límite de resolución se extendió a 3.600 segundos de tiempo computacional.

En la Tabla 1 se presenta la comparación del enfoque propuesto, en términos de los valores de la función objetivo, con los métodos que propusieron [10] para abordar el mismo caso de estudio. Los números resaltados representan los mejores valores de la función objetivo de los problemas, entre los distintos enfoques comparados.

Tabla 1. Comparación del valor de la función objetivo con otros enfoques abordados.

Problema	Enfoque		
	CP Actual	MIP-R	MIP BasB
P.01	120,28	120,33	120,33
P.02	119,48	118,17	118,17
P.03	131,62	131,48	131,48
P.04	142,07	142,1	142,1
P.05	149,65	149,66	149,66
P.06	152,88	152,34	152,34
P.07	162,5	161,47	161,47
P.08	171,35	171,37	171,37
P.09	176,23	175,82	175,82
P.10	187,75	187,75	187,75
P.11	191,18	191,25	191,25
P.12	206,42	206,42	206,42
P.13	202,67	201,76	201,76
P.14	223,55	223,56	223,56
P.15	224,68	224,71	224,71
P.16	222,58	222,06	222,06
P.17	238,48	238,04	238,04
P.18	251,98	251,49	251,49
P.19	260,45	260,52	260,52
P.20	291,72	291,75	291,75

Un análisis de la misma permite notar que con el enfoque CP, nueve de los veinte problemas brindaron un mejor valor de la función objetivo con respecto a las otras propuestas; en dos instancias los resultados fueron iguales; y en nueve, peores. En la mayoría de los escenarios no existió una diferencia significativa, en comparación con los demás métodos, en el valor obtenido de la función objetivo para cada problema.

Un aspecto final pero no menor, que resulta importante mencionar, es el de la dimensionalidad de los modelos generados. Mientras la cantidad de variables de intervalo se mantuvo constante e igual a 18 en las formulaciones resultantes para las 20 instancias (depende del número de equipos y de la cantidad de campañas máximas por producto en las unidades de envasado), la cantidad de variables de intervalo y restricciones creció. Mientras los lotes se incrementaron un 157% entre las instancias P.01 y P.20, las variables de intervalo lo hicieron un 142%, no exhibiéndose un comportamiento exponencial y/o combinatorio en ninguno de los dos elementos. Por su parte, la cantidad de restricciones creció en un 162%, lo que puede considerarse una debilidad de la formulación presentada pues, en general, los modelos CP trabajan mejor cuanto mayor es el número de restricciones presentes [11].

5. Conclusiones

En este trabajo se consideró el problema de *scheduling* en plantas cuyo proceso de manufactura es de tipo *make-and-pack*. Se desarrolló un modelo CP eficiente para hacer frente a dicho problema. El mismo fue aplicado a veinte instancias de escala industrial, correspondiente a una empresa productora de helados. Se resolvieron la mayoría de los escenarios en un tiempo de CPU relativamente bajo, y fue posible encontrar soluciones de muy buena calidad en la mayoría de ellos. Además, las soluciones iniciales halladas fueron, en general, de buena calidad y e instanciadas con rapidez. Todo esto se traduce en un apropiado desempeño computacional.

Como se observó en este trabajo, a medida que se incrementa la cantidad de lotes que deben agendarse, el número de restricciones y variables, de secuencia e intervalo también, también lo hacen, pero no exhiben un compartimiento combinatorio. Adicionalmente, el incremento no se vio reflejado en el requerimiento de mayores esfuerzos computacionales.

Por otra parte, realizando una comparación con las demás propuestas analizadas, se observó que no hubo diferencias significativas en cuanto al valor de la función objetivo, lo que, en conjunto con lo mencionado anteriormente, indica que el enfoque propuesto podría ser una buena alternativa de solución para abordar los problemas de *scheduling* en ambientes *make-and-pack* o de otras industrias con características de procesamiento similares.

Finalmente, esta aplicación del modelo y el entorno de desarrollo presentados evidencian la adaptabilidad de la metodología, la cual fue adaptada y empleada para abordar diferentes problemáticas de *scheduling* por el grupo de investigación.

Como trabajos futuros se presenta la necesidad de contemplar la naturaleza variable y estocástica de los tiempos de maduración. Dichas variaciones podrían ocasionar grandes ineficiencias en las líneas de envasado respecto de lo proyectado cuando la duración de dichos procesos biológicos se considera determinística.

6. Referencias

1. Fazel Zarandi, M.H., Sadat Asl, A.A., Sotudian, S. Castillo, O., A state of the art review of intelligent scheduling, *Artificial Intelligence Review* volume 53, 501–593 (2020)
2. Harjunkoski, I., Maravelias, C., Bongers, P., Castro, P., Engell, S., Grossmann, I., Hooker, J., Méndez, C., Sand, G., & Wassick, J., *Scope for Industrial Applications of Production Scheduling Models and Solution Methods*. *Computers and Chemical Engineering*, 62 (2014), 161-193.
3. Novara, F.M., Novas, J.M., Henning, G.P., *A novel constraint programming model for large-scale scheduling problems in multiproduct multistage batch plants: Limited resources and campaign-based operation*. *Computers and Chemical Engineering*, 93 (2016), 101–117.
4. Novara, F.M., Henning, G.P., *Resilient scheduling under uncertain processing times: a hybrid CP/TOC approach*. *Computer Aided Chemical Engineering*, Volume 44 (2018), 1261-1266.
5. Novara FM, Henning GP, *A Hybrid CP/MILP approach for big size scheduling problems of multiproduct, multistage batch plants*. *Computer Aided Chemical Engineering-37* (2015), 2027–2032.
6. Novara FM, Henning GP, *Planeación reactiva de plantas batch multiproducto, multietapa basada en programación con restricciones*. En: *Anales del 2do Simposio Argentino de Informática Industrial*, 42 Jornadas Argentinas de Informática (2013), 2313–9102.
7. IBM ILOG, 2013. CPLEX - IBM ILOG - 12.5.1
http://www.ibm.com/support/knowledgecenter/SSSA5P_12.5.1/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html?lang=es.
8. Bongers P.M.M., Bakker B.H., *Application of multi-stage scheduling*. *Proceedings 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*. Marquardt W., Pantelides C.C. (Eds.), Elsevier (2006), 1917-1922.
9. Kopanos G.M., Puigjaner L., Georgiadis M.C., *Production scheduling in multiproduct multi-stage semicontinuous food processes*. *Industrial Engineering & Chemistry Research*, 50 (2011), 6316-6324.
10. Kopanos G.M., Puigjaner L., Georgiadis M.C., *Efficient mathematical frameworks for detailed production scheduling in food processing industries*. *Computers and Chemical Engineering*, 42 (2012), 206-216.
12. Darby-Dowman K, Little J, *Properties of Some Combinatorial Optimization Problems and Their Effect on the Performance of Integer Programming and Constraint Logic Programming*. *INFORMS Journal on Computing*, Vol 10, Nro. 3 (1998), 276-286.