

***Scheduling* en plantas *batch* multiproducto, multietapa con incertidumbre en los tiempos de procesamiento: enfoque basado en Programación con Restricciones y Teoría de las Restricciones (CP-TOC)**

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

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

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

fnovara@fiq.unl.edu.ar, ghenning@intec.unl.edu.ar

Abstract. Incertidumbre y variabilidad son características comunes a cualquier ambiente industrial, que afectan las agendas de producción degradándolas y pudiendo tornarlas infactibles o económicamente inviables. La incertidumbre inherente a los procesos productivos, que impacta sobre los tiempos de procesamiento, es una de las fuentes de variabilidad más importante en el *scheduling* de corto plazo. Este trabajo realiza una contribución en el área de *scheduling* proactivo, por medio de un enfoque basado en Programación con Restricciones (CP) y Teoría de las Restricciones (TOC), capaz de considerar la incertidumbre en la etapa de decisión, sin la necesidad de recurrir a la generación de escenarios. La incertidumbre se aborda a partir de los recursos de capacidad restringida (CCR), lo que resulta en un modelo CP de tamaño y complejidad reducidos. El nuevo enfoque se aplica a tres casos de estudio. Se obtienen agendas de producción cuya ejecución es luego simulada, mostrando un mejor desempeño en relación a las agendas generadas mediante un enfoque determinístico y uno estocástico presentado anteriormente por los mismos autores.

Keywords: *Scheduling* bajo incertidumbre - Programación con restricciones - Teoría de las Restricciones - Plantas *batch* multiproducto, multietapa

1 Introducción

Los ambientes industriales se caracterizan por ser entornos dinámicos, en los cuales acontecen diariamente una diversidad ilimitada de eventos inciertos disruptivos: fallas de equipos, demoras en la recepción de materiales, arribo de órdenes urgentes o canceladas, problemas de calidad y rendimientos menores a los esperados, etc. Ante esta situación, es deseable contar con una agenda que sea capaz de “absorber” gran parte de las variaciones, manteniéndose válida sin necesidad de realizarle grandes modificaciones, brindando así estabilidad en la operación. Esto resulta especialmente importante para las funciones de soporte a la producción, cuya operación no se beneficia por el cambio constante de los programas de trabajo, lo que deriva en tiempos muer-

tos, incrementa los costos operativos y genera el denominado “nerviosismo” en el piso de planta.

Al confeccionar una agenda, debe considerarse que la incertidumbre puede provenir de cuatro fuentes [1]: (i) inherente al modelo empleado para confeccionar la agenda; (ii) inherente al proceso productivo; (iii) externa al proceso; (iv) discreta. Mientras la primera fuente incluye, por ejemplo, simplificaciones de modelado, tales como tiempos de *changeover* de duración constante, la segunda considera variaciones de flujos, temperaturas, concentración, etc. que impactan directamente en los tiempos de procesamiento; la tercera incluye variaciones en el mercado, cambios en las órdenes de clientes, etc. y la cuarta abarca aspectos tales como las fallas de equipos. Se han propuesto diversos enfoques para abordar la incertidumbre en los parámetros de planificación, que puede provenir de cualquiera de las cuatro fuentes enunciadas, abordándose en este trabajo aquéllas originadas en (ii).

Cualquiera sea la fuente de incertidumbre, una vez presentada y habiéndose perdido la factibilidad, la agenda debe ser regenerada en un proceso denominado *scheduling* reactivo. Gran parte de los aportes académicos han sido de tipo determinístico y predictivos [2]. Estos proponen la generación de una agenda inicial, empleando parámetros de planificación determinísticos, la que posteriormente es corregida de forma reactiva ante cambios en las condiciones iniciales. Un enfoque alternativo, denominado *scheduling* proactivo, consiste en generar una agenda, considerando la incertidumbre en la etapa de confección, que sea relativamente insensible o inmune a las variaciones en los parámetros de planificación. Si bien este enfoque no evita totalmente el tener que llevar a cabo el *scheduling* reactivo, disminuye su frecuencia.

La generación de una agenda proactiva requiere la utilización de técnicas de planificación estocásticas. Entre los principales enfoques empleados en problemas de *planning* y *scheduling* se encuentran *stochastic programming* [3], *robust optimization* y *robust counterpart optimization* [4]. Otros autores han empleado enfoques alternativos basados en *fuzzy programming* [5], en algoritmos genéticos [6] y programación con restricciones (*Constraint Programming*, CP) [7].

En este trabajo se propone un método de *scheduling* proactivo, que considera la incertidumbre en los tiempos de procesamiento, utilizando un novedoso enfoque basado en CP y Teoría de las Restricciones (*Theory of Constraints*, TOC) que, a diferencia de la mayoría de las restantes contribuciones (a excepción de *robust counterpart optimization* y el enfoque presentado en [7]), no emplea programación basada en escenarios. En la Sección 2 se describe el problema tratado, así como las suposiciones realizadas. En el siguiente apartado, se presenta la metodología y el modelo CP-TOC generado. En la Sección 4 se discuten los resultados computacionales vinculados a tres casos de estudio. Finalmente, se presentan conclusiones y trabajos futuros en la Sección 5.

2 Descripción del problema

2.1 El problema de scheduling.

Dado un conjunto de *batches*, cada uno de los cuales corresponde a un cierto producto, el problema de *scheduling* en una planta multiproducto, multietapa, consiste en

definir la asignación de tareas a cada equipo, estableciendo la secuencia y los tiempos de inicio y fin de éstas en cada unidad. En cada etapa operan en paralelo equipos no idénticos, que podrían no ser capaces de procesar ciertos productos. Respecto a las tareas realizadas en una misma unidad, entre aquéllas que son consecutivas se debe realizar una limpieza y/o alistamiento (*changeover/set-up*) cuya duración puede depender de la unidad y de la secuencia de productos. Debido a *changeovers* prolongados/costosos, podría ocurrir que ciertas secuencias de procesamiento estén prohibidas.

En cuanto a la concatenación de las tareas realizadas sobre un mismo *batch*, ésta depende tanto de la topología de la planta, como de la política de operación adoptada. En virtud que no necesariamente todos los equipos de una cierta etapa están conectados con todas las unidades de la siguiente, la ruta que seguirá el *batch* puede estar restringida. Además, de acuerdo a los requerimientos de proceso, las tareas podrían tener que ejecutarse una a continuación de la otra, sin demoras o con esperas limitadas o ilimitadas. En el caso que sea posible una espera entre etapas, la misma está condicionada por la presencia de tanques de almacenamiento. Este trabajo contempla que aguardar es posible, pero que no existen tanques de almacenamiento; por lo tanto, de producirse una espera, el *batch* debe hacerlo dentro de la unidad que lo procesó (política NIS/UW, *non-intermediate storage - unlimited wait*). Por extensión, si el tiempo de espera es nulo, la propuesta es igualmente válida para NIS-ZW (*zero wait*).

El objetivo es confeccionar una agenda de trabajo que minimice la tardanza total en la terminación de los *batches* respecto de su fecha de entrega (*due-date*).

3 Enfoque estocástico CP-TOC propuesto

A efectos de considerar incertidumbre en los tiempos de procesamiento, el tiempo de finalización de cada *batch* b se va a representar mediante una variable aleatoria (VA), ya que es el resultado de combinar tiempos de procesamiento (cuya duración es aleatoria) con otros tiempos que se supondrán constantes (espera, *release time* y *changeover time*). Como toda VA posee una esperanza matemática o valor esperado y tiene, además, una cierta varianza asociada. Con estos dos elementos, si se conociese la función de densidad de la VA en cuestión, sería posible determinar un valor, llámese $et-StD^P_b$ (*end time subject to deviation*), tal que si se ejecutara reiteradamente el *schedule* confeccionado, el P% de las veces, el tiempo de finalización de b resultaría menor o igual a dicho valor $et-StD^P_b$, siendo P una probabilidad establecida por el *scheduler*.

Se propone confeccionar la agenda de trabajo mediante un modelo CP, empleando tiempos de procesamiento nominales y persiguiendo el objetivo de minimizar la tardanza total. La diferencia respecto de un enfoque determinístico radica en que la tardanza de cada *batch* será computada respecto de una estimación de $et-StD^P_b$ y no respecto del tiempo de finalización nominal. Se trata de un enfoque alternativo al propuesto en [7], que se basa en la identificación de la etapa cuello de botella del proceso y en la aplicación de los conceptos de TOC [8].

Es importante destacar que el *schedule* generado sólo contendrá tareas de duración nominal y no existirán tiempos muertos o *buffers* de tiempos planificados con el fin

de absorber demoras. El ordenamiento de las tareas se hará a partir de una estimación de $et-StD^P$, resultando en una configuración que buscará compensar variaciones, evitar asignaciones de alta variabilidad, etc.

3.1 Estimación de $et-StD_b^P$

La estimación $et-StD_b^P$, que dará lugar a una VA denominada $etStDest_b$, se sustenta en las siguientes suposiciones:

- Existe una etapa limitante, denominada CCS (*Capacity Constrained Stage*), compartida por todos los productos, que puede ser identificada como CCR (*Capacity Constrained Resource*) o BN (*Bottleneck*) [8] y posee una capacidad de procesamiento inferior a la del resto;
- En las unidades pertenecientes a etapas previas y posteriores a la CCS existe capacidad ociosa, que amortiguará las variaciones en los tiempos de procesamiento por encima del valor nominal de cada tarea, sin causar demoras en el inicio de las actividades posteriores que se llevan a cabo en el mismo equipo;
- En las unidades de la CCS no existe capacidad ociosa. Cualquier atraso en el inicio de una tarea repercutirá en el tiempo de finalización del *batch* correspondiente;
- Las duraciones de los tiempos de procesamiento están normalmente distribuidas.

La estimación de $et-StD_b$ puede descomponerse en dos elementos, según (1):

- El tiempo medio esperado de finalización del *batch* b : obtenido a partir del *schedule* nominal ($nomEt_b$). Su valor es el resultado de la suma de la media de variables aleatorias (duraciones nominales de las tareas) y valores constantes (esperas, *release times* y *changeover times*).
- n veces el desvío estándar asociado al tiempo de finalización esperado de b : este elemento, a su vez, se descompone en dos partes: (i) la varianza vinculada al tiempo de finalización del *batch* b en la unidad u de la CCS ($varEnd_{b,u}$); (ii) la varianza asociada a los tiempos de procesamiento del *batch* b en las etapas ubicadas aguas abajo de CCS ($varDS_b$). Ambas varianzas pueden ser combinadas, obteniéndose el desvío estándar buscado.

$$etStDest_b = nomEt_b + n \cdot \sqrt{(varEnd_{b,u} + varDS_b)} \quad (1)$$

Por su parte, $varEnd_{b,u}$ puede descomponerse en dos elementos: (i) la varianza vinculada al inicio del *batch* b en la unidad u de la CCS que le fuera asignada ($varStart_{b,u}$); (ii) la varianza asociada a la tarea de procesamiento realizada sobre b en u ($varCCS_{b,u}$); según la expresión (2).

$$varEnd_{b,u} = varStart_{b,u} + varCCS_{b,u} \quad (2)$$

Finalmente, $varStart_{b,u}$ corresponde a la máxima varianza entre la asociada a la tarea previa realizada en dicha unidad u , sobre un *batch* b' ($varStart_{b',u} + varCCS_{b',u}$), y a la asociada al propio *batch* b en etapas previas ($varUS_b$) (expresión 3)

$$varStart_{b,u} = \max(varStart_{b',u} + varCCS_{b',u}, varUS_b) \quad (3)$$

Debe notarse que $varStart_{b,u}$ involucra la varianza asociada a la tarea previa a la ejecutada sobre el *batch* b en la CCS, la que a su vez puede involucrar la varianza de otras tareas previas, ya sea en la misma unidad o en etapas previas.

3.2 Modelo estocástico CP

El modelo desarrollado captura el problema de *scheduling* permitiendo generar una agenda de trabajo capaz de absorber variaciones en los tiempos de procesamiento. Se lo implementó en el lenguaje OPL soportado por el paquete IBM ILOG CP *Optimizer* y el entorno de desarrollo IBM ILOG CPLEX *Optimization Studio* 12.5.1 [9]. Se ha optado por este ambiente por la *performance* de los constructores de alto nivel especialmente desarrollados para problemas de *scheduling*, que permiten manejar eficientemente las variables de intervalo, empleadas para representar cada tarea de procesamiento y la varianza de su duración, además de la elevada expresividad que puede lograrse con dichos constructores. Por otra parte, el lenguaje permite tratar los *changeovers* dependientes de la secuencia y del equipo sin incrementar en forma significativa el tamaño del modelo, ni los tiempos de resolución.

Nomenclatura.

Conjuntos/Índices.

B/b : *batches* requeridos en el horizonte de planeación.

F_p : productos que no es posible procesar inmediatamente después del producto p .

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

S/s : etapas del proceso productivo.

U/u : unidades del proceso productivo.

UC_u : unidades de la etapa posterior a la de la unidad u no conectadas con ésta última.

Parámetros.

co : tripleta $\langle u, p, p' \rangle$ que contiene los tiempos de *changeover* entre un *batch* de producto p y otro de producto p' en la unidad u .

dd_b : fecha de entrega (*due-date*) del *batch* b .

n : número de desvíos estándar asociados a una probabilidad P fijada por el *scheduler*.

$pt_{b,u}$: tiempo de procesamiento del *batch* b en la unidad u .

$ptVar_{b,u}$: varianza asociada al tiempo de procesamiento $pt_{b,u}$.

Variables.

$etStDest_b$: estimación de $et-StD^P_b$ (variable de tipo flotante).

$nomEt_b$: tiempo de finalización nominal del *batch* b (variable entera).

$stTask_{b,s}$: variable de intervalo que captura el procesamiento del *batch* b en la etapa s .

$task_{b,u}$: variable de intervalo que captura el procesamiento del *batch* b en la unidad u .

$unitBatchSeq_u$: variable de secuencia definida para cada unidad u . Representa un ordenamiento de las variables de intervalo $task_{b,u}$ asociadas a la unidad u . Cada tarea está caracterizada por un tipo que es igual al producto p correspondiente al *batch*.

$var_{b,u}$: variable de intervalo que representa la varianza asociada al procesamiento del *batch* b en la unidad u de la CCS y la varianza del procesamiento aguas arriba.

$varCCS_{b,u}$: captura la varianza del procesamiento de b en u de la CCS (tipo entera)

$varDS_b$: captura la varianza del procesamiento de b aguas abajo de CCS (tipo entera)

$varEnd_{b,u}$: representa la varianza asociada al tiempo de finalización del procesamiento de b en la unidad u de la CCS (tipo entera)

$varSeq_u$: variable de secuencia definida para cada unidad u . Representa un ordenamiento de las variables de intervalo $var_{b,u}$ asociadas a la unidad u . Cada variable está caracterizada por un tipo que es igual al producto p correspondiente al $batch$.

$varUS_b$: representa la varianza del procesamiento de b aguas arriba de CCS (tipo entera)

$totalTardiness$: sumatoria de las tardanzas del $etStDest_b$ respecto del $due-date$ para cada $batch$, variable de tipo flotante que representa la función objetivo.

Restricciones.

El tiempo de inicio más temprano posible para el procesamiento de cada $batch$ en cada unidad (máximo entre el $ready-time$ de la unidad y el $release-time$ del $batch$), así como la duración de la tarea, pueden ser definidos al declarar la variable $task_{b,u}$. Es posible establecer una duración mínima y máxima, así como un rango de tiempo en que la variable puede ubicarse, en caso que esté presente en la solución. Esto constituye una de las ventajas del lenguaje OPL que permite mejorar la expresividad y reducir la cantidad de restricciones empleadas.

La restricción (4) establece que todo $batch$ debe ser procesado en una y sólo una unidad por etapa, mientras que (5) establece la relación de precedencia entre las tareas realizadas sobre un mismo $batch$ en diferentes etapas.

$$alternative(stTask_{b,s}, all(u \in U_s) task_{b,u}), \quad \forall s \in S, \forall b \in B \quad (4)$$

$$\begin{aligned} &endAtStart(stTask_{b,s}, stTask_{b,s'}) \\ &\forall b \in B, \forall s, s' \in S, s \neq Card(S), s' = s + 1 \end{aligned} \quad (5)$$

Mediante (6) se evita que una misma tarea sea asignada a dos unidades de etapas sucesivas no conectadas entre sí, en cuyo caso al menos una de las dos variables que representan estas tareas tendrá duración nula (tiempo de terminación igual a cero).

$$\begin{aligned} &minl(endOf(task_{b,u}), endOf(task_{b,u'})) = 0 \\ &\forall b \in B, \forall u' \in UC_u, \forall u, u' \in U \end{aligned} \quad (6)$$

La restricción (7) permite insertar los tiempos de $changeover$ entre tareas sucesivas, a la vez que impide que dos actividades realizadas en un mismo equipo se ejecuten con algún solapamiento. Por su parte, la expresión (8) evita la presencia de secuencias de procesamiento prohibidas. Ambas emplean variables de secuencia, cuyos detalles se presentaron oportunamente en [10] y se describen en [9].

$$noOverlap(unitBatchSeq_u, co), \quad \forall u \in U \quad (7)$$

$$typeOfNext(unitBatchSeq_u, task_{b,u}) \neq p', \forall (p, p') \in Fp, \forall u \in U, \forall b \in B_p \quad (8)$$

Mediante (9) y (10) se definen las variables $varDS_b$ y $varUS_b$. Debe notarse que en la estimación de la varianza total acumulada del $batch$ b , ya sea aguas arriba o abajo de CCS, sólo se toma en consideración la varianza de las tareas realizadas sobre b , a partir de la simplificación previamente enunciada en la que se suponía la existencia de capacidad ociosa capaz absorber variaciones en los tiempos de procesamiento de otros

batches, evitando que éstas afecten el tiempo de inicio de *b* en cada unidad. Otra simplificación que se realiza, en un entorno NIS-UW, es que el tiempo de espera de *b* dentro de una unidad es despreciable y, por ende, no absorbe parte del tiempo extra empleado por una tarea con una duración superior a la nominal. En un entorno NIS-ZW esta hipótesis no es necesaria. La expresión (11) corresponde a la varianza asociada al procesamiento de un *batch* en la CCS.

$$varUS_b = \sum_{\forall u \in U_s, \forall s \in S, s < CCS} presenceOf(task_{b,u}) \cdot ptVar_{b,u}, \forall b \in B \quad (9)$$

$$varDS_b = \sum_{\forall u \in U_s, \forall s \in S, s > CCS} presenceOf(task_{b,u}) \cdot ptVar_{b,u}, \forall b \in B \quad (10)$$

$$varCCS_{b,u} = presenceOf(task_{b,u}) \cdot ptVar_{b,u} \quad (11)$$

$$\forall b \in B, \forall u \in U_s, \forall s \in S, s = CCS$$

Las expresiones (12) y (13) permiten definir la variable $varEnd_{b,u}$. Debe notarse que $varStart_{b,u}$, introducida en (2) y (3), está implícitamente definida a partir de $varEnd_{b,u}$, y no es capturada explícitamente por ninguna variable del modelo.

En (12) se introduce la variable de intervalo (VI) $var_{b,u}$, que representa la varianza asociada al procesamiento del *batch* *b* en la unidad *u* de la CCS y a una posible varianza en el procesamiento aguas arriba que pueda demorar el inicio de *b* en la CCS. Las variables de intervalo $var_{b,u}$ son agendadas de acuerdo a las restricciones (14)-(16), en el mismo orden que las tareas de procesamiento que representan, sin solapamiento y sin tiempos muertos entre ellas. En otras palabras, para cada unidad perteneciente a la CCS se dispondrá de una secuencia de variables de intervalo $var_{b,u}$ en la cual el inicio del intervalo *n* coincidirá con el fin del (*n-1*).

La duración de cada intervalo $var_{b,u}$ es igual al tamaño de la varianza de la tarea de procesamiento correspondiente, más una magnitud *M*, que depende de la varianza asociada a las tareas ejecutadas sobre el *batch* *b* aguas arriba de la CCS. Como se mencionara, $varStart_{b,u}$ no está definida explícitamente sino que su valor es igual a $varEnd_{b',u}$, siendo *b'* el *batch* antecesor a *b* en la unidad *u*; por lo que si las tareas ejecutadas sobre *b* aguas arriba de la CSS ($varUS_b$) tuvieran una varianza mayor a la asociada a tareas predecesoras en la CCS ($varStart_{b,u}$), se generaría un *gap* entre el valor de $varEnd_{b',u}$ y $varStart_{b,u}$ que impactaría en el valor de $varEnd_{b,u}$. Este *gap* es igual a la diferencia entre $varEnd_{b',u}$ y $varUS_b$, y es el capturado por *M*. En otras palabras, $var_{b,u}$ captura la varianza total asociada al intervalo de tiempo que transcurre entre la finalización del procesamiento del *batch* *b* en la unidad *u*, y la finalización de la tarea antecesora en dicha unidad.

Por ejemplo, la Figura 1 ilustra un caso donde, siendo *u2* la unidad que corresponde a la CCS, la varianza asociada a *b1* en *u1* es mayor a la varianza asociada a tareas predecesoras a *b1* en *u2* (inexistentes, con lo cual el fin de la tarea antecesora se considera tiempo = 0), por lo que la $var_{b1,u2}$ posee una longitud equivalente a la varianza de *b1* en *u1* y en *u2*. Por el contrario, la varianza de *b2* en *u1* es menor a la de $var_{b1,u2}$, por lo en este caso $var_{b2,u2}$ será equivalente solamente a la varianza de *b2* en *u2*. El objetivo último es contemplar aquellos casos donde la variación en la CCS no sea la

única que afecte el tiempo de finalización de una tarea sino que, además, incida la variación ocurrida aguas arriba en las tareas de procesamiento del *batch* bajo análisis. Si la varianza del tiempo de procesamiento de *b2* en *u1* fuera mayor que $var_{b1,u2}$, sería esperable que la variación en el tiempo de finalización de *b2* en *u2* no dependa de $var_{b1,u2}$.

$$sizeOf(var_{b,u}) = presenceOf(var_{b,u}) \cdot \left(\max(0, varUS_b - startOf(var_{b,u})) \right) + varCCS_{b,u} \quad (12)$$

$$\forall b \in B, \forall u \in U_s, s = CCS$$

$$varEnd_b = \max_{\forall u \in U_s, s \in S, s=CCS} (endOf(var_{b,u})) \quad (13)$$

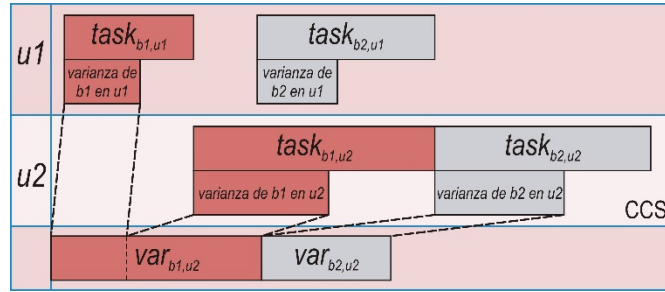


Fig. 1. Esquema conceptual de la duración de las VI que representan varianzas

Mediante (14) se establece, para cada tarea que forma parte de la solución en la CCS, que la varianza asociada también pertenezca a la solución. La expresión (15) permite realizar el secuenciamiento de las variables de intervalo que representan la varianzas de los tiempos de procesamiento en la CCS, para que sigan la misma secuencia que las tareas de procesamiento. Por su parte, (16) impide el solapamiento entre las variables de intervalo que representan varianzas en una misma unidad.

La expresión (17) establece el valor de $nomEt_b$ y la (18), el de $etStDest_b$. Finalmente, (19) introduce la función objetivo.

$$presenceOf(task_{b,u}) = presenceOf(var_{b,u}) \quad (14)$$

$$\forall b \in B, \forall u \in U_s, s = CCS$$

$$typeOfNext(unitBatchSeq_u, task_{b,u}) = typeOfNext(varSeq_u, var_{b,u}), \quad (15)$$

$$\forall (p, p') \forall b \in B, \forall u \in U_s, s = CCS$$

$$noOverlap(varSeq_u), \quad \forall u \in U_s, s = CCS \quad (16)$$

$$nomEt_b = \max_{\forall u \in U} (endOf(task_{b,u})), \quad \forall b \in B \quad (17)$$

$$etStDest_b = nomEt_b + n \cdot \sqrt{(varEnd_b + varDS_b)}, \quad \forall b \in B \quad (18)$$

$$totalTardiness = \sum_{\forall b \in B} \max(0, etStDest_b - dd_b) \quad (19)$$

4 Resultados

La metodología propuesta se validó mediante la resolución de tres casos de estudio de *scheduling* determinístico presentes en la bibliografía, incorporando incertidumbre en los tiempos de procesamiento. La política de operación considerada es de tipo NIS-UW y el objetivo es reducir la tardanza total. Además, en todos los ejemplos se modeló la incertidumbre en los tiempos de procesamiento recurriendo a una distribución triangular. Si bien la metodología se construyó sobre la suposición de normalidad en estos tiempos, en la práctica son más frecuentes los atrasos que los adelantos, lo que resulta en distribuciones asimétricas. Por este motivo se decidió probar el enfoque propuesto empleando distribuciones triangulares de este tipo. Cada tiempo de procesamiento determinístico original se tomó como la moda de la distribución. El valor mínimo se generó restando a la moda un porcentaje de sí misma; dicho porcentaje se acotó, para cada caso, entre 0 e *inf%* y se generó aleatoriamente. Similarmente, el valor máximo se generó adicionando a la moda un porcentaje aleatorio de sí misma, acotado dicho porcentaje entre 0 y *sup%*.

Caso 1. Corresponde al problema abordado por Marcheti y Cerdá en [11] (ejemplo 4). Se trata de un ambiente productivo conformado por 12 equipos, algunos de ellos operando en paralelo, pero no idénticos entre sí, distribuidos en 5 etapas. Se tienen en cuenta tiempos de *changeover* dependientes de la secuencia (se ignoran los tiempos de *set-up*). El objetivo es generar la agenda de trabajo para 12 *batches* (12 productos diferentes) y minimizar la tardanza total de acuerdo a tiempos de entrega preestablecidos. Respecto del caso de estudio original se realizaron las siguientes modificaciones: (i) se supone que el proceso opera bajo una política almacenamiento intermedio/espera de tipo NIS/UW; (ii) se restringen las conexiones entre unidades; (iii) no se consideran limitaciones en los recursos de producción (electricidad, vapor, etc.), (iv) se modifican las *due-dates* de las órdenes aleatoriamente para que exista al menos una solución determinística con una tardanza total igual a cero.

Caso 2. Corresponde al abordado en [12]. Se cuenta con 25 unidades distribuidas en 5 etapas. Deben agendarse 22 *batches* de 22 productos diferentes. Se consideran *ready-times*, *release-times*, *set-up times*, *changeover times* dependientes de la secuencia, secuencias prohibidas y restricciones topológicas. Se emplean las *due-dates* correspondientes a “Set DD2”, según referencia de los autores.

Caso 3. Ejemplo abordado en [13] (problema 15). Deben agendarse 50 *batches* de 50 productos diferentes, en un ambiente productivo que posee 20 unidades distribuidas en 5 etapas. Sólo se consideran *changeover times* dependientes de la secuencia.

La Tabla 1 resume los resultados obtenidos (tardanza total) al simular la agenda generada para cada caso mediante: (i) el enfoque CP-TOC propuesto, (ii) un enfoque determinístico [10] y (iii) otro enfoque estocástico CP propuesto por los mismos autores [7]. En todas las situaciones se realizaron 50.000 simulaciones de ejecución, con las siguientes restricciones: (i) no adelantar ninguna tarea respecto de su tiempo de

inicio planificado, en caso que la misma termine antes de lo previsto, y (ii) utilizar la estrategia de *rescheduling* “*right-shift*” en caso de demoras. Se reportan los valores promedios, expresados en unidades de tiempo genéricas.

A partir de la Tabla 1 y la Fig. 2 puede concluirse que las agendas generadas mediante el nuevo enfoque CP-TOC poseen una mejor *performance* (menor tardanza total) respecto del método determinístico. También exhiben un mejor desempeño respecto del anterior enfoque estocástico, mejorando las soluciones en 11 de los 12 casos analizados y sin la necesidad de proveer una solución inicial en problemas de gran tamaño (C2 y C3), como sí es necesario en caso de aplicar la propuesta [7]. Adicionalmente, la Fig. 3 ilustra el valor de la tardanza total obtenida para cada instancia del caso de estudio 1, mostrando que mediante la propuesta CP-TOC la agenda resultante es relativamente insensible a las variaciones en los tiempos de procesamiento.

Tabla 1. Comparación de resultados (tardanza total): Enfoque CP-TOC vs. otras propuestas

Caso de estudio	inf	sup	Enfoque Determinístico CP [10]	Enfoque estocástico	
				Enfoque CP [7]	Enfoque CP-TOC
C1a	0,05	0,12	1,8	1,2	0,0
C1b	0,075	0,18	6,4	3,8	0,0
C1c	0,1	0,24	14,5	0,1	0,0
C1d	0,125	0,3	24,7	14,4	0,3
C1e	0,15	0,36	39,2	23,4	1,3
C1f	0,175	0,42	58,7	33,9	5,5
C1g	0,2	0,48	83,2	45,1	8,9
C1h	0,225	0,54	105,2	56,8	12,2
C1i	0,25	0,6	148,0	74,1	23,2
C1j	0,275	0,66	191,4	94,3	39,0
C2	0,2	0,5	739,1	693,2	0,0
C3	0,07	0,2	1307,4	112,8	223,2

Tiempo de ejecución: 4.500 segundos CPU – Equipo utilizado: Notebook Asus X555LAB, procesador Intel Core i7-5500U, Memoria Ram 8GB

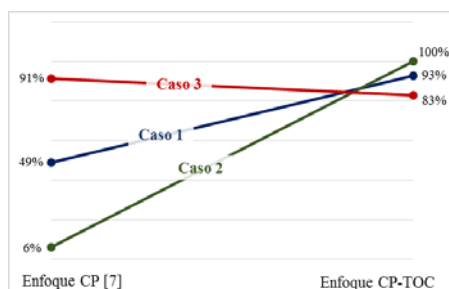


Fig. 2. Porcentaje de mejora en las soluciones obtenidas con cada enfoque estocástico respecto del determinístico

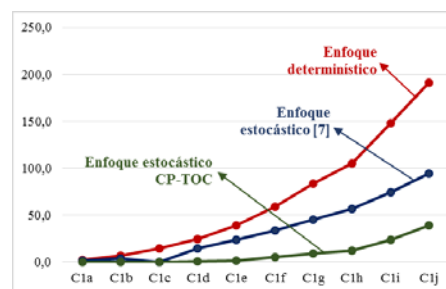


Fig. 3. Tardanza total esperada respecto el aumento de la variabilidad en los tiempos de procesamiento (Caso 1).

La Fig. 4 presenta las agendas de trabajo generadas con la metodología determinística y con la estocástica CP-TOC para el caso C1a. Mientras la simulación de la segunda no muestra órdenes tardíos, la primera presenta una tardanza promedio de 1,2 unidades de tiempo para la orden 1 y 0,6 unidades de tiempo para la orden 10. Los *due-dates* correspondientes son 80 y 100 unidades de tiempo respectivamente. Puede apreciarse como el enfoque CP-TOC genera una asignación distinta para la orden 1 en la CCS, mientras que la orden 10 presenta una asignación casi completamente diferente en todas las etapas, a excepción de la CCS.

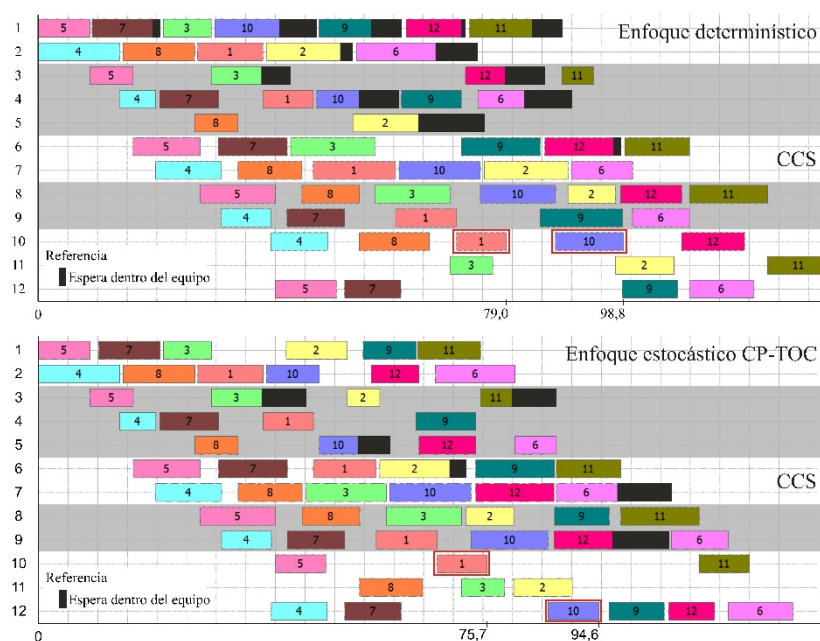


Fig. 4. Comparación de agendas obtenidas para el caso C1a.

5 Conclusiones y comentarios finales

El enfoque CP-TOC propuesto constituye una alternativa a la utilización de escenarios y distribuciones discretas a las que suele recurrirse en otros aportes de modelado estocástico del problema de *scheduling*, las que incrementan significativamente la dimensionalidad del modelo. Más aún, este enfoque resulta en modelos de dimensiones menores respecto del propuesto anteriormente por los mismos autores. El número de variables de intervalo disminuye desde $[2 \cdot \text{Card}(U) \cdot \text{Card}(B) + \text{Card}(S) \cdot \text{Card}(B)]$ a $[\text{Card}(U_{\text{CCS}}) \cdot \text{Card}(B) + \text{Card}(U) \cdot \text{Card}(B) + \text{Card}(S) \cdot \text{Card}(B)]$, y el número de variables secuencia disminuye desde $[2 \cdot \text{Card}(U)]$ a $[\text{Card}(U) + \text{Card}(U_{\text{CCS}})]$. Una limitación que comparte esta propuesta con la anteriormente introducida, es la de recurrir a distribuciones gaussianas, que no suelen capturar el comportamiento de los tiempos de procesamiento en ambientes reales. No obstante, los ejemplos, en los cuales se adoptaron distribuciones asimétricas triangulares, muestran

que el enfoque es lo suficientemente robusto para generar resultados de buena calidad cuando los tiempos de procesamiento siguen otro tipo de distribución

Como trabajo futuro se plantea el entender el enfoque para contemplar políticas de operación/espera de tipo UIS (*unlimited intermediate storage*), NIS-ZW (*non-intermediate storage, zero wait*), NIS-FW (*non-intermediate storage, finite wait*), etc.

6 Referencias

1. Pistikopoulos, E.N.: Uncertainty in process design and operations. *Comput. Chem. Eng.* 19, 553–563 (1995).
2. Harjunoski, 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.* 62, 161–193 (2014).
3. A. Bonfill, A. Espuña, L. Puigjaner: Addressing Robustness in Scheduling Batch Processes with Uncertain Operation Times. *Ind. Eng. Chem. Res.* 44, 1524–1534 (2005).
4. Li, Z., Ierapetritou, M.G.: Robust optimization for process scheduling under uncertainty. *Ind. Eng. Chem. Res.* 47, 4148–4157 (2008).
5. Balasubramanian, J., Grossmann, I.: Scheduling optimization under uncertainty—an alternative approach. *Comput. Chem. Eng.* 27, 469–490 (2003).
6. Bonfill, A., Espuña, A., Puigjaner, L.: Proactive approach to address the uncertainty in short-term scheduling. *Comput. Chem. Eng.* 32, 1689–1706 (2008).
7. Novara, F.M., Henning, G.P.: Scheduling of multiproduct multistage batch plants with uncertain processing times: an innovative constraint programming approach. In: *Proceedings Foundations of Computer-Aided Process Operations*, Tucson, AZ, EE.UU. (2017).
8. Spencer, M.S., Cox, J.F.: Optimum production technology (OPT) and the theory of constraints (TOC): analysis and genealogy. *Int. J. Prod. Res.* 33, 1495–1504 (1995).
9. ILOG, I.: 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.
10. 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. *Comput. Chem. Eng.* 93, 101–117 (2016).
11. Marchetti, P. a., Cerdá, J.: A general resource-constrained scheduling framework for multistage batch facilities with sequence-dependent changeovers. *Comput. Chem. Eng.* 33, 871–886 (2009).
12. Zeballos, L.J., Novas, J.M., Henning, G.P.: A CP formulation for scheduling multiproduct multistage batch plants. *Comput. Chem. Eng.* 35, 2973–2989 (2011).
13. Castro, P.M., Harjunoski, I., Grossmann, I.E.: Optimal short-term scheduling of large-scale multistage batch plants. *Ind. Eng. Chem. Res.* 48, 11002–11016 (2009).