

# MODELOS MIXTO-ENTEROS DE SUBLOTEADO PARA PRODUCCIÓN FLOW-SHOP MINIMIZANDO EL MAKESPAN

Área temática: Gestión de Operaciones y Logística

Rossit, Daniel<sup>(1)\*</sup>, Frutos, Mariano<sup>(1)</sup>, Tohmé, Fernando<sup>(2)</sup>,  
Broz, Diego<sup>(1)</sup>, Rossit, Diego<sup>(1)</sup>

<sup>(1)</sup> *Universidad Nacional del Sur, Departamento de Ingeniería e IIESS-CONICET.  
Av. Alem 1253, Bahía Blanca (8000), Buenos Aires, Argentina.*

[daniel.rossit@uns.edu.ar](mailto:daniel.rossit@uns.edu.ar) - [mfrutos@uns.edu.ar](mailto:mfrutos@uns.edu.ar) - [diego.broz@uns.edu.ar](mailto:diego.broz@uns.edu.ar) - [diego.rossit@uns.edu.ar](mailto:diego.rossit@uns.edu.ar)

<sup>(2)</sup> *Universidad Nacional del Sur, Departamento de Economía e IIESS-CONICET.  
[fohme@criba.edu.ar](mailto:fohme@criba.edu.ar)*

## RESUMEN

Se presenta un análisis comparativo en un entorno productivo del tipo flow-shop con loteo flexible. El objetivo del trabajo es poder determinar bajo qué condiciones es posible incorporar mayor flexibilidad al dimensionamiento de lotes productivo. Distintos autores han demostrado que en la planificación de la producción se puede aumentar significativamente la productividad si se incorpora a la programación de tareas el dimensionamiento del lote de trabajo, es decir se dimensiona el lote considerando el proceso productivo y las operaciones a ejecutarse. Mejorando así la eficiencia de la organización. La bibliografía no es concluyente y abundante sobre qué metodología implementar. Para ello se propone un análisis comparativo de 4 formas distintas de dimensionamiento de lote, contemplando si se permite ociosidad o no en las sucesivas etapas del flow-shop. Y también si se permite o no flexibilidad entre las etapas, que pasada una etapa del proceso productivo se dimensiona de nuevo el lote para la siguiente etapa. Estas técnicas son experimentadas en casos teóricos de mediana complejidad. Las conclusiones a las que se llega en el presente trabajo es que el incorporar flexibilidad entre las etapas (redimensionar el lote antes de pasar a la siguiente etapa) es la técnica que permite aumentar en mayor medida la productividad. En el caso de que se permita ociosidad o no dentro de las etapas no generó grandes diferencias de resultados.

**Palabras Claves:** flow-shop, sub-loteo, modelado, optimización.

## ABSTRACT

A comparative analysis is presented in a flow-shop production with lot streaming. The aim of this paper is to determine under what conditions it is possible to incorporate greater flexibility of lot streaming. Several authors have shown that the production planning can significantly increase productivity if incorporated into the task scheduling lot streaming. Thus improving the efficiency of the organization. The literature is inconclusive on what methodology implement. In this line is that a comparative analysis of four different forms of lot streaming is proposed, contemplating whether or not to allow idleness in the successive stages of the flow-shop. And if is allowed or not flexibility between stages, that is, to resize sub-lots between flow-shop stages. These techniques were experienced in theoretical cases of moderate complexity. The conclusions that can be reached in this paper are that incorporating flexibility between steps (lot resize before moving to the next stage) is the technique that more enhance productivity. In the case that was permitted idle or not in stages no large differences in results were generated.

**Key Words:** flow-shop, lot streaming, modeling, optimization.

## 1. INTRODUCCIÓN

En el sector industrial son bien conocidas las ventajas que reporta tener un proceso productivo industrial eficiente. La eficiencia en la producción permite reducir los costos, elevar el nivel de servicio, aumentar la rentabilidad de la inversión, disminuir las fallas de equipos, disminuir los tiempos perdidos, etc. Dada la gran cantidad de áreas vinculadas, la eficiencia industrial puede abordarse desde diferentes disciplinas. Desde la Investigación Operativa, la eficiencia productiva se centra en las áreas de planificación y programación de las actividades. Uno de los temas más tratado en la bibliografía dentro de esta área es la programación de tareas o secuenciamiento (Scheduling). Esto da lugar a variados enfoques e innovaciones con el objeto de optimizar el proceso de secuenciamiento (minimización de tiempo de procesamientos, de tardanza, de costos, etc.) [1]. El problema de programación de tareas en un ambiente flow-shop se define correctamente en [2] como un problema en el cual  $n$  tareas deben ser procesadas por un conjunto de  $m$  máquinas distintas y las tareas deben tener el mismo flujo de procesamiento o secuencia tecnológica en las máquinas. En la gran mayoría de la bibliografía que trata la programación de tareas no se analiza con mucha importancia el tamaño del lote de productos a secuenciar. Simplemente se incorpora como dato el tamaño del lote. Sin embargo, se ha demostrado que se puede mejorar en gran medida el proceso de secuenciamiento si se optimiza el dimensionamiento de los lotes en función del proceso de manufactura [3]. Esto ha dado lugar a un creciente número de publicaciones en los que se estudia este tema [4] [5] [6] [7]. En ellas se da evidencia que es relevante el modo en que se definen las cantidades a producir y se da una magnitud de la mejora en los resultados del secuenciamiento. Si bien son varios los métodos que se utilizan para determinar el tamaño del lote a producir y cómo mejoran la solución respecto a cuándo no se aplica el sub-loteo, no es tanta la información comparando distintas formas de generar el sub-loteo y las condiciones de manufactura que influyen en este proceso de optimización. Por eso, en este trabajo se hace un análisis comparativo entre distintas variantes de sub-loteo. Esta comparación se hará a través de programación entera lineal (MIP, mixed integer programming) para un entorno de producción flow-shop. El problema a resolver es una versión resumida de un problema clásico (3 productos y 3 etapas), a fin de poder comparar claramente y analizar distintas técnicas de sub-loteo. El resto del trabajo se organiza de la siguiente manera. En la sección 2 se presenta el problema. En la sección 3 se desarrollan los modelos para el problema de sub-loteo. En la sección 4 se establecen las experiencias realizadas y se exponen los resultados. Finalmente, en la sección 5, se dan las conclusiones del trabajo y se proponen futuras líneas de investigación.

## 2. PRESENTACIÓN DEL PROBLEMA

Antes de avanzar con el desarrollo del trabajo, hay una situación que debe aclararse. En la bibliografía estándar del tema se suele denominar proceso de loteo (lot sizing), a la etapa en la que se fijan las cantidades a producir para un horizonte de planificación dado, en donde se debe cumplir con la demanda y los objetivos de la organización. Este proceso, considerando las etapas de producción definidas en [8] corresponde al Plan Maestro de Producción o PMP. En cambio, el proceso de sub-loteo (lot streaming o lot splitting) consiste en tomar ese PMP y generar pequeños sub-lotes, considerando objetivos de menor horizonte temporal y de orden operativo, como lo puede ser minimizar el tiempo total de procesamiento (makespan). Por lo tanto, al desarrollar un sub-loteo se hacen consideraciones del mismo tipo que cuando se busca organizar un secuenciamiento de tareas. Se tiene que analizar el tipo de producción que se quiere optimizar (flow-shop o job-shop), los tiempos de procesamientos, los tiempos de preparación de máquinas, restricciones de capacidad, de espacio disponible para alojar el trabajo en proceso, de costos, etc. Los parámetros que toman una connotación más fuerte al generar un sub-loteo son los tiempos de transferencia de las unidades de una estación a otra. Muchas veces estos tiempos dependen de la cantidad y del tamaño de los lotes, y en este tipo de búsqueda son las variables a modificar. En este trabajo se resuelve un sub-loteo para producir una mezcla de 3 productos que deben ser procesados en 3 etapas. El objetivo del trabajo es comparar distintas formas de sub-lotear y proponer aplicaciones de los modelos propuestos. Las aplicaciones de sub-loteo propuestas podrían bien aplicarse en organizaciones pequeñas o medianas, donde deben enfrentar una alta demanda de productos. Si bien, el sub-loteo puede disminuir la capacidad necesaria para cumplir con esa demanda disminuyendo el tiempo total de procesamiento [6] [7], también permite ir cumpliendo parcialmente un plan, favoreciendo la planificación. Una de las principales ventajas que reporta un sub-loteo para una empresa que debe afrontar una gran producción, es el hecho de

reducir el volumen de trabajo en proceso, conllevando a que la necesidad de materia prima sea escalonada. Esto, para una pequeña empresa, podría significar seguir trabajando con los proveedores que solía hacerlo, manteniendo los beneficios que implica tener una cadena de abastecimiento ya desarrollada. También es menor el capital invertido en trabajo en proceso. Otro aspecto favorable, es que al disminuir la cantidad de productos en proceso, se disminuye la necesidad de espacio físico o buffers dentro del sistema productivo [9] [10]. Hay menos productos esperando para ser procesados por determinadas máquinas. También permite una detención más organizada ante la necesidad de suspender abruptamente la producción por alguna contingencia económica o de otra índole. Lo que se propone exponer, es que mediante un proceso de sub-loteo una pequeña o mediana empresa aumente las probabilidades de producir una alta cantidad de productos. Generando sub-lotes se favorece no solo la producción en sí misma, sino el funcionamiento global de la empresa. Los escenarios planteados a resolver fueron casos de una producción de tipo flow-shop, donde debe cumplirse con cierta producción (PMP). Estos PMP correspondían a 3 productos distintos. Los 3 productos utilizan las mismas máquinas, en la misma secuencia. Los datos e información necesaria para definir el problema (tiempos de procesamiento, tiempos de preparación, etc.) se explican en la sección 4, al igual que los tamaños de PMP.

### **3. MODELOS PARA EL PROBLEMA DE SUB-LOTEO**

En esta sección se presentan 4 modelos distintos de sub-loteo, tomando como base lo publicado en [11]. Los cuatro modelos se dividen en dos grupos. En el primer grupo, se permiten tiempos ociosos en las máquinas o no. Esta división se justifica en entornos donde las máquinas no pueden estar ociosas o, al menos, no para la producción de ese producto. Es decir, la prioridad en la producción la tiene la máquina. No importa que el producto demore un poco más, sino que la máquina esté lo antes posible libre. Al relajar esta restricción, es decir, permitir tiempo ocioso, se incorporan a la búsqueda aquellas soluciones que no tengan un uso continuo de las máquinas. De esta forma puede que se mejore o no la solución anterior, pero no empeorarse. Permitir tiempo ocioso en las máquinas representa una producción en la que no es prioritario el uso de la máquina frente al producto o bien la producción no se hace en ventanas de tiempo. En el segundo grupo, se considera en los modelos el tipo de sub-lote en sí mismo. En muchos casos, la bibliografía establece y fija de ante mano cómo deben ser los sub-lotes, si son todos iguales entre sí o pueden ser distintos entre sí. Es muy frecuente que se consideren sub-lotes de igual tamaño [12] [13] [14] [15]. Los resultados mostrados en los casos citados, evidencian un mejor resultado final en el secuenciamiento cuando se sub-lotea respecto a procesar el lote entero. Además, en el caso de que se deba trasladar las unidades, el hecho de que sean los sub-lotes todos iguales o similares permite estandarizar las técnicas y elementos para el traslado de los mismos de una estación de trabajo a la otra. También es posible relajar este supuesto y permitir que los sub-lotes no sean obligatoriamente todos iguales, es decir, es el caso de sub-lotes de tamaño variables [15] [17]. En el caso del presente trabajo no se exigió igualdad entre los sub-lotes pero sí se puso una cota superior para el tamaño de este respecto al lote entero (una fracción del lote entero). Esta cota, si bien no tiene un fin estrictamente de optimización (restringe la búsqueda), representa una decisión de mantener todos los beneficios mencionados en el apartado 2 de este trabajo, porque el modelo tiene como objetivo el minimizar el makespan, y no contempla esos beneficios. Por eso, con esta cota se les da una representación. Otro punto de interés es, si son modificables los sub-lotes al ir pasando por las distintas operaciones, es decir, si son consistentes o no. Se define como sub-lote consistente cuando no varía su conformación en todo el proceso (tiene siempre la misma cantidad de unidades), y los inconsistentes, sí pueden modificarse [7]. En este trabajo, se consideraron dos casos: sub-lotes variables consistentes y sub-lotes variables inconsistentes. El caso de sub-lotes iguales quedó implícito en la variabilidad de los sub-lotes, no se restringe el caso que sean iguales. Estas variaciones se analizaron porque pueden resultar herramientas muy útiles a la hora de procesar un pedido. El caso de sub-lotes consistentes, en general, es el más aplicable. A nivel operativo, el hecho de que se definan los sub-lotes y se mantengan fijos para todas las etapas del flow-shop resulta más práctico. Tanto que en algunos casos sería muy difícil, cuando no imposible, ir cambiando el número de unidades de un sub-lote, por ejemplo en productos grandes como automóviles, no resultaría del todo práctico modificar en cada etapa el sub-lote. O en el caso de que su implementación no sea la mejor, puede ser motivo de confusión. Sin embargo, hay casos en que es posible trabajar con sub-lotes inconsistentes. Esto agrega flexibilidad al sistema, y permite encontrar nuevas soluciones manteniendo las anteriores, por lo que es posible que el resultado final mejore. A nivel de fábrica es posible aplicarlo en productos de pequeña envergadura, o de fácil manipuleo. También es viable en escenarios en los que intervienen

procesos químicos de tipo batch y donde cada etapa pueda tener distinta capacidad. Entonces agregar flexibilidad al sub-lote representa mejor la realidad. No obstante, se restringe cierta flexibilidad a la reconfiguración de los sub-lotes. En esta presentación no se considera la reconfiguración de los sub-lotes con productos que ya han pasado por una etapa. Primero se procesa todo el sub-lote y después se reconfigura.

El primer modelo a desarrollar es para el caso en el que se permiten tiempos ociosos en las máquinas, y los sub-lotes son de tamaño variable y consistentes. Los modelos matemáticos se encuentran expresados en términos de:

Sub-índices:

f: define el sub-lote en cuestión (el rango de variación se establece para cada prueba).

j: producto (los casos resueltos son para 3 productos).

m: máquina o etapa del flow-shop (los casos resueltos son para 3 etapas).

Variables:

$C_{fjm}$ : tiempo de procesamiento del sub-lote f del producto j en la etapa m.

$s_{fjm}$ : cantidad de unidades del sub-lote f del producto j en la etapa m.

Parámetros:

$U_j$ : tamaño del lote del producto j.

$TR_{jm}$ : tiempo para remover el lote j en la máquina m.

$p_{jm}$ : tiempo de procesamiento unitario del producto j en la máquina m.

$t_{jm}$ : tiempo de configuración de la máquina m para procesar el producto j.

$TF_j$ : tiempo de transferencia fijo del producto j.

$TV_j$ : tiempo de transferencia variable por unidad del producto j.

G: número positivo muy grande.

En la ecuación (1) se define como función objetivo el makespan ( $C_{m\acute{a}x}$ ), y las restricciones se establecen de (2) a (9). La restricción (2) verifica que la suma de unidades dentro de cada sub-lote sea igual al lote planificado a producir (PMP). La restricción (3) establece la consistencia, la cual asegura que cada sub-lote definido sea constante a lo largo de toda la producción. La cota superior para el tamaño del sub-lote se fija en la restricción (4). Dicha cota representa la decisión de asegurar los beneficios del sub-loteo a nivel global de la empresa. Las condiciones de no negatividad se expresan en la restricción (5). La restricción (6) verifica que un sub-lote debe terminarse de procesar en la máquina previa antes de pasar a la siguiente. La restricción (7) permite trabajar con ociosidad. Finalmente, las restricciones (8) y (9) miden que el tiempo final de producción de cada sub-lote incorpore el tiempo suficiente para efectuar la configuración correspondiente, y que el sub-lote anterior haya terminado su procesamiento.

$$Min \rightarrow C_{m\acute{a}x} \quad (1)$$

s.t.:

$$\sum_{f=1}^{f_j} s_{fjm} = U_j, \forall j, \forall m \quad (2)$$

$$s_{fjm} = s_{fj(m+1)}, \forall f, j, m = 1, \dots, (m-1) \quad (3)$$

$$s_{fjm} \leq (\text{cota superior}) * U_j, \forall f, \forall j, \forall m \quad (4)$$

$$s_{fjm} \geq 0, \forall f, \forall j, m \quad (5)$$

$$C_{fj(m+1)} - p_{j(m+1)} * s_{fj(m+1)} \geq C_{fjm} + FT_j + VT_j * s_{fjm}, \forall f, \forall j, m = 1, \dots, (m-1) \quad (6)$$

$$C_{f'j'm} - p_{j'm} * s_{f'j'm} - C_{fjm} + G(1 - y_{ff'j'}) \geq RT_{jm} + t_{j'm}, \forall (f, j) y (f', j'), \forall m : si j = j', entonces f \neq f' \quad (7)$$

$$C_{fjm} - p_{jm} * s_{fjm} \geq t_{jm}, \forall f, \forall j, \forall m \quad (8)$$

$$C_{(f+1)jm} - p_{jm} * s_{(f+1)jm} \geq C_{fjm}, f = 1, \dots, (f_j - 1), \forall m, \forall j \quad (9)$$

El segundo modelo a representar es el que corresponde al caso en que no se permite tiempo ocioso en las máquinas, y los sub-lotes son de tamaño variable y consistentes. Las ecuaciones se corresponden con las del modelo anterior, y sólo se diferencian en la restricción de ociosidad (7), la cual pasa a ser (10).

$$\left( C_{f'j'm} - p_{j'm} * s_{f'j'm} \right) - \left( C_{fjm} - p_{jm} * s_{fjm} \right) + G(1 - y_{jj'}) \geq \left( U_j - \sum_{u=1}^{f-1} s_{ujm} \right) * p_{jm} + RT_{jm} + t_{j'm} + p_{j'm} * \sum_{u=1}^{f'-1} s_{uj'm} \quad (10)$$

$$\forall (f, j) y (f', j') : j \neq j', f = 1, \dots, f_j, \forall j, f', j', m$$

Donde la variable binaria y se modifica. En lugar de estar indexada en f y j, pasa a estarlo sólo en j. Toma valor 1 cuando j precede a j', y es 0 en otro caso.

El tercer modelo se da la situación de permitir tiempos ociosos en las máquinas, pero a diferencia del primer modelo se modifica el tipo de sub-lote. Pasa a ser de tamaño variable e inconsistente. Por este motivo se modifica sólo la restricción (3), y se convierte en (11).

$$\sum_{f=1}^{f-1} s_{fj(m-1)} - \sum_{g=1}^G s_{gjm} + G * x_{ffjm} \geq 0, \forall f, f' \geq 2, \forall j, m = 2, \dots, m, x_{1fjm} = 1 \quad (11)$$

Donde se agrega la variable binaria  $x_{ffjm}$  que vale 1 si el sub-lote f del producto j en la máquina m empieza después de que el sub-lote f' del mismo producto haya sido terminado en la máquina m-1 y es 0 en otro caso.

Por último, el cuarto modelo, correspondería al segundo modelo, donde la restricción (3) pasa a ser (11). Quedando un modelo que no admite tiempos ociosos en las máquinas, con sub-lotes de tamaño variable e inconsistente.

#### 4. EXPERIENCIAS Y RESULTADOS

Para cada modelo se hicieron 1000 experimentaciones, para las cuales se generaron PMP con distribución uniforme entre 100 y 200 unidades de cada producto. A su vez, los tiempos de procesamiento tuvieron valores entre 5 y 20 unidades de tiempo uniformemente distribuidos. Los tiempos de preparación  $t_{jm}$  entre 10 y 35 unidades de tiempo y los de remoción  $RT_{jm}$  entre 2 y 6. Cada uno de los 4.000 escenarios de prueba fue simulado con un conjunto de parámetros independientes. Los PMP fueron divididos en distintas cantidades de sub-lotes. Se hicieron pruebas con 3, 4, 5, 6 y 7 sub-lotes. De esta manera se pudo evaluar el comportamiento de los modelos. Una cuestión importante que está ligada a la cantidad de sub-lotes es la cota superior para la cantidad de unidades que puede contener cada uno de ellos. Esta cota tiene su fundamento en la explicación que se dio anteriormente sobre los beneficios organizacionales que reporta el sub-loteo. El criterio que se utilizó para fijar la cota, fue el de exigir al modelo usar todos los sub-lotes en cantidades representativas. Es decir, si se observan los valores de las cotas (Tabla 1), por ejemplo, para 4 sub-lotes se la fijó en el 30% del lote entero. Esto implica que debe usar sí o sí los 4 sub-lotes con cantidades no menores al 10%, ya que el máximo posible es el 30% para cada sub-lote (3 sub-lotes del 30% logran ubicar el 90 % del PMP). De esta forma se logra que cada sub-lote sea representativo, y se evitan sub-lotes de 2 o 3 unidades de un PMP de 150 unidades lo cual no reportaría ningún beneficio. También se consideró que la cota no excluya el caso de sub-lotes iguales.

Tabla 1 Cotas superiores para el tamaño de los sub-lotes, en proporción al PMP.

	Cota Superior
<b>3 sub-lotes</b>	45%
<b>4 sub-lotes</b>	30%
<b>5 sub-lotes</b>	23%
<b>6 sub-lotes</b>	18%
<b>7 sub-lotes</b>	15%

Esta cantidad de sub-lotes se aplicaron a distintos PMP. Se resolvió para cada caso y se compararon los resultados. Se tomaron PMP con cantidades que van de 100 a 200 unidades, resolviéndose los cuatro modelos para cada caso. Las experiencias se llevaron a cabo en una PC con 4 GB de RAM, y un procesador Intel core i5-3570. Se resolvieron utilizando GAMS/CPLEX, (Versión GAMS 24.0.2 y Solver CPLEX 12.5). Una distinción clara entre los modelos fue la cantidad de variables que requiere cada uno, especialmente las variables discretas porque son las que agregan mayor complejidad. Los modelos en los que no se permitía ociosidad, tanto consistente como inconsistente, utilizan muchas menos variables para resolver los problemas. La cantidad de variables depende de manera proporcional de la cantidad de sub-lotes en los que se quiere dividir el lote, generándose así un problema más grande cuanto mayor sea el número de sub-lotes. La Tabla 2 muestra el detalle de la cantidad de variables discretas según el modelo y según la cantidad de sub-lotes.

Tabla 2 Cantidad de variables discretas para los distintos modelos y problemas.

	Ocio/Consistente	Ocio/Inconsistente	Sin Ocio/Consistente	Sin Ocio/Inconsistente
<b>3 sub-lotes</b>	99	153	33	87
<b>4 sub-lotes</b>	168	264	42	138
<b>5 sub-lotes</b>	255	405	51	201
<b>6 sub-lotes</b>	360	576	60	276
<b>7 sub-lotes</b>	483	777	69	363

La diferencia de cantidad de variables discretas entre ocioso y no ocioso se debe a que la variable binaria utilizada en la restricción (7), está indexada en f y j mientras que la de no ociosidad (10) sólo en j. Por lo tanto, resolver un problema de iguales características sin ociosidad genera un problema más chico. La inconsistencia genera un crecimiento del problema al incorporar variables binarias para la reconfiguración. A su vez, en la Figura 1, se ve claramente cómo influye la cantidad de sub-lotes en el tamaño del problema, es decir, a más sub-lotes mayor cantidad de variables discretas. También muestra que los modelos con ociosidad generan más sub-lotes que los no ociosos, y ésta tendencia se magnifica.

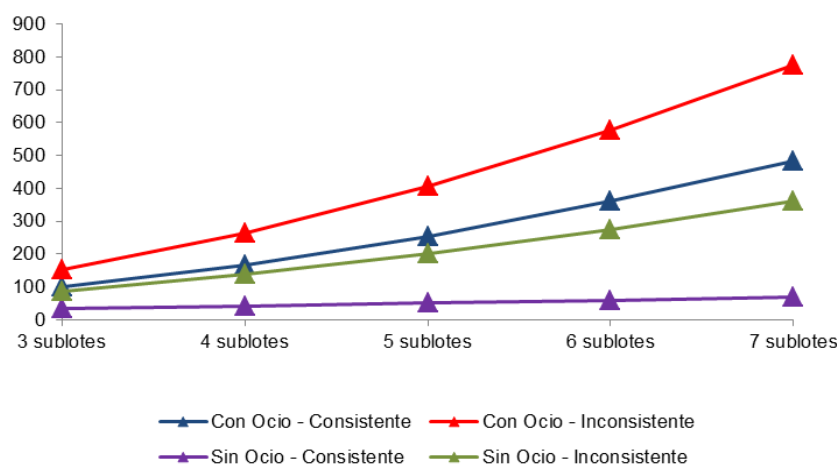


Figura 1 Cantidad de variables discretas vs. cantidad de sub-lotes.

Considerando el objetivo de la optimización, no se notaron diferencias significativas entre los makespan finales entre las condiciones de ociosidad y no ociosidad. Se dio una paridad también en el tamaño de sub-lotes. Esta igualdad puede deberse a la forma de modelar el secuenciamiento, que fue la misma en ambos casos. El modelo tiende a minimizar el tiempo entre la finalización de dos sub-lotes consecutivos. Si la comparación se hace entre los modelos que permitieron inconsistencia en los sub-lotes y los que no la permitieron (Tabla 3), el makespan baja significativamente. Las diferencias más grandes se dan para los números chicos de sub-lotes (3 y 4). En estos casos la diferencia llegó a ser de casi el 60%. Cuando la partición se hace en más sub-lotes, la diferencia estuvo entre 35% y 45%. Considerando el efecto que genera la inconsistencia, parece lógico que su efecto disminuya al agrandar el número de sub-lotes para un PMP máximo similar al que se usó para pocos sub-lotes, ya que de esta forma se reduce la variación posible de tamaño de sub-lotes. Más aun cuando ningún sub-lote podía contener menos de ciertas unidades (debido a la cota superior implementada), por ejemplo en el caso de 7 sub-lotes, si bien se permite reconfiguración, el margen que tiene para reconfigurar cada sub-lote va del 10% al 15% del lote entero, mientras que para 3 sub-lotes, cada sub-lote podía contener desde el 10% al 45% de las unidades, teniendo así mayores posibilidades de reconfigurar. En todos los casos, es notable cómo se logra reducir el tiempo de procesamiento aplicando esta flexibilidad a la formación de sub-lotes.

Tabla 3 *Resultados promediados para casos de Ociosidad y de no ociosidad.*

	Consistente	Inconsistente	Mejora
3 sub-lotes	21.240	9.308	56%
4 sub-lotes	24.390	12.438	49%
5 sub-lotes	27.308	15.335	44%
6 sub-lotes	30.779	18.787	39%
7 sub-lotes	34.129	22.056	35%

#### 4. CONCLUSIONES

Las conclusiones a las que nos permite llegar este trabajo es que cuando se permite generar un nuevo sub-loteo en cada etapa (sub-lotes inconsistentes) se generan los mejores resultados en tiempos de procesamiento o makespan. Este efecto se ve disminuido a medida que el número de sub-lotes crece. Esto se debe a que la variación que puede tenerse en cada sub-lote se ve disminuida por la cantidad de particiones. A su vez, el máximo impuesto a cada sub-lote agudizó esa limitación en la variación en cada sub-lote. Respecto al secuenciamiento ocioso y no ocioso, puede decirse que no hubo diferencias debido al modelado del secuenciamiento. Como futuras líneas de investigación, estamos interesados en aumentar la flexibilidad de la inconsistencia, retirar la necesidad de procesar el sub-lote entero para poder reconfigurar, y permitir la reconfiguración con las unidades ya procesadas. Respecto de la ociosidad, resultaría interesante trasladarla al producto. De esta forma se llegarían a resultados más concluyentes sobre la ventaja o no de permitir ociosidad en estos modelos de sub-loteo.

#### 5. REFERENCIAS

- [1] Hejazi S. Reza, Saghafian S. (2005). "Flow-shop-scheduling problems with makespan criterion: a review", *International Journal of Production Economics*, vol.43, pp. 2895-2929.
- [2] Toro Ocampo E. M., Restrepo Grisales Y. S., Granada Echeverri M. (2006). "Algoritmo genético modificado aplicado al problema de secuenciamiento de tareas en sistemas de producción lineal flow-shop", *Scientia et Technica*, vol. 12, pp. 285-291.
- [3] Chang Jen H., Chiu Huan N. (2005). "A comprehensive review of lot streaming", *International Journal of Production Research*, vol. 43, pp. 1515-1536.
- [4] Potts C. N., Baker K. R. (1989). "Flow shop scheduling with lot streaming", *Operations Research Letters*, vol. 8, pp. 297-303.
- [5] Wagner Bret J., Ragatz Gary L. (1994). "The impact of lot splitting on due date performance", *Journal of Operations management*, vol.12, pp 13-25.

- [6] Kalir Adar A., Sarin Subhash C. (2000). "Evaluation of the potential benefits of lot streaming in flow-shop systems", *International Journal of Production Economics*, vol. 66, pp. 131-142.
  - [7] Low C., Hsu C., Huang K. (2004). "Benefits of lot splitting in job-shop scheduling", *The International Journal of Advanced Manufacturing Technology*, vol. 24, pp. 773-780
  - [8] Vollmann T. E., Berry W. L., Whybark D. C., Jacobs R. (2005). *Planeación y control de la producción, administración de la cadena suministros*, McGraw Hill Interamericana, México, México.
  - [9] Frutos M., Méndez M., Tohmé F., Broz D. (2013). "Comparison of Multiobjective Evolutionary Algorithms for Operations Scheduling under Machine Availability Constraints", *The Scientific World Journal: Bioinspired Computation and Its Applications in Operation Management*. vol. 2013. pp. 1-9.
  - [10] Duclos L. K., Spencer M. S. (1995). "The impact of a constraint buffer in a flow-shop", *International Journal of Production Economics*, vol.42, pp. 175-185.
  - [11] Sarin S. C., Jaiprakash P. (2007). *Flow-shop lot streaming*, Springer, USA.
  - [12] Pan Q. K., Tasgetiren M. F., Suganthan P. N., Chua T. J. (2011). "A discrete artificial bee colony algorithm for the lot-streaming flow-shop scheduling problem", *Information Sciences*, vol. 181, pp. 2455-2468.
  - [13] Tseng Chao-Tang, Liao Ching-Jong (2008). "A discrete particle swarm optimization for lot-streaming flow-shop scheduling problem", *European Journal of Operational Research*, vol.191, pp. 360-373.
  - [14] Marimuthu S., Ponnambalam S. G., Jawaharc N. (2008). "Evolutionary algorithms for scheduling m-machine flow-shop with lot streaming", *Robotics and Computer-Integrated Manufacturing*, vol. 24, pp. 125-139.
  - [15] Marimuthu S., Ponnambalam S. G., Jawaharc N. (2009). "Threshold accepting and Ant-colony optimization algorithms for scheduling m-machine flow-shops with lot streaming", *Journal of Materials Processing Technology*, vol. 209, pp. 1026-1041.
  - [16] Zhang W., Yin C., Liu J., Linn R. J. (2005). "Multi-job lot streaming to minimize the mean completion time in m-1 hybrid flow-shops", *International Journal of Production Economics*, vol. 96, pp. 189-200.
  - [17] Martin C. H. (2009). "A hybrid genetic algorithm/mathematical programming approach to the multi-family flow-shop scheduling problem with lot streaming", *Omega*, vol. 37, pp. 126-137.
- Bettinger, P.; Boston, K.; Siry, J.; Grebner, D. (2009). *Forest management and planning*. 1ra. Edición. Academic Press. USA.

### **Agradecimientos**

Los autores de este trabajo desean agradecer al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) por el apoyo económico brindado. Además, se agradece la financiación de los proyectos PGI 24/J056 (Secretaría General de Ciencia y Tecnología (SGCyT), Universidad Nacional del Sur (UNS)) y PICT 2011-0396 (Fondo para la Investigación Científica y Tecnológica (FONCyT), Agencia Nacional de Promoción Científica y Tecnológica (AGENCIA)).