

## MODELO DE SUBLOTEADO CONSIDERANDO EL EFECTO APRENDIZAJE EN CONFIGURACIONES PRODUCTIVAS FLOW-SHOP

Rossit Daniel<sup>1</sup>, Frutos Mariano<sup>1</sup>, Tohmé Fernando<sup>2</sup>, Broz Diego<sup>1,3</sup>

<sup>1</sup> Departamento de Ingeniería, Universidad Nacional del Sur.  
Av. Alem 1253, Bahía Blanca, Argentina.  
IIESS-CONICET  
rossit.daniel@uns.edu.ar  
mfrutos@uns.edu.ar  
diego.broz@uns.edu.ar

<sup>2</sup> Departamento de Economía, Universidad Nacional del Sur.  
12 de Octubre 1198, Bahía Blanca, Argentina.  
INMABB-CONICET  
ftohme@criba.edu.ar

<sup>3</sup> Facultad de Ciencias Forestales, Universidad Nacional de Misiones.  
Bertoni 124, El dorado, Argentina.

**Resumen.** El sublotado (*lot streaming*) es una de las metodologías más estudiadas en los problemas de secuenciación en configuraciones productivas tipo flow-shop, aunque no lo es el impacto del sublotado sobre el procesamiento propio de los trabajos. Es por eso, que en este trabajo, se estudia el impacto del efecto aprendizaje en base a la aplicación de sublotado. El efecto de aprendizaje aplicado considera el trabajo acumulado de los sublotes ya procesados del mismo producto. De esta forma no se sobredimensiona el aprendizaje. Se experimentó con instancias de mediana complejidad y se resolvió con GAMS/CPLEX. Se demuestra que el aprendizaje que se genera como consecuencia del sublotado es significativo en términos del *makespan*. Este resultado es más cercano a la realidad de los sistemas en los que el factor humano tiene importante injerencia.

**Palabras claves:** Sublotado, efecto aprendizaje, flow-shop, optimización.

**Abstract.** Lot streaming is one of the most studied techniques for solving flow-shop scheduling problems. However, the impact of lot streaming on job processing is not so well studied. In order of researching that impact, this paper studies learning effect in lot streaming. Learning effect is modeled considering only already produced sublots, avoiding overestimating learning consequences. For testing, medium-size instances are used and solved by GAMS/CPLEX. It is shown that the learning effect, fruit of applying lot streaming, has a meaningful impact in terms of makespan. These results describe better systems reality where human factor is important.

**Keywords:** lot streaming, learning effect, flow-shop, optimization.

## 1 Introducción

La planificación de la producción tiene como principal objetivo tomar decisiones de planificación optimizando los recursos cumpliendo los requisitos de los pedidos de los clientes (cantidad, fechas, etc.). El horizonte de planificación es el factor más influyente en el tipo de planificación a realizarse. En el horizonte de corto plazo la planificación se centra, principalmente, en la programación de tareas. La programación de tareas (*scheduling*) es un proceso de toma de decisiones que juega un importante papel en la mayoría de los sistemas de manufactura y producción [1], y el tipo de configuración productiva condiciona el tipo de programación a realizar. Una de las configuraciones más utilizadas en la producción y estudiada en la literatura es la configuración tipo flow-shop. En el flow-shop todos los trabajos a producir atraviesan por los mismos centros de trabajo con la misma secuencia de producción. Dentro de las metodologías para optimizar el problema del secuenciamiento de actividades en un flow-shop, el sublotteo reviste de especial interés (*lot streaming*). El sublotteo divide los trabajos a realizar (lotes de productos) en fracciones de esos lotes generando un solapamiento de actividades a lo largo del proceso productivo. Es decir, fracciona el lote de producto permitiendo que esas fracciones avancen a la siguiente etapa del sistema de producción en una fecha anterior a la que lo harían si se debiera esperar el procesamiento del lote entero. Esta forma de planificar permite reducir los tiempos máximos de procesamiento, reducir la cantidad de trabajo en proceso y también el ciclo de producción [2,3,4].

Desde las últimas décadas el proceso de *lot streaming* ha sido estudiado y se continúa estudiando con mucha profundidad, tal como se demuestra en [5]. Sin embargo, no se ha estudiado el efecto interno que genera en el sistema de procesamiento, es decir, cómo afecta e impacta en la producción propiamente, sólo se han hecho un análisis a nivel global del sistema, tal como: tiempos totales de procesamiento (*makespan*), tardanzas totales, niveles de trabajo en proceso, etc. De los efectos internos que se generan en una configuración flow-shop, en la que se aplica sublotteo, es de especial interés la repetición de tareas que de otra forma no se repetirían. Un ejemplo claro es el caso de remover las piezas ya procesadas del sublote del centro de trabajo. Si se procesa el lote entero la remoción se hace una sola vez, pero si se divide en  $n$  sublotes la remoción se repetirá  $n$  veces. Esto genera un aprendizaje sobre este tipo de tareas. El efecto aprendizaje tiene lugar en los procesos en donde se realizan operaciones idénticas consecutivamente y los trabajadores incorporan mayor aptitud para la operación por la repetición de las actividades, lo que se traduce en una reducción de los tiempos de operación.

En [6], se analiza el impacto del aprendizaje en el secuenciamiento de actividades trabajando con la programación en una sola máquina. En [7], se profundizó los resultados sobre la programación en una sola máquina y en [8], se estudió el caso de máquinas paralelas idénticas. En [9] se analiza por primera vez el efecto aprendizaje en configuraciones flow-shop, aunque fue limitado a dos máquinas. En la última década la literatura sobre el efecto aprendizaje en secuenciamiento de actividades ha sido muy prolífica [10,11,12,13,14,15].

Pese al auge bibliográfico del efecto aprendizaje en el secuenciamiento, las metodologías de subloteo no fueron incluidas en estos estudios. En [16], se considera por primera vez el subloteo y el efecto aprendizaje, resolviendo para una sola máquina y un solo producto, optimizando el valor actual neto. En [17], se extiende esto a más de un producto. En [18], se presenta un sistema en donde la producción tiene reprocesamientos y desperdicios, y lo resuelven utilizando subloteo. Sin embargo, limitan el estudio a un solo producto. Recientemente, en [19], se aplica el efecto aprendizaje a problemas de subloteo pero centrando el estudio en los métodos de resolución y ejemplificando solo para algunos casos.

Dado que en la bibliografía no hay estudios sobre el procesamiento de más de un producto en subloteo en configuraciones flow-shop optimizando criterios temporales, se pretende generar un aporte en ese aspecto. El enfoque aplicado en este trabajo considera sublotes consistentes como los define [2] en su libro, es decir, se mantienen constante en la diversas etapas, y de igual tamaño [18,20,4]. El resto de este trabajo se organiza de la siguiente manera. En la sección 2, se describe el modelo y el efecto aprendizaje. En la sección 3, se establece la experimentación. En la sección 4, se presentan los resultados. Y por último, en la sección 5, se presentan las conclusiones y las futuras líneas de investigación.

## 2 Presentación del problema

En esta sección, se describe la configuración flow-shop en la cual el efecto aprendizaje tiene injerencia y se resuelve el secuenciamiento aplicando subloteo. En una primera instancia se introduce el problema sin aprendizaje y sin subloteo para describir el sistema productivo en términos clásicos. En el apartado 2.1 se presentará el sistema aplicando subloteo y en el 2.2 aplicando subloteo y efecto aprendizaje.

Un aspecto a considerar es el cómputo del tiempo de procesamiento. En este caso se distinguen dos tipos de tiempos, uno asignado al tiempo de procesamiento unitario ( $pr_{(j,m)}$ ) para producir una unidad de producto  $j$  en la máquina  $m$ , y otro tiempo asignado a las tareas que son independientes del tiempo unitario ( $t_{(j,m)}$ ) las cuales se asignan cada vez que se procesa un nuevo lote o sublote. Este tiempo existe en todo proceso y representa la preparación de la máquina, la carga de materia prima, la lectura de datos previos, el traslado del contenedor del sublote, etc. Es de interés mencionar que si sólo se consideraran tiempos proporcionales a la cantidad de unidades contenidas en el sublote, el *makespan* se minimizaría con el sublote unitario [21].

Los supuestos que se hacen para el sistema de producción estudiado son: a) Las máquinas no pueden procesar más de un trabajo a la vez, b) Un trabajo no puede ser procesado por más de una máquina a la vez, c) Todas las máquinas están disponibles al inicio del horizonte de planificación, d) Todos los trabajos están disponibles al inicio del horizonte de planificación, e) Las máquinas pueden estar ociosas durante el horizonte de planificación, f) Los tiempos de procesamientos y setup son conocidos de antemano, g) No es posible interrumpir el procesamiento de un trabajo o sublote y h) Cada trabajo se procesa una sola vez en cada máquina y en el mismo orden.

Estos supuestos son los más utilizados para resolver configuraciones flow-shop de  $n$ -trabajos y  $m$ -máquinas minimizando el *makespan*. A continuación se presenta el modelo tradicional para flow-shop. El modelo está formulado mediante lógica posicional donde cada trabajo ocupa una posición en la secuencia de procesamiento.  $j$  hace referencia al producto y  $m$  a la máquina o etapa del flow-shop. Los parámetros del modelo son:  $U_{(j)}$  (tamaño del lote del producto  $j$ ),  $pr_{(j,m)}$  (tiempo de procesamiento unitario del producto  $j$  en la máquina  $m$ ),  $t_{(j,m)}$  (tiempo por sublote independiente del tiempo de procesamiento unitario en la máquina  $m$  del producto  $j$ ) y  $BigM$  (número positivo muy grande). Las variables son:  $C_{(j,p,m)}$  (fecha de finalización del lote del producto  $j$  en la posición  $p$  en la máquina  $m$ ),  $TP_{(j,m)}$  (tiempo total de ocupación de la máquina  $m$  por el producto  $j$ ) y  $x_{(j,p)}$  (variable binaria, es 1 si el producto  $j$  es procesado en la posición  $p$ ). El modelo queda expresado de ecuación 1 a 8.

$$Min \rightarrow C_{máx} \quad (1)$$

S.t.:

$$TP_{(j,m)} = pr_{(m,j)} * U_{(j)} + t_{(m,j)}, \forall m, j \quad (2)$$

$$\sum_{p=1}^p x_{(j,p)} = 1, \forall j \quad (3)$$

$$\sum_{j=1}^j x_{(j,p)} = 1, \forall p \quad (4)$$

$$C_{(j,p,m)} \geq TP_{(j,m)} - BigM * (1 - x_{(j,p)}), m = 1, p = 1, \forall j \quad (5)$$

$$C_{(j,p,m)} \geq C_{(j,p,m-1)} + TP_{(j,m)} - BigM * (1 - x_{(j,p)}), m > 1, \forall j, p \quad (6)$$

$$C_{(j,p,m)} \geq \sum_{j'} C_{(j',p-1,m)} + TP_{(j,m)} - BigM * (1 - x_{(j,p)}), p > 1, \forall j, m \quad (7)$$

$$C_{(j,p,m)}, TP_{(j,m)} \geq 0, \forall j, p, m \quad (8)$$

La ecuación (1) corresponde al objetivo a minimizar, el cual corresponde con el *makespan*. La ecuación (2) calcula el tiempo de procesamiento de cada producto  $j$  en cada máquina  $m$ . Esta ecuación será la más transformada por los distintos enfoques (subloteo y subloteo con aprendizaje). La ecuación (3) asegura que para cada trabajo  $j$  no esté asignada más que una posición  $p$ . La ecuación (4) verifica que cada posición  $p$  esté ocupada por un solo trabajo  $j$ . La ecuación (5) calcula el tiempo de finalización del trabajo  $j$  ubicado en la primera posición. La ecuación (6) comprueba que para que un trabajo  $j$  termine en una etapa tiene que haber terminado la etapa anterior. La ecuación (7) comprueba que para que el producto  $j$  de la posición  $p$  termine tiene que haber terminado el producto de la posición  $p-1$ . La ecuación (7) verifica las condiciones lógicas de no negatividad.

Esta configuración flow-shop procesa el total de las unidades contenidas en cada lote de productos antes de pasar al siguiente producto. Esto genera grandes tiempos ociosos en la producción. Para evitar esto se trabaja como método de solución el subloteo (*lot streaming*).

## 2.1 Flow-shop con subloteo

Para ampliar el modelo a flow-shop con subloteo es necesario incorporar un nuevo subíndice  $f$  el cual representa el sublote. Al incorporar subloteo la indexación de las variables se modifica. A continuación se presenta la nomenclatura propia del modelo con subloteo. Para conservar paralelismo con el modelo tradicional de flow-shop se conservan los mismos nombres de las variables y se las diferencia con los subíndices. Las variables quedan de la siguiente manera:  $s_{(f,j)}$  (cantidad de unidades de producto  $j$  en el sublote  $f$ ),  $C_{(f,j,p,m)}$  (fecha de finalización del sublote  $f$  del producto  $j$  en la posición  $p$  en la máquina  $m$ ),  $TP_{(f,j,m)}$  (tiempo total de ocupación de la máquina  $m$  por el sublote  $f$  del producto  $j$ ) y  $x_{(f,j,p)}$  (variable binaria, es 1 si sublote  $f$  del producto  $j$  es procesado en la posición  $p$ ). El modelo queda expresado de ecuación 9 a 18.

$$C_{(j,p,m)}, TP_{(j,m)} \geq 0, \forall j, p, m \quad (9)$$

S.t.:

$$\sum_f s_{(f,j)} = U_{(j)}, \forall j \quad (10)$$

$$s_{(f,j)} = s_{(f+1,j)}, \forall f, j \quad (11)$$

$$\sum_{p=1}^p x_{(f,j,p)} = 1, \forall f, j \quad (12)$$

$$\sum_j \sum_f x_{(f,j,p)} = 1, \forall p \quad (13)$$

$$TP_{(f,j,m)} = pr_{(m,j)} * s_{(f,j)} + t_{(m,j)}, \forall m, f, j \quad (14)$$

$$C_{(f,j,p,m)} \geq TP_{(f,j,m)} - BigM * (1 - x_{(f,j,p)}), m = 1, p = 1, \forall f, j \quad (15)$$

$$C_{(f,j,p,m)} \geq C_{(f,j,p,m-1)} + TP_{(f,j,m)} - BigM * (1 - x_{(f,j,p)}), m > 1, \forall f, j, p \quad (16)$$

$$C_{(f,j,p,m)} \geq \sum_{j'} \sum_{f'} C_{(f',j',p-1,m)} + TP_{(f,j,m)} - BigM * (1 - x_{(f,j,p)}), p > 1, \forall f, j, m \quad (17)$$

$$C_{(f,j,p,m)}, TP_{(f,j,m)}, s_{(f,j)} \geq 0, \forall f, j, p, m \quad (18)$$

La ecuación (9) corresponde al objetivo a minimizar, el cual corresponde con el *makespan*. La ecuación (10) expresa que la suma de los sublotes  $s_{(f,j)}$  debe asegurar el total

de unidades a producir del producto  $j$ . La ecuación (11) verifica que los sublotes sean iguales entre sí. Las ecuaciones (12) y (13) son similares a las (3) y (4) con la salvedad que verifican las posiciones por sublotes y no por producto. Una aclaración pertinente es que la cantidad de posiciones se modifica, ahora hay tantas posiciones como sublotes haya. La ecuación (14) es similar a la (2) sólo que ahora calcula el tiempo total para el sublote  $f$ . Para el primer sublote a procesar ( $p=1$ ) se calcula el tiempo de finalización con la ecuación (15). El sentido de las ecuaciones (15), (16) y (17) es el mismo que el de la (5), (6) y (7), respectivamente, pero calculando para un sublote y no para el lote entero. Las condiciones de no negatividad se verifican con la ecuación (18).

Esta situación es la que realmente se suele representar en la bibliografía, un secuenciamiento que fracciona la producción con el objeto de lograr minimizar el *makespan*. Sin embargo, la partición obliga a la repetición de actividades, por ejemplo carga y descarga de las máquinas, provocando aprendizaje sobre estas actividades.

## 2.2 Flow-shop con subloteo y efecto aprendizaje

El aprendizaje tiene gran impacto en los procesos en que la mano de obra es un factor productivo significativo. En la bibliografía sobre aprendizaje suelen considerarse dos enfoques para representarlo: efecto aprendizaje basado en las posiciones y efecto aprendizaje basado en la suma del tiempo procesado [12]. El aprendizaje basado en posiciones representa el aprendizaje que se da en las tareas que son independientes del tiempo de procesamiento unitario, como lectura de datos, preparación de máquinas, etc. La repetición de estas tareas genera el aprendizaje. En cambio, el aprendizaje basado en la suma del tiempo procesado representa los sistemas en que el factor humano tiene un impacto importante en el procesamiento, por lo que, al ir acumulando experiencia en el procesamiento de trabajos previos se genera el aprendizaje.

En este trabajo se presenta un enfoque lineal [22], el cual describe el aprendizaje como un ajuste al tiempo estándar de procesamiento  $p_i$  (ecuación (19)).

$$p_{ir} = p_i - v_i * r \quad (19)$$

Donde  $p_{ir}$  es el tiempo de procesamiento del trabajo  $i$  en la posición  $r$  y  $v_i$  es el coeficiente de aprendizaje para ese trabajo el cual cumple  $v_i < p_i / n$  para asegurar tiempos positivos de procesamiento  $p_{ir}$ . Algo objetable de esta representación es que no considera el tipo de trabajo que hizo previamente, el aprendizaje es proporcional a la cantidad de unidades ya procesadas sin importar que tan distintas sean esas operaciones. Otro aspecto a comentar es propio del tipo de aprendizaje por posiciones, que no considera el tiempo de los procesos anteriores. Estas objeciones toman relevancia en la medida en que los productos a procesar tengan diferencias considerables en sus procesamientos, y más aún en situaciones de subloteo, porque la cantidad de trabajo previo es flexible. Por lo expuesto, se presenta como expresión de aprendizaje la ecuación (20).

$$TP_{(f,j,m)}^* = TP_{(f,j,m)} - LR_{(j,m)} * \sum_f^{f-1} s_{(f,j)} \quad (20)$$

Donde  $TP_{(f,j,m)}^*$  es el tiempo de procesamiento sujeto a aprendizaje.  $LR_{(j,m)}$  es el coeficiente de aprendizaje del producto  $j$  en la máquina  $m$ . De esta forma se computa el aprendizaje aprendido sólo por el procesamiento de las unidades iguales ya procesadas de ese producto para esa operación. De esta forma el aprendizaje contempla la repetición de actividades iguales y la cantidad ya realizada. El coeficiente  $LR_{(j,m)}$  cumple una condición similar a la impuesta al parámetro  $v_i$  de [22], pero en este caso  $LR_{(j,m)} < (pr_{(j,m)} + t_{(j,m)}) / U_{(j)}$ , el cual asegura no negatividad en la variable  $TP_{(f,j,m)}^*$ . Entonces el modelo de flow-shop con subleoteo y efecto aprendizaje es similar al presentado en el apartado 2.1. Sólo se modifican las ecuaciones (16) y (17) por (21) y (22), respectivamente. Cabe aclarar que la ecuación (13) no se ve afectada por el aprendizaje, dado que es el primer trabajo a realizarse en esa máquina, por lo que no hay conocimiento por repetición de actividades.

$$C_{(f,j,p,m)} \geq C_{(f,j,p,m-1)} + TP_{(f,j,m)}^* - BigM * (1 - x_{(f,j,p)}), m > 1, \forall f, j, p \quad (21)$$

$$C_{(f,j,p,m)} \geq \sum_{j'} \sum_{j''} C_{(f',j',p-1,m)} + TP_{(f,j,m)}^* - BigM * (1 - x_{(f,j,p)}), p > 1, \forall f, j, m \quad (22)$$

Para ilustrar el impacto del efecto aprendizaje modelado se presentan las Figuras 1 y 2. Los diagramas de Gantt mostrados en ambas figuras corresponden a una configuración flow-shop de 2 productos (A y B) a dividirse en 2 sublotes (A1, A2, B1 y B2) iguales para ser procesados en 3 máquinas. En ambas figuras se consideran sublotes de igual tamaño, es decir, la cantidad de unidades de producto A en cada sublote es la misma, análogo para el producto B. En la Figura 1 los tiempos que cada sublote requiere en cada máquina son iguales para cada producto. En cambio, en la Figura 2 donde se considera efecto aprendizaje, se muestra como el procesamiento del segundo sublote de cada producto insume menos tiempo, redundando en un *makespan* menor.

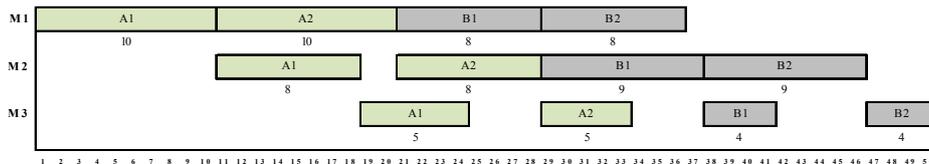


Figura 1. Diagrama de Gantt para una configuración flow-shop (sin efecto aprendizaje).

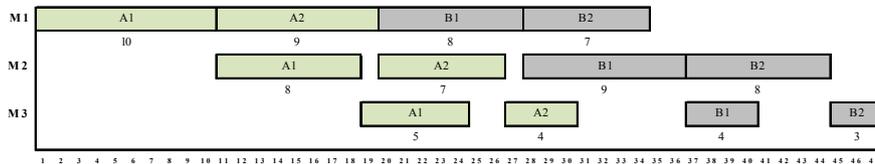


Figura 2. Diagrama de Gantt para una configuración flow-shop (con efecto aprendizaje).

### 3 Experimentaciones del modelo

Se realizaron diferentes ensayos sobre distintos escenarios para poder comparar y analizar el impacto del efecto aprendizaje en configuraciones flow-shop en los que se pueda generar subloteo. Los escenarios seleccionados fueron de 2, 4 y 6 productos a procesarse en distintas configuraciones productivas de 5, 10 y 15 máquinas. Cada producto tiene asociado un lote de producción  $U(j)$  que cumple con las demandas. Para los distintos escenarios se simularon los datos: los lotes de producción se obtuvieron con una distribución uniforme entre 50 y 150 unidades, los tiempos de procesamiento entre 5 y 15, y los tiempos independientes del tamaño del sublote entre 40 y 80. Cada instancia del problema fue resuelto por las tres formas presentadas: flow-shop tradicional, flow-shop con subloteo y flow-shop con subloteo y efecto aprendizaje. En los casos de subloteo la cantidad de sublotes fue de 2, 4 y 6. Todas las instancias se resolvieron con GAMS/CPLEX.

Se experimentaron los modelos en 27 instancias diferentes (3 variantes en productos, 3 en máquinas y 3 en sublotes), salvo en el caso de flow-shop tradicional en donde las variantes de sublotes no lo afectaron. Para cada una de las 27 instancias se simularon 3 escenarios distintos.

### 4 Resultados

El promedio de los resultados obtenidos se presentan en la Tabla 1 en unidades de tiempo (u.t.). En ella se diferencia las instancias por cantidad de productos, cantidad de máquinas y de sublotes en el caso de que lo afecte.

$j$	Instancias		Sin subloteo (u.t.)	Con subloteo (u.t.)	Con subloteo y efecto aprendizaje (u.t.)	
	$m$	$f$				
2	5	2		1243	1211	
		4	1444	1328	1160	
		6		1534	1116	
	10	2			2020	1965
			4	2670	1888	1698
			6		2042	1620
		15	2		2757	2696
			4	3845	2422	2216
			6		2507	2084
4	5	2		1883	1808	
		4	1971	2239	1850	
		6		2708	1832	
	10	2			2661	2568
			4	3173	2817	2388
			6		3239	2305

		2		3434	3323
	15	4	4385	3393	2945
		6		3751	2812
		2		2530	2428
	5	4	2496	3126	2566
		6		3834	2624
		2		3307	3193
6	10	4	3686	3734	3182
		6		4429	3195
		2		4130	4007
	15	4	4985	4326	3741
		6		4949	3669

**Table 1.** Resultados promedios para cada una de las instancias.

Como primer acercamiento a los resultados uno comprueba que en general al aplicar subloteo la función objetivo mejora, lo cual concuerda con la bibliografía [23,21,24,3,25,4,26]. Este mejoramiento del *makespan* se evidenció en el 66% de los casos, logrando mejoras de hasta el 37% para 2 productos, 15 máquinas y 4 sublotes. Comparando los problemas de subloteo consigo mismo, puede notarse que en algunos casos al aumentar el número de sublotes el *makespan* mejoró y en otros no, especialmente para 4 y 6 productos. Este empeoramiento se debe a que al aumentar los sublotes crece el número de tareas fijas, alcanzando, incluso, a ser peor que el valor para el flow-shop tradicional. Esto coincide con los resultados de [21], en donde la reducción del *makespan* de una configuración flow-shop es proporcional al número de sublotes hasta cierto punto. Luego, si se sigue fraccionando el lote, el *makespan* comienza a crecer.

<i>j</i>	Instancias		Sin subloteo	Con subloteo
	<i>m</i>	<i>f</i>		
		2	16%	3%
	5	4	20%	13%
		6	23%	27%
		2	26%	3%
2	10	4	36%	10%
		6	39%	21%
		2	30%	2%
	15	4	42%	9%
		6	46%	17%
		2	8%	4%
4	5	4	6%	17%
		6	7%	32%

		2	19%	3%
	10	4	25%	15%
		6	27%	29%
		2	24%	3%
	15	4	33%	15%
		6	36%	29%
		2	3%	4%
	5	4	-3%	18%
		6	-5%	32%
		2	13%	3%
6	10	4	14%	15%
		6	13%	28%
		2	20%	3%
	15	4	25%	14%
		6	26%	26%

**Table 2.** Diferencia porcentual entre *makespan* con subloteo y efecto aprendizaje y el resto de los casos. El positivo implica el porcentaje de ganancia.

Sin embargo, ese tipo de tareas fijas, generalmente, son muy demandantes de mano de obra (lectura de datos, preparación de máquina, carga de materia prima, remoción de trabajo procesado, limpieza de máquina, etc.), por lo que es correcto analizar el impacto del efecto aprendizaje. Los resultados de subloteo con efecto aprendizaje evidencian ese impacto. La Tabla 2 muestra la diferencia porcentual del *makespan* obtenidos con subloteo con efecto aprendizaje respecto de los otros modelos. En todos los casos se logra mejorar el *makespan* alcanzado considerando sólo subloteo, alcanzando el 32% de reducción para 4 productos, 5 máquinas y 6 sublotes. También, se obtienen menores resultados que con el flow-shop tradicional donde la reducción alcanza hasta el 46%. Aunque en los casos de 6 productos, 5 máquinas y más de 4 sublotes la función objetivo empeora, sigue siendo mejor el desempeño con aprendizaje que con sólo subloteo.

## 5 Conclusión

En este trabajo se analizó y estudió los efectos internos que genera aplicar subloteo en un problema de secuenciamiento de flow-shop y desde esa perspectiva se analizó el impacto del efecto aprendizaje. El subloteo no sólo es útil como metodología de secuenciamiento, sino que además, brinda el entorno apropiado para que aquellos sistemas en los que la mano de obra es un factor determinante, se genere aprendizaje. El impacto del efecto aprendizaje en los resultados obtenidos, confirma que es un aspecto a considerar y tener en cuenta, incluso para un enfoque conservador como el aplicado en este trabajo. Esto, a su vez, se encuentra respaldado por experimentacio-

nes realizadas en instancias de mediana complejidad. Se propone como futuras líneas de investigación, ampliar el análisis a otros enfoques de efectos aprendizaje como también a considerar otro tipo de objetivos temporales.

## Referencias

1. Pinedo, M. L. (2012). *Scheduling: theory, algorithms, and systems*. Springer Science & Business Media
2. Sarin, S. C., & Jaiprakash, P. (2007). *Flow shop lot streaming*. Springer Science & Business Media.
3. Defersha, F. M., & Chen, M. (2010). A hybrid genetic algorithm for flow-shop lot streaming with setups and variable sublots. *International Journal of Production Research*, 48(6), 1705-1726.
4. Ventura, J. A., & Yoon, S. H. (2013). A new genetic algorithm for lot-streaming flow shop scheduling with limited capacity buffers. *Journal of Intelligent Manufacturing*, 24(6), 1185-1196.
5. Cheng, M., Mukherjee, N. J., & Sarin, S. C. (2013). A review of lot streaming. *International Journal of Production Research*, 51(23-24), 7023-7046.
6. Biskup, D. (1999). Single-machine scheduling with learning considerations. *European Journal of Operational Research*, 115(1), 173-178.
7. Mosheiov, G. (2001<sup>a</sup>). Scheduling problems with a learning effect. *European Journal of Operational Research*, 132(3), 687-693.
8. Mosheiov, G. (2001<sup>b</sup>). Parallel machine scheduling with a learning effect. *Journal of the Operational Research Society*, 1165-1169.
9. Lee, W. C., & Wu, C. C. (2004). Minimizing total completion time in a two-machine flow-shop with a learning effect. *International Journal of Production Economics*, 88(1), 85-93.
10. Wang, J. B., & Xia, Z. Q. (2005). Flow-shop scheduling with a learning effect. *Journal of the Operational Research Society*, 56(11), 1325-1330.
11. Kuo, W. H., & Yang, D. L. (2006). Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect. *European Journal of Operational Research*, 174(2), 1184-1190.
12. Biskup, D. (2008). A state-of-the-art review on scheduling with learning effects. *European Journal of Operational Research*, 188(2), 315-329.
13. Ji, M., & Cheng, T. E. (2010). Scheduling with job-dependent learning effects and multiple rate-modifying activities. *Information Processing Letters*, 110(11), 460-463.
14. Wang, J. B., Wang, M. Z., & Ji, P. (2012). Single machine total completion time minimization scheduling with a time-dependent learning effect and deteriorating jobs. *International Journal of Systems Science*, 43(5), 861-868.
15. Wang, J. B., & Wang, J. J. (2014). Flow-shop scheduling with a general exponential learning effect. *Computers & Operations Research*, 43, 292-308.
16. EYNAN, A., & Li, C. L. (1997). Lot-splitting decisions and learning effects. *IIE transactions*, 29(2), 139-146.
17. Serin, Y., & Kayaligil, S. (2003). Lot splitting under learning effects with minimal revenue requirements and multiple lot types. *IIE Transactions*, 35(8), 689-698.
18. Jaber, M. Y., & Khan, M. (2010). Managing yield by lot splitting in a serial production line with learning, rework and scrap. *International Journal of Production Economics*, 124(1), 32-39.

19. Mukherjee, N. J., & Sarin, S. C. (2014). Lot streaming in the presence of learning. *International Journal of Planning and Scheduling*, 2(1), 40-52.
20. 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*, 181(12), 2455-2468.
21. Feldmann, M., & Biskup, D. (2008). Lot streaming in a multiple product permutation flow shop with intermingling. *International Journal of Production Research*, 46(1), 197-216.
22. Bachman, A., & Janiak, A. (2004). Scheduling jobs with position-dependent processing times. *Journal of the Operational Research Society*, 55(3), 257-264.
23. Biskup, D., & Feldmann, M. (2006). Lot streaming with variable sublots: an integer programming formulation. *Journal of the Operational Research Society*, 57(3), 296-303.
24. Wong, T. C., Chan, F. T., & Chan, L. Y. (2009). A resource-constrained assembly job shop scheduling problem with lot streaming technique. *Computers & Industrial Engineering*, 57(3), 983-995.
25. Defersha, F. M., & Chen, M. (2012). Mathematical model and parallel genetic algorithm for hybrid flexible flow-shop lot streaming problem. *The International Journal of Advanced Manufacturing Technology*, 62(1-4), 249-265.
26. Framinan, J. M., Leisten, R., & García, R. R. (2014). *Manufacturing Scheduling Systems*. Springer.