

Aproximación lineal para problema de scheduling flow shop con efecto de aprendizaje no-lineal

Augusto Ferraro¹, Daniel Alejandro Rossit^{1,2}, Mariano Frutos¹ e Ignacio Grossmann³

¹ Departamento de Ingeniería, Universidad Nacional del Sur, Bahía Blanca, Argentina.

² INMABB, CONICET, Bahía Blanca, Argentina.

³ Chemical Engineering Department, Carnegie Mellon University, Pittsburgh, PA, USA.

augusto.ferraro@uns.edu.ar - daniel.rossit@uns.edu.ar

mfrutos@uns.edu.ar - grossmann@cmu.edu

Abstract. El desempeño de los procesos de manufactura suele depender de las capacidades y habilidades de los recursos productivos involucrados, generalmente maquinaria y mano de obra. Es sabido que la aptitud de la mano de obra es afectada por el aprendizaje, a mayor aprendizaje mayor aptitud. En entornos competitivos como el actual en donde los procesos de manufactura deben ajustarse a constantes cambios para lograr satisfacer demandas cada vez más exigentes y que requieren especificaciones particulares, esta capacidad de aprendizaje se vuelve un factor importante dentro del sistema. Es por eso que en muchos casos la planificación de la producción ya contempla este aprendizaje dentro del proceso, para sacar el mayor rédito posible. Para ello deben modelarse fenómenos complejos que permitan estimar la capacidad de producción a medida que se mejora la aptitud de producción, dando lugar a expresiones del tipo no-lineal. Esto limita el tipo de herramientas computacionales para su resolución. En este trabajo se aborda esta limitación, proponiendo métodos de aproximación lineal que permitan resolver esos problemas de planificación minimizando los requerimientos de software y capacidades computacionales. Los métodos propuestos son comparados contra métodos capaces de resolver problemas mixto-entero no lineales, y los resultados muestran una buena calidad del método aquí propuesto.

Keywords: Manufactura; Flow Shop Scheduling; efecto aprendizaje, Mixto-entero no-lineal.

1 Introducción

El estudio del fenómeno del aprendizaje en las operaciones de producción del tipo manufactura fue abordado de manera científica a principios del siglo XX [1]. El estudio seminal al respecto fue el desarrollado por Wright [2], quien en su trabajo demostró que, al duplicar la cantidad de unidades producidas, la cantidad de horas de mano de obra directa para producir una unidad decrece a una tasa constante, cuyo valor depende del proceso particular que se esté observando. La curva que relaciona el tiempo de producción con la cantidad de unidades producidas, llamada curva de aprendizaje, adquirió gran relevancia en años posteriores, sobre todo en la Segunda Guerra Mundial para

predecir costos y requerimientos de tiempo de los contratistas del gobierno estadounidense en la industria aeronáutica.

Ya en la segunda mitad del siglo XX, Yelle [3] realizó un estudio exhaustivo sobre el tema, recopilando y organizando todos los avances hasta ese momento. En su revisión, Yelle, presenta los modelos alternativos a la curva de Wright que reflejan con mayor precisión el fenómeno del aprendizaje, como también la estimación de parámetros para esos modelos alternativos. Por otra parte, Yelle también muestra algunas aplicaciones de la curva de aprendizaje, como la fijación de estándares de trabajo y su uso para el pronóstico de los requerimientos de mano de obra. Además, indica los estudios realizados sobre la relación de la curva de aprendizaje con el sistema de control de gestión, estándares de presupuestos y la planificación agregada, entre otros.

Es recién en 1999 cuando Biskup [1] hace foco en el efecto aprendizaje aplicado a problemas de secuenciación, en particular en los de una sola máquina. En su trabajo, demuestra que estos problemas siguen siendo resolubles en tiempo polinomial para los objetivos considerados.

En su revisión del 2008, Biskup [4] indica algunas características claves de los ambientes de producción en los que intervienen los problemas de secuenciación y que tienen un alto uso del factor humano. Entre ellas destaca la preparación de cada máquina para llevar a cabo la operación, su limpieza luego del procesamiento de un trabajo, la operación y el control de las máquinas, su mantenimiento planificado, la lectura, entendimiento e interpretación de la información de la maquinaria y todo tipo de trabajos manuales, ya sea durante el tiempo de procesamiento regular o luego para corregir errores. Biskup advierte que, en los casos en los que se procese por años la misma variedad de productos, los trabajadores sean los mismos, y las maquinarias no se modifiquen, el efecto aprendizaje alcanza su máximo nivel, y ya no se observan mejoras significativas. Sin embargo, cuando el ambiente de producción cambia, ya sea por la incorporación de empleados sin experiencia, por nuevas inversiones, recambio de maquinaria o renovación de productos, el efecto aprendizaje vuelve a tener impacto. Otros ejemplos de pequeñas modificaciones en el sistema productivo en los que puede observarse el fenómeno de efecto aprendizaje pueden ser, una actualización de un software, un nuevo diseño del formato de documentos importantes como la lista de materiales o una nueva organización del depósito de piezas de repuesto.

Un aspecto clave para poder enlazar el efecto aprendizaje con los problemas de secuenciación es la forma de modelarlo. Biskup indica claramente dos opciones: el aprendizaje basado en la posición y el enfoque basado en la suma de los tiempos de procesamiento. La primera implica que el aprendizaje se produce por la cantidad de veces que los productos fueron procesados. En cambio, la segunda toma en cuenta el tiempo de procesamiento de todos los productos que ya han sido procesados. Cada uno, mantiene Biskup, tiene su propia validez.

A partir del trabajo de Biskup de 1999, se ha puesto mucha atención en la secuenciación de operaciones con efecto aprendizaje (véase [5-14]). Todos ellos se refieren al problema de secuenciación con efecto aprendizaje en una sola máquina o en dos, ofreciendo soluciones óptimas mediante reglas de secuenciación.

La mayoría de los avances anteriormente descriptos no enfocan su atención en el problema flow shop con múltiples máquinas. En este trabajo se presenta un modelo de

programación matemática lineal mixto entero para resolver un problema de flow shop con efecto aprendizaje, que tiene en consideración la representación del efecto aprendizaje de forma no lineal. Para ello, se aproxima la curva de aprendizaje propuesta en [11] con restricciones lineales, las cuales se obtienen a partir de resolver un problema no lineal continuo. En la experimentación, se resuelven problemas de hasta 15 máquinas y 8 trabajos con el modelo propuesto y se comparan los resultados con el problema sin considerar el efecto aprendizaje. Este mismo problema fue abordado inicialmente en [16], y aquí se propone una extensión de ese trabajo al modificar la forma en que se generarán los puntos iniciales utilizados para resolver el problema no lineal continuo.

La motivación del enfoque adoptado en este trabajo es la misma que la adoptada en [16], generar métodos de optimización capaces de aprovechar herramientas de software lo más básicas posibles (del tipo de herramientas con las que suelen contar los planificadores de PYMES) para resolver problemas complejos como el problema mixto-entero no lineal aquí considerado. Aunque en este caso, se propone seguir mejorando y ajustando estos métodos respecto a los propuestos en [16].

2 Presentación del problema y descripción del modelo

El problema a resolver consiste en encontrar la secuenciación óptima de N trabajos que van a ser procesados en M máquinas de forma tal, de conseguir el menor tiempo de producción total y considerando que los tiempos de producción de cada trabajo no son constantes, sino que existe el fenómeno de aprendizaje. El impacto de este fenómeno es proporcional al tiempo de producción acumulado, y dado que se trata de un problema con productos distintos entre sí, el tiempo de producción acumulado para un producto dado, será dependiente de la secuencia pasada. Este fenómeno fue modelado originalmente por Wang y otros [11] de la siguiente forma:

$$p_{jk} = p_{jk}^* \left(\alpha l^{\sum_{i=1}^N p_{ik}^{x_{ji}}} + \beta \right) \quad (1)$$

donde p_{jk} es el tiempo de procesamiento considerando el efecto aprendizaje del trabajo j en la máquina k , p_{jk}^* es el tiempo de procesamiento estándar del trabajo j en la máquina k , α y β son constantes que representan la proporción del trabajo que puede ser eliminada con el efecto del aprendizaje y la que no, respectivamente. Además, α y β , deben cumplir $\alpha + \beta = 1$. La constante l ($0 < l < 1$) es una constante que indica la velocidad de aprendizaje y mientras más cercana sea a 0, mayor será la rapidez de aprendizaje. La variable x_{ji} es una variable binaria cuyo valor será uno si el trabajo j' precede al trabajo j y 0, en caso contrario. La consecuencia de utilizar la Ecuación (1) como representación del efecto aprendizaje es que los tiempos de procesamiento reales variarán entre el tiempo de producción estándar como máximo y una proporción β de ese tiempo de procesamiento como cota inferior. Una representación gráfica de la forma en la que varía el tiempo de procesamiento real con el tiempo de producción pasado puede encontrarse en la Figura 1.

2.1 Modelo con efecto aprendizaje – Mixto-entero no lineal (MINLP)

El modelo real para resolver el problema previamente definido se indica a continuación:

Conjuntos:

j : índice para los trabajos, $j = 1, 2, \dots, N$

k : índice para las máquinas, $k = 1, 2, \dots, M$

Parámetros:

p^*_{jk} : tiempo de procesamiento estándar del trabajo j en la máquina k

G : número arbitrariamente grande

α : proporción del trabajo que presenta efecto aprendizaje

β : proporción del trabajo que no presenta efecto aprendizaje

l : coeficiente de aprendizaje

Variables:

c_{max} : tiempo de finalización de todos los trabajos

c_{jk} : tiempo de finalización del trabajo j en la máquina k

p_{jk} : tiempo de procesamiento estándar del trabajo j en la máquina k

$x_{jj'}$: variable binaria que toma valor 1 si el trabajo j' está antes que j y 0, en caso contrario

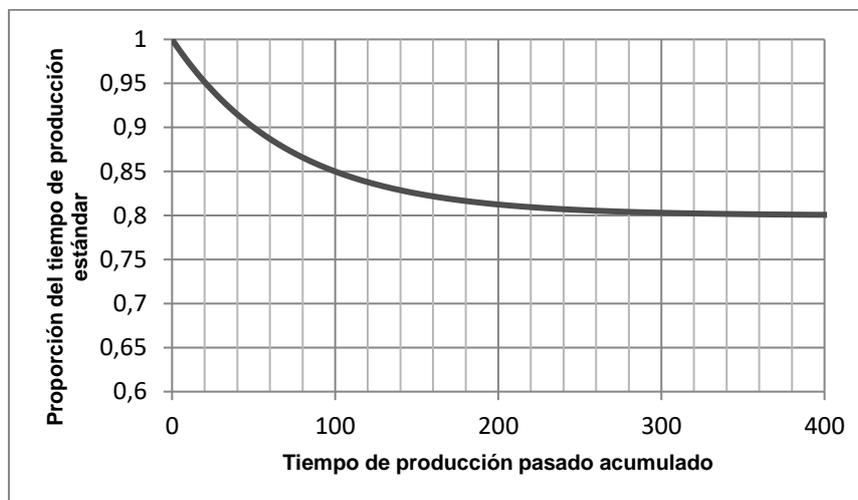


Fig. 1. Curva de efecto aprendizaje ($\beta=0,8$ y $\alpha=0,2$)

Función objetivo:

$$\text{Min } z = c_{max} \quad (2)$$

Restricciones:

$$c_{jk} - p_{jk} \geq c_{j'k} - (1 - x_{j'j})G \quad \forall j, j', k (j \neq j') \quad (3)$$

$$x_{j'j} + x_{jj'} = 1 \quad \forall j, j' (j \neq j') \quad (4)$$

$$c_{jk} \geq c_{jk-1} + p_{jk} \quad \forall j, k \neq 1 \quad (5)$$

$$c_{j1} \geq p_{j1} \quad \forall j \quad (6)$$

$$c_{max} \geq c_{jM} \quad \forall j \quad (7)$$

$$p_{jk} = p_{jk}^* \left(\alpha \lambda^{\sum_{i=1}^N p_{ik}^* x_{ij}} + \beta \right), \quad \forall j, k \quad (8)$$

$$x_{j'j} \in \{0,1\} \quad \forall j \neq j'; c_{jk}, p_{jk} \geq 0 \quad (9)$$

La Ecuación (2) indica que la función objetivo a minimizar es el tiempo total del programa de producción. La Ecuación (3) indica que el instante de inicio de un trabajo debe ser posterior al de uno que se secuenció antes en la misma máquina. La Ecuación (4) indica que, si un trabajo j precede a otro j' , este último no puede preceder a j . La Ecuación (5) sostiene que para que un trabajo sea procesado en una máquina, debe ser procesado previamente en la máquina anterior de la secuencia tecnológica. La Ecuación (6) expresa que el instante de finalización de todo trabajo secuenciado en la primera máquina no puede ser menor que su tiempo de procesamiento. La Ecuación (7) asegura que el tiempo de finalización del programa de producción sea el máximo de los tiempos de finalización de cada trabajo en la última máquina. La Ecuación (8) indica el valor del tiempo de procesamiento real. Mientras que la ecuación (9) representa las condiciones de factibilidad.

Este modelo matemático tiene la dificultad de que la Ecuación (8), en particular, indica una dependencia exponencial entre las variables. De esta forma el modelo se clasifica como mixto entero no lineal. La intención es, entonces, aproximar la curva de aprendizaje de forma tal que el modelo resultante pueda ser un modelo de programación matemática mixto entero lineal. En la siguiente sección, se explica el enfoque utilizado.

2.2 Modelo aproximado – Mixto-entero lineal (MILP)

La curva aprendizaje a aproximar es la expresada en (1), y considerando específicamente la expresión encerrada entre paréntesis, la misma puede expresarse de la siguiente forma:

$$p_{jk} = p_{jk}^*(y) \quad (10)$$

En la Ecuación (10), la expresión no lineal entre paréntesis en (1) está representado por y . Esta variable auxiliar y representa el efecto aprendizaje que se pretende modelar. Este efecto aprendizaje tal como se ve en (1) tiene una fórmula exponencial, que impide abordar el problema como un simple modelo mixto-entero lineal, entrando en los sistemas no lineales. Dentro de los problemas que esto genera, uno de los más importantes es el de la multimodalidad, lo cual implica la existencia de óptimos locales. En consecuencia, resolver el problema de forma óptima exige métodos de optimización adecuados que generalmente requieren un mayor esfuerzo computacional, derivando en una

restricción en la capacidad de resolución de problemas flow shop de instancias medianas y grandes. Por eso se presenta la siguiente aproximación para salvar esa complejidad, representando de forma genérica la no linealidad en cuestión en la Ecuación (11).

$$y = \alpha l^x + \beta \quad (11)$$

La Ecuación (11) presenta la expresión genérica no lineal de (1). En donde α , β y l siguen teniendo el mismo sentido de ser constantes, y la variable x indica el exponente. En nuestro caso, x reemplaza el tiempo de procesamiento acumulado en trabajos anteriores de la expresión (1). Dado que el valor de x depende de los trabajos secuenciados previos, se la modelará en términos de los valores promedios. Para ello si se requiere secuenciar N trabajos, el último trabajo tendrá un tiempo acumulado precedente de procesamiento promedio de $(N - 1)\bar{p}^*$, siendo \bar{p}^* el tiempo de procesamiento promedio de los trabajos. Por lo tanto nuestra aproximación de la función será entre $x = 0$ y $x = (N - 1)\cdot\bar{p}^*$. Es así que dicha aproximación será distinta si el problema a considerar tiene una cantidad de trabajos a secuenciar distinta.

Para aproximar la curva de aprendizaje, se considerará el siguiente razonamiento: la Ecuación (8) debería reemplazarse por otra expresión que sea lineal para que el modelo resultante dé lugar a un problema de programación lineal mixto entero. Sin embargo, una propuesta alternativa consiste en aproximar la Ecuación (11) por el mayor valor de un conjunto de ecuaciones lineales de la siguiente manera:

$$y = \max\{y_1, y_2, y_3\} \quad (12)$$

$$y_1 = a_1 - b_1 \cdot x \quad (13)$$

$$y_2 = a_2 - b_2 \cdot x \quad (14)$$

$$y_3 = a_3 - b_3 \cdot x \quad (15)$$

Las ecuaciones (12-15) consideran un número arbitrario de tres ecuaciones a los fines de poder instanciar la propuesta, en donde $a_1 > a_2 > a_3$ y $b_1 > b_2 > b_3$. Estas constantes deberán elegirse de manera tal de poder minimizar el error producido por la aproximación. Nos ocuparemos de este tema luego en la Sección 2.3 de este trabajo. Si bien las Ecuaciones (13), (14) y (15) son lineales, la ecuación (12) continúa sin satisfacer las condiciones para desarrollar un modelo lineal. Aun así, es posible reemplazar las cuatro ecuaciones anteriores por las siguientes tres desigualdades (16), (17) y (18), ya que es posible demostrar que alguna de las desigualdades será activa:

$$y \geq a_1 - b_1 \cdot x \quad (16)$$

$$y \geq a_2 - b_2 \cdot x \quad (17)$$

$$y \geq a_3 - b_3 \cdot x \quad (18)$$

Con estas tres ecuaciones se logra el mismo resultado que con las cuatro anteriores (12-15). En conclusión, la Ecuación (8) de nuestro modelo no lineal puede ser aproximada por las siguientes tres restricciones lineales:

$$p_{jk} \geq p_{jk}^* \left(a_1 - b_1 \cdot \sum_{i=1}^N p_{ik}^* y_{ij} \right) \quad (19)$$

$$p_{jk} \geq p_{jk}^* \left(a_2 - b_2 \cdot \sum_{i=1}^N p_{ik}^* y_{ij} \right) \quad (20)$$

$$p_{jk} \geq p_{jk}^* \left(a_3 - b_3 \cdot \sum_{i=1}^N p_{ik}^* y_{ij} \right) \quad (21)$$

De esta manera la curva de aprendizaje podrá ser aproximada por tres funciones lineales tal como se muestra en la Figura 2:

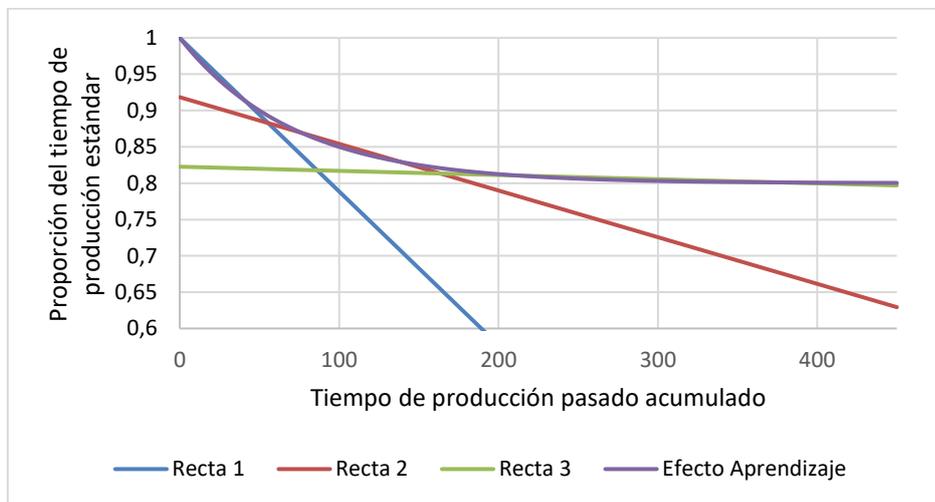


Fig. 2. Aproximación de la curva de efecto aprendizaje

La determinación del número de rectas a utilizar para la aproximación de la curva de efecto aprendizaje depende de la precisión que se pretenda conseguir, para este trabajo se proponen tres rectas. Para mayores detalles véase [16]. Esto permite un buen grado de aproximación, así como también una correcta presentación del enfoque de aproximación.

2.3 Estimación de parámetros

La cuestión radica ahora en asignar valores a los parámetros a y b de las ecuaciones (18-20) de forma de conseguir la mejor aproximación posible. En primer lugar, dado que todos los trabajos que se secuencian en la primera máquina no tienen efecto aprendizaje, la constante a_1 deberá tomar necesariamente el valor 1. Para determinar el resto de los parámetros se elige aproximar la curva real de efecto aprendizaje siguiendo el

criterio de mínimos cuadrados. Es decir, se elegirán a_2, a_3, b_1, b_2 y b_3 de forma tal que se minimice la siguiente expresión:

$$\sum_{i=1}^{i=3} \int_{x_{i-1}}^{x_i} (y_i - y)^2 dx \quad (22)$$

La Ecuación (21) indica la suma de tres integrales, las cuales están definidas entre dos valores de x , con $x_0 = 0$ y $x_3 = (N - 1) \cdot \bar{p}^*$. El resto de los extremos de las integrales surgen de la intersección de las rectas. La función y representa la curva de aprendizaje real y la y_i la recta que la aproxima en el tramo i . Minimizando esta expresión es posible determinar los valores óptimos de los parámetros para cada cantidad de trabajos a secuenciar considerada. Para lograr resolver el problema planteado en (21) se plantea utilizar un modelo de mínimos cuadrados, el cual también queda no-lineal, pero en este caso es un problema continuo (el detalle del modelo de mínimo cuadrados puede encontrarse en [16], ecuaciones 21-27). Para resolver este problema, se propone utilizar el mismo algoritmo que en Ferraro et al. [16], el cual utiliza el solver GRG de Excel. Ese algoritmo, básicamente resuelve de manera iterativa a partir de puntos iniciales, aquellos valores de a y b (para un número de rectas establecido) que mejor aproximen la curva exponencial en cuestión. Como la gran mayoría de los métodos de optimización no-lineal, GRG mejora sustancialmente su rendimiento si se le otorgan puntos iniciales de buena calidad. Es aquí donde este trabajo incorpora mejoras respecto a [16].

En [16] para generar esos puntos iniciales se utilizó un muestreo aleatorio de una distribución de probabilidad *ad hoc*, la cual fue diseñada siguiendo el criterio de la derivada segunda. Es decir, la función de probabilidad acumulada era proporcional a la derivada segunda de la curva, de esta forma se pudo asignar mayor probabilidad a los puntos cercanos al “codo” de la curva, lo cual es más difícil de modelar con trazos rectos, por lo que es razonable utilizar mayor cantidad de trazos rectos, mientras que la parte del *plateau* de la curva exponencial no es necesario utilizar tantos trazos.

En este caso se utilizará una lógica similar, pero considerando otro recurso de análisis para modelar esos puntos iniciales. La función de probabilidad *ad hoc* en este caso será proporcional al radio de curvatura. El radio de curvatura es una magnitud que mide la curvatura de un objeto geométrico tal como una línea curva, una superficie o más en general una variedad diferenciable embebida en un espacio euclídeo. Intuitivamente, la curvatura es una medida de la tasa instantánea de cambio de dirección de un punto que se mueve en la curva: cuanto mayor es la curvatura, mayor es esta tasa de cambio. Para una curva, la tasa de cambio es igual al radio del arco circular que mejor aproxima la curva en ese punto y es el recíproco de la curvatura.

$$\kappa = \frac{1}{r} \quad (23)$$

La curvatura de una curva diferenciable puede definirse a través del círculo osculador, que es el círculo que mejor se aproxima a la curva en un punto, tal como lo esquematiza la Figura 3.

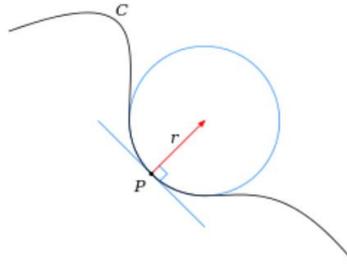


Fig. 3. Ilustración de radio de curvatura para un punto “P”.

En caso que pueda escribirse $y = f(x)$, de tal modo que para cada punto de la curva exista un único valor de x , como ocurre con el caso aquí estudiado, el radio de curvatura se puede calcular simplemente como:

$$r = \frac{\left[1 + \left(\frac{df}{dx}\right)^2\right]^{\frac{3}{2}}}{\left|\frac{d^2f}{dx^2}\right|} \quad (24)$$

Y la curvatura es, por lo tanto:

$$\kappa = \frac{\left|\frac{d^2f}{dx^2}\right|}{\left[1 + \left(\frac{df}{dx}\right)^2\right]^{\frac{3}{2}}} \quad (25)$$

Por lo tanto, se generó una función de probabilidad *ad hoc* utilizando la expresión (25), la cual permite asignar mayores probabilidades a los puntos en el entorno del codo de la curva (donde la curvatura es máxima, y el radio es mínimo). Mientras que, en el *plateau* de la curva, el radio de curvatura es tendiente a infinito, y en consecuencia, la curvatura de esa parte es tendiente a cero (nótese que en una recta la curvatura es cero o nula).

3 Experimentos y resultados

A continuación, se presentan los datos utilizados para desarrollar los experimentos, y luego, los resultados obtenidos.

3.1 Datos y configuración de experimentos

Para evaluar el nuevo modelo de aproximación lineal basado en las rectas obtenidas por radio de curvatura como puntos iniciales se procedió a resolver una serie de instancias de problema de scheduling, considerando distinto número de máquinas y trabajos. A saber, 3, 4 y 8 trabajos, y 5, 10 y 15 máquinas, considerando las siguientes combinaciones ($n \times m$): 3x5; 3x10; 5x5; 5x10; 8x15. Los tiempos de procesamiento estándar se obtuvieron utilizando una distribución uniforme [1;99], y un valor de α y β de 0,2 y 0,8, respectivamente. Para este trabajo se consideraron 3 rectas para aproximar la curva exponencial de los tiempos de procesamientos reales (considerando efecto aprendizaje). Se generaron 5 conjuntos distintos de datos para cada tamaño de problema.

Para resolver el modelo MILP (el cual considera la aproximación lineal) se utilizó como solver CPLEX y se modeló en Pyomo. Mientras que el modelo MINLP, se resolvió utilizando DICOPT como solver, y como lenguaje de programación GAMS.

3.2 Resultados

Los resultados promedios obtenidos de los 5 conjuntos de datos para cada uno de las instancias del problema estudiado se presentan en la Tabla 1. En esta tabla se presentan en la columna " C_{max} promedio MILP" los valores obtenidos por el método propuesto aquí, es decir el método de aproximación lineal considerando las rectas obtenidas utilizando como puntos iniciales los definidos por el método propuesto de curvatura de la curva; y en la columna " C_{max} promedio MINLP" los valores obtenidos por DICOPT.

Tabla 1. Resultados de los experimentos de optimización obtenidos por el método de aproximación lineal y por el modelo MINLP original, columnas " C_{max} promedio MILP" y " C_{max} promedio MINLP", respectivamente.

Instancias ($n \times m$)	C_{max} promedio MILP	C_{max} promedio MINLP
3x5	346	346
3x10	612	612
5x5	425	425
5x10	704	704
8x15	1062	1062

De la Tabla 1 se observa que en todos los casos los valores de ambas columnas coinciden. Esto permite asegurar que ambos enfoques convergen al óptimo del problema, ya que al tratarse de un problema MINLP convexo DICOPT encuentra el óptimo del problema (se configuró la corrida para que así sea). Por lo tanto, el método de aproximación propuesto, para los casos evaluados, también encontró la solución óptima.

Tabla 2. Tamaño de los problemas MINLP resueltos en términos de variables, restricciones y tiempos de CPU.

Instancias ($n \times m$)	Nro. Var. Disc.	Nro. Var. Con.	Nro. Restricciones	T CPU
3x5	6	37	81	< 1 seg
3x10	6	67	156	< 1 seg
5x5	20	71	195	< 1 seg
5x10	20	121	370	0.1 seg
8x15	56	297	1256	2 seg

En las Tablas 2 y 3 se presentan los tamaños de los modelos en términos de variables discretas, continuas y restricciones, junto con los tiempos de CPU utilizados. La table 2 recopila los datos de los modelo MINLP, y la Tabla 3 de los modelos MILP con el ajuste de nuevas ecuaciones. Al comparar ambas tablas, se observa que no hay una diferencia sustancial entre ambos modelos, ya que no se definen nuevas variables al pasar al modelo MILP sino que se agregan más restricciones. En cuanto a los tiempos de CPU, se observa que corren en tiempos similares.

Tabla 3. Tamaño de los problemas MILP resueltos en términos de variables, restricciones y tiempos de CPU.

Instancias ($n \times m$)	Nro. Var. Disc.	Nro. Var. Con.	Nro. Restricciones	T CPU
3x5	6	37	84	1 seg
3x10	6	67	189	1 seg
5x5	20	71	225	1 seg
5x10	20	121	425	1 seg
8x15	56	297	1384	1 seg

4 Conclusiones

En este trabajo se estudió un enfoque de modelado por aproximación lineal para un problema Mixto entero no lineal de scheduling en configuraciones flow shop. Puntualmente, el efecto aprendizaje considerado es del tipo past-sequence dependent. El método propuesto permite abordar el problema de manera apropiada, generando soluciones de buena calidad.

Como futura línea de trabajo, siguiendo la motivación central del trabajo de desarrollar métodos de resolución que permitan resolver este tipo de problemas con la menor infraestructura de cálculo posible, tanto en recursos de software como en recursos humanos, se propone avanzar en el abordaje de problemas de mayor tamaño y para ello desarrollar de métodos heurísticos que puedan contemplar consideraciones particulares de la curva de aprendizaje. En este sentido resultan interesante explorar enfoques que puedan considerar el resultado de la literatura para el caso *single-machine* en donde la regla de despacho SPT (Shortest processing time) asegura el óptimo. En consecuencia,

resultaría interesante analizar posibles extensiones y alternativas de esa regla de despacho para el caso con más de una máquina como el flow shop aquí considerado.

Referencias

1. Biskup, D. (1999). Single-machine scheduling with learning considerations. *European Journal of Operational Research*, 115(1), 173-178.
2. Wright, T. P. (1936). Factors affecting the cost of airplanes. *Journal of the aeronautical sciences*, 3(4), 122-128.
3. Yelle, L. E. (1979). The learning curve: Historical review and comprehensive survey. *Decision Sciences*, 10(2), 302-328.
4. Biskup, D. (2008). A state-of-the-art review on scheduling with learning effects. *European Journal of Operational Research*, 188(2), 315-329.
5. Mosheiov, G. (2001). Scheduling problems with a learning effect. *European Journal of Operational Research*, 132(3), 687-693.
6. Lee, W. C., & Wu, C. C. (2004). Minimizing total completion time in a two-machine flowshop with a learning effect. *International Journal of Production Economics*, 88(1), 85-93.
7. Wang, J. B., & Xia, Z. Q. (2005). Flow-shop scheduling with a learning effect. *Journal of the Operational Research Society*, 56(11), 1325-1330.
8. 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.
9. Koulamas, C., & Kyparisis, G. J. (2008). Single-machine scheduling problems with past-sequence-dependent setup times. *European Journal of Operational Research*, 187(3), 1045-1049.
10. Wang, J. B. (2008). Single-machine scheduling with past-sequence-dependent setup times and time-dependent learning effect. *Computers & Industrial Engineering*, 55(3), 584-591.
11. Wang, J. B., Wang, D., Wang, L. Y., Lin, L., Yin, N., & Wang, W. W. (2009). Single machine scheduling with exponential time-dependent learning effect and past-sequence-dependent setup times. *Computers & Mathematics with Applications*, 57(1), 9-16.
12. Ji, M., & Cheng, T. E. (2010). Scheduling with job-dependent learning effects and multiple rate-modifying activities. *Information Processing Letters*, 110(11), 460-463.
13. 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.
14. Wang, J. B., & Wang, J. J. (2014). Flowshop scheduling with a general exponential learning effect. *Computers & Operations Research*, 43, 292-308.
15. Soroush, H. M. (2015). Scheduling with job-dependent past-sequence-dependent setup times and job-dependent position-based learning effects on a single processor. *European Journal of Industrial Engineering*, 9(3), 277-307.
16. Ferraro, A., Rossit, D. A., & Toncovich, A. (2023). Flow shop scheduling problem with non-linear learning effects: A linear approximation scheme for non-technical users. *Journal of Computational and Applied Mathematics*, 424, 114983.