

Capítulo 7. Secuenciación de trabajos en una línea de SMT con enfoque multiobjetivo

Juan Pablo Horn Gallardo*
Elías Olivares Benítez**
Omar Rojas***

Introducción

La industria electrónica mexicana se ha ido consolidando hasta convertirse en uno de los sectores primordiales del país. México es uno de los principales países exportadores y ensambladores del mundo; en él se ubican nueve de las diez principales empresas transnacionales de servicio de manufactura, por lo que México se ha posicionado como un importante destino de inversión.

* Universidad Panamericana, Facultad de Ingeniería, 0170875@up.edu.mx

** Universidad Panamericana, Facultad de Ingeniería, colivaresb@up.edu.mx

*** Universidad Panamericana, Facultad de Ingeniería, orojas@up.edu.mx

En general, los productos electrónicos están formados por Ensamblajes de Circuitos Impresos, también conocidos como PCA (Printed Circuit Assemblies, en inglés), componentes mecánicos (pueden ser metálicos y/o plásticos, como tapas, soportes, teclados, marcos de pantalla, bases, chasis, entre otros), material impreso (como etiquetas y manuales de servicio) y material de empaque. Los PCA están conformados por componentes electrónicos, *software* embebido, circuitos impresos y componentes eléctricos.

La empresa analizada pertenece al mercado de la electrónica, pero está en el segmento de medio-bajo volumen, alta complejidad y mezcla, y es parte de las Electronics Manufacturing Services (EMS). EMS es un término utilizado para describir a compañías que diseñan, producen, prueban, distribuyen y ofrecen servicio postventa. Todo esto lo hacen para las Original Equipment Manufacturer (OEM). Los sectores en los cuales se desarrolla la empresa son los siguientes:

- Salud y ciencias de la vida.
- Industrial y comercial.
- Comunicaciones.
- Aeroespacial y defensa.

La empresa usa procesos de Tecnología de Montaje Superficial (SMT, por las siglas en inglés de Surface Mount Technology). La empresa maneja un modelo Make to Order, lo cual significa que no comienza a producir un producto hasta que el cliente haya hecho un pedido de un ensamble (o una mezcla de ensamblajes) en específico. El cliente envía por adelantado su demanda por medio de un pronóstico, el cual elabora para un periodo determinado, para que así la empresa pueda llevar a cabo la compra o la colocación de órdenes para los materiales necesarios para su producción.

Ya que la manera de trabajar de la empresa es por medio de un sistema Make to Order, el proceso de producción comienza después de recibir la orden de compra del cliente. Sabiendo los requerimientos del cliente, el planeador de la producción analiza que el producto no presente escasez de materiales. Si se tienen todos los materiales necesarios para llevar a cabo la orden se considera que la orden está Clear To Build (CTB) y se le asigna un espacio dentro del plan de la producción (al igual de capacidad) de la planta y se le notifica al cliente una fecha de compromiso para dicho pedido. De otra manera, en caso de que

el pedido del cliente presente escasez, se le notifica al cliente el estado del material y se programa una fecha tentativa de entrega en función de la llegada del material. Para cuestiones de practicidad, sólo se analizaron productos los cuales se encontrarán CTB, ya que si no se cumplió el compromiso del cliente porque algún material no llegó cuando se tenía contemplado o este presentó una escasez en el mercado, estas situaciones quedan fuera del alcance de este trabajo. El estudio se concentró principalmente en el sector industrial comercial de la empresa.

Hoy en día la empresa presenta problemas cumpliendo con el On Time Delivery (OTD). El OTD es un métrico que mide la relación de trabajos comprometidos con el cliente y trabajos entregados a tiempo (en su totalidad, lo que implica que sólo se considera que una entrega está completa hasta que todas las unidades del pedido son entregadas al cliente). La empresa tiene como objetivo lograr un OTD de 80% o mayor para todos los compromisos que haga con el cliente. Esta medición se hace semana con semana. Ocasionalmente la empresa tiene problemas para llegar a dicha meta; estos problemas son causados por malas prácticas en la planeación de la producción, lo cual se convierte en un efecto dominó semana con semana. Al no poder completar un pedido una semana, se pasa a la siguiente semana el remanente de la orden y claramente esto les quita tiempo a los compromisos de la semana en curso.

El OTD es un métrico sumamente importante para cualquier empresa manufacturera, ya que este simboliza la imagen que una empresa muestra al cliente. Si se tiene un OTD muy bajo significa que la empresa está constantemente fallando los compromisos hechos con nuestros clientes, lo cual podría llegar a causar que se terminen las relaciones existentes. De igual manera, si constantemente se incumplen los compromisos del cliente, esto tiene repercusiones económicas negativas sobre la empresa. Como ya mencionamos, existe la posibilidad de perder a los clientes, pero no sólo eso, ya que los costos de inventarios suelen afectar financieramente a la empresa por cada día que se tengan parados tanto los productos terminados como la misma materia prima de las ordenes incompletas.

Un factor muy importante que se tiene que tomar en cuenta, ya que limita el proceso de producción, es la capacidad instalada de la planta. Esto es un problema porque hay veces que los proyectos (orden-producto por cliente) terminan compartiendo líneas de producción entre sí, lo cual complica un poco la secuenciación de los trabajos dentro de las líneas. Lo que un planeador acabe

metiendo al plan de producción tendrá que estar en función de la asignación del tiempo dentro del turno. La empresa trabaja tres turnos de lunes a sábado.

El trabajo se conformará de la siguiente manera: en la siguiente sección de describirá la revisión de la literatura; se hablará un poco de la historia de este tipo de problemas al igual de como otros autores han abordado en problema para escenarios similares al que se estudia en este trabajo. Después, en la sección de métodos, se describirán los modelos matemáticos que representan el problema y se explicará cómo se llevó a cabo el trabajo, incluyendo los datos utilizados para resolverlos. Posteriormente se mostrarán los resultados y su discusión, y finalmente la sección de conclusiones.

Fundamentación teórica

Desde la revolución industrial, la manera en que la humanidad ha desarrollado la manufactura ha crecido de manera exponencial. Desde la creación de la línea de producción por parte de Henry Ford para la producción del modelo T en el año 1908, hasta la industria 4.0. Hoy en día las técnicas de manufactura son tan sofisticadas que es posible saber con exactitud los tiempos de procesamiento que un producto requiere en cierto proceso o si es que estos siguen una distribución de probabilidad. Esto a su vez, hace posible hacer cálculos matemáticos para saber cuál secuencia de trabajos resulta ser la mejor dependiendo del métrico que se desee medir. A este tipo de modelos se les denomina modelos de programación de líneas de flujo. Estos modelos han ido evolucionando desde su primera introducción por Johnson (1954), en el cual analizó el comportamiento de una línea de producción con dos máquinas utilizando diferentes objetivos o métodos de solución para encontrar sus objetivos. A continuación, se presentará algunos de los métodos que se han ido creando para la solución de los problemas de programación de línea.

Los problemas clásicos de programación de líneas de flujo existentes en la literatura al día de hoy se concentran principalmente en generar una secuencia de trabajos en un ambiente determinístico y estático (Liu, Wang *et al.*, 2017). El problema de programación de líneas de flujo y sus variaciones han sido resueltas mediante un gran número de métodos. De igual manera, este problema tiene una gran variedad de objetivos los cuales con las correctas adaptaciones se pueden llegar a cumplir. A continuación, se explicarán principalmente los

siguientes objetivos: minimización del tiempo total de procesamiento, minimización de la tardanza máxima y minimización del número de los trabajos tardíos. Y algunas otras variaciones las cuales han aplicado otros autores anteriormente.

El primer objetivo es el de la minimización del tiempo máximo de procesamiento. Este objetivo consiste en hacer que el número total de trabajos haga el menor tiempo posible en ser procesados en todas las máquinas. Por eso se define al tiempo máximo de procesamiento como el tiempo de terminación del último trabajo en la última máquina. Johnson (1954) resolvió un problema de n trabajos y dos máquinas para minimizar el tiempo máximo de procesamiento. Este trabajo es considerado el más importante para los problemas de programación de líneas de flujo, ya que al pasar los años se ha adaptado para resolver problemas de n trabajos y m máquinas, tal como se aplicó en este trabajo. Lin y Ying (2016) propusieron dos metaheurísticas para optimizar el tiempo máximo de procesamiento para problemas de programación de línea que involucraban un esquema sin esperas. Nagano y Araújo (2013) abordan el problema de secuenciación de trabajos sin espera y con tiempos de procesamiento dependientes a la secuencia para minimizar el tiempo máximo de procesamiento. Campbell *et al.* (1970) introdujeron lo que ahora se conoce como el algoritmo CDS (Campbell, Dudek, Smith), el cual es una de las primeras heurísticas creadas que minimizaran el tiempo máximo de procesamiento. Nawaz *et al.* (1983) proponen una heurística la cual se llama NEH (Nawaz, Enscore, Ham) que utiliza el algoritmo más eficiente para la minimización del tiempo máximo de procesamiento, según Ruiz y Maroto (2005). Lee *et al.* (2019) propusieron un método que aplicaba un algoritmo voraz para la minimización del tiempo máximo de procesamiento en un enfoque de programación en línea. Soewanda *et al.* (2007) abordaron el problema con algoritmo genético híbrido para minimizar el tiempo máximo de procesamiento. Framinan *et al.* (2019) analizaron un sistema multietapas de producción en el cual podría haber una reorganización de los trabajos en cualquier punto de la producción los cuales tienen un tiempo de procesamiento variable. Claramente, al alterar la secuencia de los trabajos se altera el tiempo máximo de procesamiento. Liu, Jin y Price (2017) abarcan el problema de reordenar nuevas órdenes entrantes. Han *et al.* (2011) aplicaron un Discrete Artificial Bee Colony (DABC) y un Blocking Flow Shop (BFS). Cheng *et al.* (2014) consideraron trabajos dependientes del tiempo los cuales se deterioraban.

El segundo objetivo es el de la minimización de la máxima tardanza. Este objetivo se encarga de que el trabajo que llega más tarde en la secuencia minimice su tardanza. Esto conlleva que en cierto modo el algoritmo actúe similar al del tiempo máximo de procesamiento. Chang *et al.* (2007) definieron un Mining Gene on Sub-Population Genetic Algorithm (MGSPGA) para compararlo con un Sub-Population Genetic Algorithm (SPGA), el cual tenía como objetivo ambos: el tiempo máximo de procesamiento y la tardanza máxima. Hamdi y Loukil (2014) abordaron el problema de permutación con ambos, el mínimo y el máximo retraso utilizando el criterio de la tardanza total. Li *et al.* (2018) estudiaron un problema de programación de líneas de flujo que se concentraba en las fechas de entrega, el cual tenía como objetivo minimizar ambos el costo de las entregas tardías como de las entregas que se entregaran antes del tiempo acordado. Este es un caso interesante ya que muchas veces en la industria por más que la empresa manufacturera pueda llegar a tener listos los requerimientos del cliente antes de la fecha programada, es posible que el cliente no pueda recibir el pedido antes de la fecha acordada, lo cual incurre un costo para la empresa manufacturera. Cheng (1991) consideró un problema con una máquina para la asignación de días de entrega con tiempos de procesamiento estocásticos, cuyo objetivo era minimizar el costo de las penalizaciones por entregar antes o después los pedidos. Li *et al.* (2010) y Li *et al.* (2011) propusieron modelos con un ambiente difuso, en el cual los tiempos de procesamiento eran presentados como variables difusas con el objetivo de minimizar el costo de las llegadas prontas y tardías de los trabajos. Liu *et al.* (2013) desarrollaron seis diferentes heurísticas para resolver un problema de programación de líneas de flujo sin esperas, utilizando la tardanza total como criterio al igual que el algoritmo NEH. Kalman *et al.* (2019) estudiaron un problema de secuenciación en una fábrica de muebles que buscaba minimizar el tiempo máximo de flujo y la tardanza, usando un algoritmo genético y un modelo de programación matemática.

Finalmente, nuestro tercer último objetivo es el de minimizar el número de trabajos tardíos. Este objetivo hace uso de las fechas de entrega de los trabajos para así realizar la secuenciación de tal forma que el número de trabajos que no cumplen con su fecha de entrega sea menor. Dhouib *et al.* (2013) consideraron un algoritmo de recocido simulado para minimizar ambos el tiempo máximo de procesamiento y el número de trabajos tardíos. Lee *et al.* (1991) realizaron un estudio donde trató de minimizar el número de trabajos tardíos y la ponderación de las llegadas tempranas y tardías de trabajos con fechas de entrega iguales.

Todos los objetivos mencionados anteriormente pueden ser utilizados dentro del modelo de Sawik (2011), pero en este trabajo se decidió abordar el dilema con un problema multiobjetivo que engloba los tres objetivos (tiempo de flujo máximo, tardanza máxima y número de trabajos tardíos).

Murata *et al.* (1996) y Toktas *et al.* (2004) consideran que, si bien los problemas monoobjetivo son muy útiles, los problemas multiobjetivo son más realistas. Daniels y Chambers (1990), Marichelvam y Geetha (2014), Rahimi-Vahed *et al.* (2008) y Murata *et al.* (1996) también opinan que los problemas multiobjetivo se adaptan mejor a los problemas de la industria actual. Normalmente los objetivos de problemas multiobjetivo son conflictivos entre sí, esto quiere decir que es difícil que todos los objetivos se cumplan (Murata *et al.*, 1996). En otras palabras, son mutuamente excluyentes. Según Rahimi-Vahed *et al.* (2008), la forma general de una función multiobjetivo de minimización con p variables de decisión y q objetivos se puede escribir de la siguiente manera:

$$\text{Minimizar } y=f(x)=(f_1(x), f_2(x), \dots, f_q(x))$$

donde $x \in R^p$, $y \in R^q$.

Collette y Siarry (2004) clasificaron los métodos multiobjetivo de la siguiente manera:

1. Métodos Borrosos.
2. Métodos de Ayuda a la Decisión.
3. Métodos Escalares.
4. Métodos Interactivos
5. Métodos que Utilizan Metaheurísticas.

Danneberg *et al.* (1999) atacaron el problema de programación de líneas de flujo biobjetivo con esperas con tiempos de preparación y trabajos particionados en grupos o familias. Propusieron el tiempo máximo de procesamiento al igual que la suma ponderada de los tiempos de terminación de los trabajos. Ponnambalam *et al.* (2004) propusieron un algoritmo del agente viajero de la mano de un algoritmo genético para optimizar la suma ponderada del tiempo máximo de procesamiento, el tiempo promedio de flujo y los tiempos ociosos de las máquinas. Loukil *et al.* (2005) propusieron un algoritmo recocido

simulado multiobjetivo para optimizar la producción en varios escenarios; estos escenarios eran escenarios con problemas una máquina, con máquinas paralelas y con problemas de permutaciones. También propusieron varias funciones objetivo, las cuales fueron: tiempo de terminación promedio ponderado, la tardanza promedio ponderada, la precocidad promedio ponderada, tiempo máximo de procesamiento, máxima tardanza, máxima precocidad y el número de los trabajos tardíos. Murata *et al.* (1996) abordaron un problema con tres objetivos: minimizar el tiempo máximo de procesamiento, minimizar la tardanza total y minimizar el tiempo total de flujo. Chang *et al.* (2007) estudiaron el caso de algoritmos genéticos de subpoblaciones. Propusieron varios métodos que pretendían mejorar el proceso de minar los cromosomas. Zelazny y Pempera (2015) propusieron una búsqueda tabú paralela para un problema biobjetivo de secuenciación. También menciona que los problemas multiobjetivo son principalmente resueltos por métodos evolutivos. Marichelvam y Geetha (2014) propusieron un algoritmo de luciérnaga discreto triobjetivo el cual minimizaba los siguientes objetivos: tiempo máximo de procesamiento, tiempo máximo de flujo y el tiempo ocioso de las máquinas dentro de un esquema de multietapas. Jungwattanakit *et al.* (2009) formularon un algoritmo entero mixto para resolver problemas de programación de líneas de flujo flexibles. Su objetivo fue minimizar el tiempo máximo de procesamiento y la suma de las tardanzas y llegadas prontas. Behnamian y Fatemi Ghomi (2011) aplicaron un algoritmo genético híbrido para minimizar el tiempo máximo de procesamiento y la asignación total de los recursos. Daniels y Chambers (1990) realizaron un algoritmo que se concentraba en optimizar la máxima tardanza y el tiempo máximo de procesamiento. Guinet y Solomon (1996) compararon varias heurísticas para ver cuál tenía un mejor resultado minimizando la máxima tardanza o el tiempo máximo de procesamiento.

Metodología

El problema se planea resolver utilizando una adaptación del modelo de programación de líneas de flujo de Sawik (2011). Este modelo tiene como función objetivo minimizar los siguientes propósitos: el tiempo total de procesamiento, la tardanza máxima y el número de trabajos tardíos. Se estará analizando el comportamiento del flujo de trabajos en una línea de producción durante una

semana, la cual tuvo un desempeño debajo de 80%. Esto, con el fin de lograr un modelo que pueda llevar a cabo la planeación de los trabajos que pasen por esa línea. Se compararán los resultados obtenidos con el modelo matemático a los resultados reales de dicha semana. Teniendo éxito con lo anterior, se replicará el procedimiento en las demás líneas de producción. Ya que contando con un solo modelo para toda la planta, complica la solución del problema bastante, y dividiendo el problema en instancias por línea facilita su solución. La dificultad de resolución de los problemas de programación de línea de flujo es del orden: $(n!)^m$, donde n representa el número de trabajos a secuenciar y m representa el número de máquinas o procesos por los que tienen que pasar los trabajos. Por lo mismo, el número de productos y familias por analizar sigue por definirse, ya que aún no se sabe cuál sería un número conveniente de productos y/o familias (que se procesen por la misma línea) para analizar.

Utilizando el ejemplo de la semana debajo de 80% de OTD, se analizará si es posible minimizar el número de trabajos tardíos, dándole preferencia a los remanentes de semanas anteriores, y tratar que la inserción de estos trabajos no perjudique el desempeño del flujo de los trabajos de la semana actual.

En este trabajo se tuvieron las siguientes consideraciones:

1. El número de máquinas es conocido y fijo.
2. El número de trabajos y sus tiempos de procesamiento en cada máquina es conocido.
3. Todos los trabajos están disponibles en el tiempo igual a cero.
4. Los tiempos de preparación están incluidos dentro de los tiempos de procesamiento y son independientes de la secuencia.
5. Cada máquina puede procesar sólo un trabajo a la vez.
6. Algunos trabajos no pasan por alguna de las máquinas. Para estos trabajos el tiempo de procesamiento en dichas máquinas es igual a cero.

El modelo utilizado en este trabajo fue evolucionando conforme a los resultados que fue presentando para la instancia utilizada. Por lo que la función objetivo del modelo al igual que algunas restricciones fueron cambiando con el tiempo. Al principio se quería optimizar sólo el tiempo máximo de procesamiento y la máxima tardanza. Después, al ver que los resultados obtenidos no eran tan buenos se decidió por trabajar con una función objetivo la cual minimizaba el tiempo máximo de procesamiento y la sumatoria de

las tardanzas de los trabajos. De nuevo parecía que los resultados obtenidos con este segundo modelo tampoco eran tan buenos, por lo que se llegó al último modelo triobjetivo que trata de minimizar los siguientes tres objetivos: el tiempo máximo de procesamiento, la máxima tardanza y el número de trabajos tardíos. A continuación, se presentan las tres versiones del modelo para mostrar cómo fue evolucionando el modelo. Los modelos se implementaron en lenguaje de optimización AMPL, con motor de optimización GUROBI, y para la instancia estudiada se resolvieron en tiempos menores a una hora, considerando la construcción completa de los frentes de Pareto, en una computadora con procesador Intel Core i7-1165G7 @ 2.80GHz, 32 Gb RAM.

Modelo 1

Este modelo tiene como función objetivo minimizar el flujo máximo de los trabajos (el tiempo total de procesamiento) y la tardanza máxima de todos los trabajos. El modelo se presenta de la siguiente manera:

Índices

i = Proceso, $i \in I = \{1, \dots, m\}$

j = Trabajo en proceso $j \in J_i = \{1, \dots, n\}$

Parámetros

m = Número de procesos

n = Número de trabajos

p_{ik} = Tiempo de procesamiento del trabajo k en el proceso i

Q = Coeficiente positivo grande (Big M)

D_j = Fecha de entrega del trabajo j

α = ponderación para la métrica

Factor = Número positivo que sirve para que las métricas se encuentren en la misma magnitud

Variables de decisión

c_{ik} = Tiempo de finalización del trabajo k en el proceso i

y_{kl} = 1, si el trabajo k precede al trabajo l en la secuencia de trabajos; o en otro caso

C_{MAX} = Máximo tiempo de flujo de cualquier trabajo en la última máquina.

L_{MAX} = Máximo retardo de cualquier trabajo.

La función objetivo del modelo y sus restricciones se presentan a continuación:

$$\text{Minimizar } z = \text{Factor} \cdot \alpha \cdot C_{MAX} + (1-\alpha) \cdot L_{MAX} \quad (1)$$

Sujeto a:

$$c_{ik} \geq p_{ik}, \quad k \in K, \quad (2)$$

$$c_{ik} - c_{i-1k} \geq p_{ik}, \quad i \in I, k \in K: i > 1, \quad (3)$$

$$c_{ik} + Qy_{kl} \geq c_{il} + p_{ik}, \quad i \in I; k, l \in K: i < l, \quad (4)$$

$$c_{il} + Q(1 - y_{kl}) \geq c_{ik} + p_{il}, \quad i \in I; k, l \in K: i < l, \quad (5)$$

$$c_{ik} \leq C_{MAX}, \quad i \in I, k \in K, \quad (6)$$

$$c_{ik} - D_j \leq L_{MAX}, \quad i \in I, k \in K, \quad (7)$$

$$y_{kl} + y_{lk} = 1, \quad \forall k \neq l, \quad (8)$$

$$y_{kl} = 0, \quad \forall k = l, \quad (9)$$

$$C_{MAX} \geq 0, \quad (10)$$

$$L_{MAX} \in \mathbb{R}, \quad (11)$$

$$c_{ik} \geq 0, \quad i \in I, k \in K, \quad (12)$$

$$y_{kl} \in \{1, 0\}, \quad k, l \in K: k < l. \quad (13)$$

La función objetivo (1) expresa la minimización del tiempo máximo de flujo (el tiempo total de procesamiento) y de la tardanza máxima de los trabajos. Las restricciones (2) y (3) nos indican que todos los trabajos tienen que pasar por todas las máquinas. Las restricciones (4) y (5) previenen que dos trabajos no pueden ser procesados simultáneamente en la misma máquina. La restricción (6) nos muestra que la duración total de la secuencia será el tiempo de flujo del último trabajo en la última máquina. La restricción (7) permite calcular la tardanza máxima que presenten los trabajos. La restricción (8) asegura que los trabajos aparezcan sólo una vez en la secuencia. La restricción (9) da valor de cero para las variables que presenten valores iguales de k y l . Las restricciones (10), (11) (12) y (13) indican la naturaleza de las variables.

Modelo 2

Este modelo tiene como función objetivo minimizar el flujo máximo de los trabajos (el tiempo total de procesamiento) y la sumatoria de las tardanzas de todos los trabajos. El modelo se ve de la siguiente manera:

Índices

i = Proceso, $i \in I = \{1, \dots, m\}$

j = Trabajo en proceso $j \in J_i = \{1, \dots, n\}$

Parámetros

m = Número de procesos

n = Número de trabajos

p_{ik} = Tiempo de procesamiento del trabajo k en el proceso i

Q = Coeficiente positivo grande (Big M)

D_j = Fecha de entrega del trabajo j

α = ponderación para la métrica

Factor = Número positivo que sirve para que las métricas se encuentren en la misma magnitud

Variables de decisión

c_{ik} = Tiempo de finalización del trabajo k en el proceso i

L_k = Tardanza del trabajo k

$y_{kl} = 1$, si el trabajo k precede al trabajo l en la secuencia de trabajos; o en otro caso.

La función objetivo del modelo y sus restricciones se presentan a continuación:

$$\text{Minimizar } z = \alpha \cdot C_{MAX} + \text{Factor} \cdot (1 - \alpha) \sum_{k=1}^n L_k \quad (14)$$

Sujeto a:

$$c_{ik} \geq p_{ik}, \quad k \in K, \quad (15)$$

$$c_{ik} - c_{i-1k} \geq p_{ik}, \quad i \in I, k \in K: i > 1, \quad (16)$$

$$c_{ik} + Qy_{kl} \geq c_{il} + p_{ik}, \quad i \in I; k, l \in K: i < l, \quad (17)$$

$$c_{il} + Q(1 - y_{kl}) \geq c_{ik} + p_{il}, \quad i \in I; k, l \in K: i < l, \quad (18)$$

$$c_{ik} \leq CMAX, \quad i \in I, k \in K, \quad (19)$$

$$c_{ik} - D_j \leq L_k, \quad i \in I, k \in K \quad (20)$$

$$y_{kl} + y_{lk} = 1, \quad \forall k \neq l, \quad (21)$$

$$y_{kk} = 0, \quad \forall k = l, \quad (22)$$

$$CMAX \geq 0, \quad (23)$$

$$L_k \in \mathbb{R}, \quad k \in K, \quad (24)$$

$$c_{ik} \geq 0, \quad i \in I, k \in K, \quad (25)$$

$$y_{kl} \in \{1, 0\}, \quad k, l \in K: k < l. \quad (26)$$

La función objetivo (14) expresa la minimización del tiempo máximo de flujo (el tiempo total de procesamiento) y de la sumatoria de las tardanzas de los trabajos. Las restricciones (15) y (16) nos indican que todos los trabajos tienen que pasar por todas las máquinas. Las restricciones (17) y (18) previenen que dos trabajos no pueden ser procesados simultáneamente en la misma máquina. La restricción (19) nos muestra que la duración total de la secuencia será el tiempo de flujo del último trabajo en la última máquina. La restricción (20) nos muestra las tardanzas de todos los trabajos. La restricción (21) asegura que los trabajos aparezcan sólo una vez en la secuencia. La restricción (22) da valor de cero para las variables que presenten valores iguales de k y l . Las restricciones (23), (24), (25) y (26) indican la naturaleza de las variables.

Modelo 3

Este modelo tiene como función objetivo minimizar el flujo máximo de los trabajos (el tiempo total de procesamiento), la tardanza máxima de todos los trabajos y el número de trabajos tardíos. El modelo se presenta de la siguiente manera:

Índices

$i =$ Proceso, $i \in I = \{1, \dots, m\}$

$j =$ Trabajo en proceso $j \in J_i = \{1, \dots, n\}$

Parámetros

$m =$ Número de procesos

$n =$ Número de trabajos

p_{ik} = Tiempo de procesamiento del trabajo k en el proceso i
 Q = Coeficiente positivo grande (Big M)
 Q_2 = Coeficiente positivo grande (Big M)
 D_j = Fecha de entrega del trabajo j
 α_r = ponderación para la métrica
 $Factor_r$ = Número positivo que sirve para que las métricas r , se encuentren en la misma magnitud

Variables de decisión

c_{ik} = Tiempo de finalización del trabajo k en el proceso i
 L_k = Tardanza del trabajo k
 $y_{kl} = 1$, si el trabajo k precede al trabajo l en la secuencia de trabajos; o en otro caso
 $N_k = 1$, si el trabajo k tiene una tardanza positiva; o en otro caso
 C_{MAX} = Máximo tiempo de Flujo de cualquier trabajo en la última máquina.
 L_{MAX} = Máximo retardo de cualquier trabajo.

La función objetivo del modelo y sus restricciones se presentan a continuación:

$$\text{Minimizar } z = \frac{\alpha_1 \cdot C_{MAX}}{Factor_1} + \frac{\alpha_2 \cdot L_{MAX}}{Factor_2} + \frac{\alpha_3 \cdot \sum_{k=1}^n N_k}{Factor_3} \quad (27)$$

Sujeto a:

$$c_{ik} \geq p_{ik}, \quad k \in K, \quad (28)$$

$$c_{ik} - c_{i-1k} \geq p_{ik}, \quad i \in I, k \in K: i > 1, \quad (29)$$

$$c_{ik} + Qy_{kl} \geq c_{il} + p_{ik}, \quad i \in I; k, l \in K: i < l, \quad (30)$$

$$c_{il} + Q(1 - y_{kl}) \geq c_{ik} + p_{il}, \quad i \in I; k, l \in K: i < l, \quad (31)$$

$$c_{ik} \leq C_{MAX}, \quad i \in I, k \in K, \quad (32)$$

$$c_{ik} - D_j \leq L_{MAX}, \quad i \in I, k \in K, \quad (33)$$

$$L_k \leq Q_2 * N_k, \quad k \in K, \quad (34)$$

$$y_{kl} + y_{lk} = 1, \quad \forall k \neq l, \quad (35)$$

$$y_{kl} = 0, \quad \forall k = l, \quad (36)$$

$$C_{MAX} \geq 0, \quad (37)$$

$$L_{MAX} \in R, \quad (38)$$

$$c_{ik} \geq 0, \quad i \in I, k \in K, \quad (39)$$

$$y_{kl} \in \{1, 0\}, \quad k, l \in K: k < l, \quad (40)$$

$$L_k \in \mathbb{R}, \quad k \in K, \quad (41)$$

$$N_k \in \{1, 0\}, \quad k \in K. \quad (42)$$

La función objetivo (27) expresa la minimización de la sumatoria del tiempo máximo de flujo (el tiempo total de procesamiento), la tardanza máxima de todos los trabajos y el número de trabajos tardíos. Las restricciones (28) y (29) nos indican que todos los trabajos tienen que pasar por todas las máquinas. Las restricciones (30) y (31) previenen que dos trabajos no pueden ser procesados simultáneamente en la misma máquina. La restricción (32) nos muestra que la duración total de la secuencia será el tiempo de flujo del último trabajo en la última máquina. La restricción (33) nos muestra las tardanzas de todos los trabajos. La restricción (34) activa un indicador binario si el trabajo presenta tardanza positiva. La restricción (35) asegura que los trabajos aparezcan sólo una vez en la secuencia. La restricción (36) da valor de cero para las variables que presenten valores iguales de k y l . Las restricciones (37), (38), (39), (40), (41) y (42) indican la naturaleza de las variables.

Instancia

El estudio se llevó a cabo en una sola línea de producción la cual estaba compuesta por 33 procesos diferentes. Los procesos están acomodados de manera secuencial, sin flujos paralelos. En el estudio se analizó una semana en específico la cual contaba con ocho trabajos asignados para la línea. A los trabajos se les asignó un ID para este estudio (Q1, G4, B4, G1, N4, F1, J1, F4). La información sobre la demanda y los tiempos de procesamiento de cada trabajo en cada máquina se puede observar en las tablas 1 y 2.

Tabla 1. Tiempos de procesamiento del trabajo en la máquina (en segundos)

Máquina\ Trabajo	F ₁	G ₁	J ₁	Q ₁	B ₄	F ₄	G ₄	N ₄
SF1	0	0	0	34.440	0	0	0	176.400
QAS	40.262	39.144	78.288	5.520	72.226	15.670	14.774	13.112
PK1	18.144	17.640	35.280	1.464	59.894	8.791	4.925	61.740
SMA	19.354	18.816	37.632	0	73.987	14.906	5.443	2.352
MV1	107.136	10.416	20.832	0	22.901	5.733	2.851	1.176
SMB	26.093	25.368	0	0	0	0	0	0
MV2	20.218	19.656	0	0	0	0	0	0
RO1	11.059	10.752	21.504	0	45.802	11.466	5.443	0
PTH	14.170	13.776	0	0	31.709	14.524	8.035	15.288
WV1	5.875	5.712	0	0	12.331	2.675	1.555	2.352
PW1	23.155	22.512	0	0	24.662	10.702	2.851	41.160
IF1	0	336.000	0	0	125.074	37.456	10.368	0
WS1	0	3.024	0	0	26.424	4.969	3.888	0
WI1	0	2.520	0	0	22.901	3.440	12.960	0
BP1	0	6.720	0	0	61.656	14.906	7.776	0
BP2	0	5.880	0	0	42.278	14.906	3.888	0
FI1	0	18.312	0	0	110.981	48.539	1.814	0
F1	0	0	8.400	0	0	0	0	0
MM1	0	6.888	0	0	61.656	3.058	0	0
CC1	0	3.528	0	0	35.232	76.440	2.592	0
CC2	0	8.400	0	0	81.034	17.581	5.184	0
FI2	0	18.312	0	0	110.981	48.539	1.814	0
MI1	0	0	65.184	0	84.557	74.147	0	0
SS1	0	0	49.056	0	100.411	0	0	0
SI1	0	0	99.456	0	61.656	0	0	0
QA1	0	0	78.288	0	0	0	0	0
LL1	0	0	0	59.880	0	0	0	22.932
DP1	0	0	0	0	0	0	0	11.760
QV1	0	0	0	0	410.453	0	0	0

<i>Máquina\ Trabajo</i>	<i>F₁</i>	<i>G₁</i>	<i>J₁</i>	<i>Q₁</i>	<i>B₄</i>	<i>F₄</i>	<i>G₄</i>	<i>N₄</i>
HC1	0	0	0	0	0	19.874	16.330	0
LP1	0	0	0	0	0	0	0	29.400
PO1	0	0	0	0	0	0	0	92.904
FI4	0	0	0	0	0	0	0	137.004

Fuente: elaboración propia.

Tabla 2. Demanda por trabajo (en unidades)

<i>Trabajo</i>	<i>Demanda</i>
F ₁	288
G ₁	280
J ₁	560
Q ₁	200
B ₄	2936
F ₄	637
G ₄	432
N ₄	980

Fuente: elaboración propia.

Resultados y discusión

El modelo con el cual se obtuvo un mejor resultado fue el modelo 3, ya que presentaba el menor número de trabajos tardíos. Además, se hicieron todas las corridas necesarias para ver cómo se comportaba el modelo si ponderábamos las tres métricas. Para obtener el frente de Pareto, lo que se hizo fue darles una ponderación entre 0 y 1 a cada métrica de tal manera que en conjunto sumaran uno. Después se limpió la lista haciendo comparaciones entre cada ponderación para concluir cuáles de todas las ponderaciones dieron los mejores resultados (soluciones no-dominadas). Los resultados se pueden ver en la Tabla 3. La columna de *Alfas* muestra la ponderación de cada métrica. La columna de *Cmax* muestra el tiempo máximo de flujo, la columna de *Lmax*

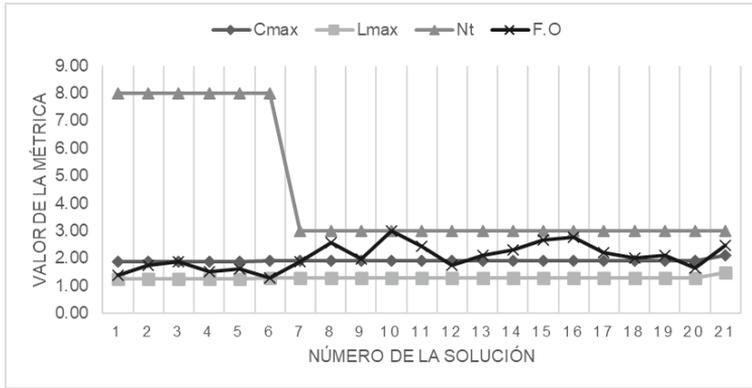
muestra la tardanza máxima de todos los trabajos, y la columna de Nt muestra el número de trabajos tardíos. La columna de $F.O.$ muestra la ponderación de las tres métricas anteriores de acuerdo con los valores de alfa, y a los factores aplicados. Se pueden ver los comportamientos de los tres objetivos de manera gráfica para cada ponderación en las Figuras 1 y 2.

Tabla 3. Soluciones no-dominadas con el Modelo 3

Número de la solución	Alfas	C_{max}	L_{max}	Nt	$F.O.$	Secuencia
1	0.2, 0.8, 0	1.8683172	1.2635172	8	1.3844772	G4-F4-J1-N4-B4-G1-F1-Q1
2	0.8, 0.2, 0	1.8683172	1.2635172	8	1.7473572	G4-J1-F4-N4-B4-Q1-G1-F1
3	1, 0, 0	1.8683172	1.2635172	8	1.8683172	G4-F4-N4-B4-Q1-J1-G1-F1
4	0.4, 0.6, 0	1.8683172	1.2635172	8	1.5054372	F4-G4-N4-B4-G1-J1-Q1-F1
5	0.6, 0.4, 0	1.8683172	1.2635172	8	1.6263972	G4-F4-N4-B4-Q1-J1-F1-G1
6	0, 1, 0	1.8987618	1.2939618	8	1.2939618	N4-G4-F4-B4-Q1-J1-G1-F1
7	0.4, 0.4, 0.2	1.9027572	1.2979572	3	1.88028576	J1-G4-Q1-F4-F1-N4-B4-G1
8	0.4, 0, 0.6	1.9027572	1.2979572	3	2.56110288	J1-G4-Q1-F4-F1-N4-B4-G1
9	0, 0.6, 0.4	1.9027572	1.2979572	3	1.97877432	J1-G4-Q1-F4-F1-N4-B4-G1
10	0, 0, 1	1.9027572	1.2979572	3	3	J1-G4-Q1-F4-F1-N4-B4-G1
11	0.2, 0.2, 0.6	1.9027572	1.2979572	3	2.44014288	J1-G4-Q1-F4-F1-N4-B4-G1
12	0.2, 0.6, 0.2	1.9027572	1.2979572	3	1.75932576	J1-G4-Q1-F4-F1-N4-B4-G1
13	0.8, 0, 0.2	1.9027572	1.2979572	3	2.12220576	J1-G4-Q1-F4-F1-N4-B4-G1
14	0, 0.4, 0.6	1.9027572	1.2979572	3	2.31918288	J1-G4-Q1-F4-F1-N4-B4-G1
15	0, 0.2, 0.8	1.9027572	1.2979572	3	2.65959144	J1-G4-Q1-F4-F1-N4-B4-G1
16	0.2, 0, 0.8	1.9027572	1.2979572	3	2.78055144	J1-Q1-G4-F4-F1-N4-B4-G1
17	0.4, 0.2, 0.4	1.9027572	1.2979572	3	2.22069432	J1-G4-Q1-F4-F1-N4-B4-G1
18	0.6, 0.2, 0.2	1.9027572	1.2979572	3	2.00124576	J1-G4-Q1-F4-F1-N4-B4-G1
19	0.2, 0.4, 0.4	1.9027572	1.2979572	3	2.09973432	J1-G4-Q1-F4-F1-N4-B4-G1
20	0, 0.8, 0.2	1.9027572	1.2979572	3	1.63836576	J1-G4-Q1-F4-F1-N4-B4-G1
21	0.6, 0, 0.4	2.0943102	1.4895102	3	2.45658612	J1-G4-Q1-F4-F1-G1-N4-B4

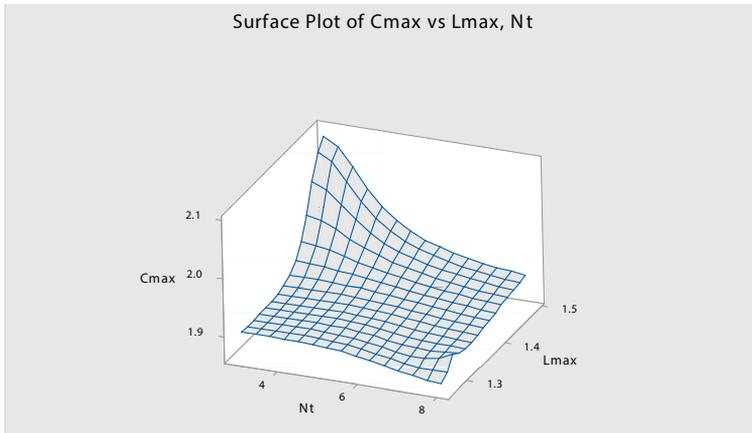
Fuente: elaboración propia.

Figura 1. Comportamiento de los objetivos para cada solución no-dominada



Fuente: elaboración propia.

Figura 2. Comportamiento de los objetivos para cada solución no-dominada en 3 dimensiones



Fuente: elaboración propia.

Podemos ver claramente que cuando el número de trabajos tardíos no se toma en cuenta (cuando su ponderación es cero), el desempeño del modelo no es nada bueno cuando analizamos los resultados obtenidos para el número de trabajos tardíos. Por ello, aunque la función objetivo ponderada es mayor que para otras soluciones en el frente de Pareto, preferimos las soluciones 7 a

21, que muestran el menor número de trabajos tardíos. Observando la tabla podemos ver que las soluciones que dieron un mejor desempeño en el número de trabajos tardíos fueron las siguientes:

J1-G4-Q1-F4-F1-N4-B4-G1

J1-Q1-G4-F4-F1-N4-B4-G1

J1-G4-Q1-F4-F1-G1-N4-B4

Para tener una referencia de comparación, la secuencia seguida por la empresa se muestra en la Tabla 4. Esta secuencia fue calculada con un criterio de fecha de entrega más próxima, ya que la empresa trabaja bajo este esquema para intentar tener un mayor nivel de servicio a su cliente. Se puede observar que esta solución no es mejor que ninguna de las soluciones obtenidas con el modelo, por lo que sería una solución dominada. Esto demuestra la superioridad de la utilización del modelo comparado con la utilización de una regla simple de secuenciación. Comparando contra la mejor solución no-dominada, con tres trabajos tardíos, se logra reducir en 57.14% el número de trabajos tardíos, y también se reducen en menor medida el tiempo total de procesamiento (6.36%) y la máxima tardanza (9.06%).

Tabla 4. Solución adoptada por la empresa para la instancia estudiada

<i>C_{max}</i>	<i>L_{max}</i>	<i>N_t</i>	<i>F.O.</i>	<i>Secuencia</i>
2.032071	1.427271	7	10.459342	J1-F4-B4-N4-F1-Q1-G4-G1

Fuente: elaboración propia.

Conclusiones

En este trabajo se propuso un modelo triobjetivo, con el fin de mejorar la situación que una empresa de la industria electrónica estaba viviendo. Esta empresa estaba teniendo entregas tardías a sus clientes, resultando en un bajo nivel de servicio. Se decidió usar este modelo ya que los modelos biobjetivo que se habían planteado anteriormente no estaban arrojando buenos resultados. Los dos primeros modelos no contemplaban el número de trabajos

tardíos, por lo que podemos concluir que cuando se toma en cuenta el número de trabajos tardíos en la optimización de una línea de producción se obtienen mejores resultados.

Tras analizar los resultados que los tres modelos arrojaron, podemos concluir que el modelo matemático planteado en este trabajo logró obtener mejores resultados que los métodos utilizados actualmente por la empresa. Un caso interesante que se propone en este trabajo podría ser el hacer la programación de los trabajos considerando subensambles. Esto volvería al modelo aún más complicado de lo que ya es, pero podría llegar a dar resultados interesantes para la industria electrónica.

Con los resultados de este trabajo se propone la utilización de modelos matemáticos como el presentado en este trabajo para así poder lograr una mejor planeación de la producción en las empresas manufactureras.

Lista de referencias

- Behnamian, J. y Fatemi Ghomi, S. M. T. (2011). Hybrid flowshop scheduling with machine and resource-dependent processing times. *Applied Mathematical Modelling*, 35(3), 1107-1123. Recuperado de <https://doi.org/10.1016/j.apm.2010.07.057>
- Campbell, H. G., Dudek, R. A. y Smith, M. L. (1970). A Heuristic Algorithm for the n Job, m Machine Sequencing Problem. *Management Science*, 16(10), B-630-B-637. Recuperado de <https://doi.org/10.1287/mnsc.16.10.b630>
- Chang, P., Chen, S. y Liu, C. (2007). Sub-population genetic algorithm with mining gene structures for multiobjective flowshop scheduling problems. *Expert Systems with Applications*, 33(3), 762-771. Recuperado de <https://doi.org/10.1016/j.eswa.2006.06.019>
- Cheng, M., Tadikamalla, P. R., Shang, J. y Zhang, S. (2014). Bicriteria hierarchical optimization of two-machine flow shop scheduling problem with time-dependent deteriorating jobs. *European Journal of Operational Research*, 234(3), 650-657. Recuperado de <https://doi.org/10.1016/j.ejor.2013.09.033>
- Cheng, T. C. E. (1991). Optimal assignment of slack due-dates and sequencing of jobs with random processing times on a single machine. *European*

- Journal of Operational Research*, 51(3), 348-353. Recuperado de [https://doi.org/10.1016/0377-2217\(91\)90310-r](https://doi.org/10.1016/0377-2217(91)90310-r)
- Collette, Y. y Siarry, P. (2004). Multiobjective Optimization. En *Decision Engineering*. Springer Berlin Heidelberg. Recuperado de <https://doi.org/10.1007/978-3-662-08883-8>
- Daniels, R. L. y Chambers, R. J. (1990). Multiobjective flow-shop scheduling. *Naval Research Logistics*, 37, 981-995. Recuperado de [https://doi.org/10.1002/1520-6750\(199012\)37:6<981::AID-NAV3220370617>3.0.CO;2-H](https://doi.org/10.1002/1520-6750(199012)37:6<981::AID-NAV3220370617>3.0.CO;2-H)
- Danneberg, D., Tautenhahn, T. y Werner, F. (1999). A comparison of heuristic algorithms for flow shop scheduling problems with setup times and limited batch size. *Mathematical and Computer Modelling*, 29(9), 101-126. [https://doi.org/10.1016/s0895-7177\(99\)00085-0](https://doi.org/10.1016/s0895-7177(99)00085-0)
- Dhouib, E., Teghem, J. y Loukil, T. (2013). Lexicographic optimization of a permutation flow shop scheduling problem with time lag constraints. *International Transactions in Operational Research*, 20(2), 213-232. Recuperado de <https://doi.org/10.1111/j.1475-3995.2012.00876.x>
- Framinan, J. M., Fernandez-Viagas, V. y Perez-Gonzalez, P. (2019). Using real-time information to reschedule jobs in a flowshop with variable processing times. *Computers & Industrial Engineering*, 129, 113-125. Recuperado de <https://doi.org/10.1016/j.cie.2019.01.036>
- Guinet, A. G. P. y Solomon, M. M. (1996). Scheduling hybrid flowshops to minimize maximum tardiness or maximum completion time. *International Journal of Production Research*, 34(6), 1643-1654. Recuperado de <https://doi.org/10.1080/00207549608904988>
- Hamdi, I. Loukil, T. (2014). Minimizing total tardiness in the permutation flowshop scheduling problem with minimal and maximal time lags. *Operational Research*, 15(1), 95-114. Recuperado de <https://doi.org/10.1007/s12351-014-0166-5>
- Han, Y.-Y., Duan, J.-H y Zhang, M. (2011). Apply the discrete artificial bee colony algorithm to the blocking flow shop problem with makespan criterion. *2011 Chinese Control and Decision Conference (CCDC)*, 2131-2135. Recuperado de <https://doi.org/10.1109/ccdc.2011.5968558>
- Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1(1), 6168. Recuperado de <https://doi.org/10.1002/nav.3800010110>

- Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P. y Werner, F. (2009). A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *Computers & Operations Research*, 36(2), 358-378. Recuperado de <https://doi.org/10.1016/j.cor.2007.10.004>
- Kalman, M. S., Rojas, O. G., Olivares-Benitez, E. y Nucamendi-Guillén, S. M. (2019). A MILP and Genetic Algorithm Approach for a Furniture Manufacturing Flow Shop Scheduling Problem. *Advances in Civil and Industrial Engineering*, 238-259. Recuperado de <https://doi.org/10.4018/978-1-5225-8223-6.ch011>
- Lee, C.-Y., Danusaputro, S. L. y Lin, C.-S. (1991). Minimizing weighted number of tardy jobs and weighted earliness-tardiness penalties about a common due date. *Computers & Operations Research*, 18(4), 379-389. Recuperado de [https://doi.org/10.1016/0305-0548\(91\)90098-c](https://doi.org/10.1016/0305-0548(91)90098-c)
- Lee, K., Zheng, F. y Pinedo, M. L. (2019). Online scheduling of ordered flow shops. *European Journal of Operational Research*, 272(1), 50-60. Recuperado de <https://doi.org/10.1016/j.ejor.2018.06.008>
- Li, J., Sun, K., Xu, D. y Li, H. (2010). Single machine due date assignment scheduling problem with customer service level in fuzzy environment. *Applied Soft Computing*, 10(3), 849-858. Recuperado de <https://doi.org/10.1016/j.asoc.2009.10.002>
- Li, J., Xu, D. y Li, H. (2018). Single Machine Due Date Assignment Scheduling Problem with Precedence Constraints and Controllable Processing Times in Fuzzy Environment. *Iranian Journal of Fuzzy Systems*, 15(6), 121-143. Recuperado de <https://dx.doi.org/10.22111/ijfs.2018.4371>
- Li, J., Yuan, X., Lee, E. S. y Xu, D. (2011). Setting due dates to minimize the total weighted possibilistic mean value of the weighted earliness-tardiness costs on a single machine. *Computers & Mathematics with Applications*, 62(11), 4126-4139. Recuperado de <https://doi.org/10.1016/j.camwa.2011.09.063>
- Lin, S.-W. y Ying, K.-C. (2016). Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics. *Omega*, 64, 115-125. Recuperado de <https://doi.org/10.1016/j.omega.2015.12.002>
- Liu, F., Wang, S., Hong, Y. y Yue, X. (2017). On the Robust and Stable Flowshop Scheduling Under Stochastic and Dynamic Disruptions. *IEEE Transactions on Engineering Management*, 64(4), 539-553. Recuperado de <https://doi.org/10.1109/tem.2017.2712611>

- Liu, G., Song, S. y Wu, C. (2013). Some heuristics for no-wait flowshops with total tardiness criterion. *Computers & Operations Research*, 40(2), 521-525. Recuperado de <https://doi.org/10.1016/j.cor.2012.07.019>
- Liu, W., Jin, Y. y Price, M. (2017). New scheduling algorithms and digital tool for dynamic permutation flowshop with newly arrived order. *International Journal of Production Research*, 55(11), 3234-3248. Recuperado de <https://doi.org/10.1080/00207543.2017.1285077>
- Loukil, T., Teghem, J. y Tuytens, D. (2005). Solving multi-objective production scheduling problems using metaheuristics. *European Journal of Operational Research*, 161(1), 42-61. Recuperado de <https://doi.org/10.1016/j.ejor.2003.08.029>
- Marichelvam, M. K. y Geetha, M. (2014). Solving tri-objective multistage hybrid flow shop scheduling problems using a discrete firefly algorithm. *International Journal of Intelligent Engineering Informatics*, 2(4), 284. Recuperado de <https://doi.org/10.1504/ijiei.2014.067190>
- Murata, T., Ishibuchi, H. y Tanaka, H. (1996). Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers & Industrial Engineering*, 30(4), 957-968. Recuperado de [https://doi.org/10.1016/0360-8352\(96\)00045-9](https://doi.org/10.1016/0360-8352(96)00045-9)
- Nagano, M. S. y Araújo, D. C. (2013). New heuristics for the no-wait flowshop with sequence-dependent setup times problem. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 36(1), 139-151. Recuperado de <https://doi.org/10.1007/s40430-013-0064-4>
- Nawaz, M., Ensore, E. E. y Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1), 91-95. Recuperado de [https://doi.org/10.1016/0305-0483\(83\)90088-9](https://doi.org/10.1016/0305-0483(83)90088-9)
- Ponnambalam, S. G., Jagannathan, H., Kataria, M. y Gadicherla, A. (2004). A TSP-GA multi-objective algorithm for flow-shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 23(11-12). Recuperado de <https://doi.org/10.1007/s00170-003-1731-x>
- Rahimi-Vahed, A. R., Javadi, B., Rabbani, M. y Tavakkoli-Moghaddam, R. (2008). A multi-objective scatter search for a bi-criteria no-wait flow shop scheduling problem. *Engineering Optimization*, 40(4), 331-346. Recuperado de <https://doi.org/10.1080/03052150701732509>

- Ruiz, R. y & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, 165(2), 479-494. Recuperado de <https://doi.org/10.1016/j.ejor.2004.04.017>
- Sawik T. (2011). *Scheduling in Supply Chains using Mixed Integer Programming*. Wiley, Hoboken, USA. ISBN 978-0-470-93573-6.
- Soewanda, J., Octavia, T. y Halim Sahputra, I. (2007). Robust-hybrid genetic algorithm for a flow-shop scheduling problem (A Case Study at PT FSCM Manufacturing Indonesia). *Jurnal Teknik Industri*, 9(2), 144-151. Recuperado de <https://doi.org/10.9744/jti.9.2.pp.%20144-151>
- Toktaş, B., Azizoglu, M. y Köksalan, S. K. (2004). Two-machine flow shop scheduling with two criteria: Maximum earliness and makespan. *European Journal of Operational Research*, 157(2), 286-295. Recuperado de [https://doi.org/10.1016/s0377-2217\(03\)00192-9](https://doi.org/10.1016/s0377-2217(03)00192-9)
- Zelazny, D. y Pempera, J. (2015). Solving multi-objective permutation flowshop scheduling problem using CUDA. *2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 347-352. Recuperado de <https://doi.org/10.1109/mmar.2015.7283900>

