

Optimización de la secuenciación con penalizaciones por adelanto y atraso con trabajos que se traslapan

Optimization of a scheduling problem with early and quadratic tardy penalties with overlapping jobs

Camila Bolaños Valenciano^{1*}, Mauren Valeria González Núñez¹, Luis Daniel Mora Mora¹, Alejandro Valverde Solano¹,
Luis Vargas Rodríguez¹

¹Universidad de Costa Rica, Facultad de Ingeniería, Escuela de Ingeniería Industrial, Costa Rica

Fecha de recepción: 4 de julio de 2022. **Fecha de aceptación:** 15 de enero de 2023.

***Autor de correspondencia:** camila.bolanos@ucr.ac.cr

Resumen. El objetivo principal de la presente investigación es resolver el problema de secuenciación con penalizaciones por anticipación y penalizaciones cuadráticas por tardanza para la finalización tardía del trabajo. Se llevó a cabo una amplia investigación para identificar un faltante en los modelos existentes. Debido a esto, fue posible identificar que los modelos abordados por la literatura carecían de la posibilidad de superposición de trabajos. Por lo tanto, la propuesta parte de un modelo preexistente que optimiza las penalizaciones por entrega anticipada al insertar tiempos muertos que provocan la reducción de las penalizaciones por anticipación. El modelo propuesto parte de un método heurístico que proporciona una solución inicial. Además de esto, se plantea un algoritmo para insertar tiempo de inactividad que genera una primera optimización seguida de una segunda optimización debido a la superposición de trabajos de modo que un trabajo pueda comenzar antes de que se complete el trabajo que lo precede con el objetivo de reducir la penalización por finalización tardía. Para este estudio, se hicieron las siguientes suposiciones: se permite el tiempo de inactividad, el primer trabajo comienza en el tiempo cero, todos los trabajos son independientes y estos pueden superponerse hasta cierto límite de cumplimiento. Los resultados obtenidos al realizar la parametrización y simulaciones demuestran que a través de la superposición de trabajos se logra la reducción de las penalizaciones por tardanza, lo que lleva a la validación de la propuesta.

Palabras clave. Fecha de vencimiento más temprana, penalizaciones por adelanto y atraso, programación, secuenciación, trabajos superpuestos.

Abstract. The main purpose of the present research is to solve the sequencing problem with earliness penalties and quadratic tardy penalties for late job completion. Extensive research was carried out to identify a possible gap in the existing models. Due to this, it was possible to identify that the models addressed by the literature lacked the possibility of overlapping jobs. The proposal therefore originates from a pre-existing model that optimizes the penalties for early delivery by inserting idle times that cause the reduction of the earliness penalties. The proposed model starts with a heuristic method that provides an initial solution. In addition to this, the algorithm for inserting idle time generates a first optimization followed by a second optimization that overlaps the jobs so that a job can start before the initial job is completed with the objective of reducing the penalty for late completion. For the present study, the following assumptions were made idle time is allowed, the first job starts at time zero, all jobs are independent, and these can overlap up to a certain limit of compliance. The parameterization and simulations were carried out where the results obtained show that through the overlapping of jobs, the reduction of lateness penalties is achieved, which therefore leads to the validation of the proposal.

Keywords. Earliest due date, early and tardy penalties, overlapping jobs, scheduling, sequencing.

1. Introducción

En la actualidad, las empresas de manufactura deben analizar múltiples variables con el fin de satisfacer las necesidades del cliente de forma eficaz y eficiente, para esto es

necesario la evaluación de los sistemas para la planificación y el control de la producción. Una parte importante dentro de este proceso es establecer el orden de producción de cada pedido, esto con el objetivo de cumplir con las fechas de entrega, tener mayor capacidad de respuesta y disminuir el *makespan*. A lo

largo de la historia, se han desarrollado diversos modelos de secuenciamiento. En el presente documento se aborda específicamente el secuenciamiento con penalidades por completar tarde o temprano (el modelo E/T).

Dicho lo anterior, plantear una penalización por las tardanzas o adelantos es importante ya que esto promueve que las entregas sean justo en el momento preciso, debido a que si se entrega después de la fecha requerida se incumpliría con la necesidad del cliente y se pueden obtener ventas perdidas. En el caso contrario, si se finaliza antes de la fecha requerida se pueden incurrir en costos de acarreo de inventario o tener problemas con la vida útil de los productos.

En el proceso de investigación se analiza un modelo de la literatura que busca reducir las penalidades por adelanto mediante la inserción del *idle time*, no obstante, en dicho modelo no se determina ninguna mejora con el otro tipo de penalización. De tal forma, en el presente documento se plantea la propuesta de parametrización de un algoritmo que permita disminuir las penalidades por completar tarde mediante el traslape de dos pedidos consecutivos durante su procesamiento. Para lo cual se plantean una serie de supuestos necesarios y un código mediante el lenguaje de programación de Python, ejecutado a través de la herramienta de Colab.

La importancia de la investigación radica en que se plantea una propuesta que optimiza el modelo E/T, planteando reducir al máximo ambos tipos de penalizaciones. Lo cual es de gran interés para las empresas que buscan evitar gastos por penalidades, aumentar su margen y a la vez cumplir las expectativas de sus clientes, entregando los productos justo en el momento indicado.

1.1 Estructura del artículo

En cuanto a la estructura del documento, primeramente, se realiza una revisión de la bibliografía, mediante la cual se identifica un modelo de secuenciación que posee una oportunidad de mejora para la cual se establece la crítica. Posteriormente se plantea una propuesta en la cual se definen los supuestos, la metodología a realizar, la parametrización del algoritmo y la simulación mediante un código de programación. Lo anterior permite realizar una validación de los datos y análisis de la información obtenida. Finalmente se plantean las conclusiones y posibles líneas de investigación para el futuro.

1.2 Estado del arte

A través de la revisión bibliográfica se pretende exponer los conceptos relacionados al secuenciamiento con penalidades por completar temprano o tarde; esto con el fin de establecer

una base de conocimiento, que permita determinar adecuadamente la propuesta a desarrollar.

1.2.1 Secuenciamiento

El secuenciamiento hace referencia a la asignación de recursos para llevar a cabo un proceso productivo [1]; es decir, busca establecer el orden en que se procesan los trabajos en una línea de producción, de tal manera que se optimice el tiempo y el uso de las maquinas o estaciones de trabajo. Para Dragcevi [2] el secuenciamiento tiene como objetivo utilizar de la mejor manera la capacidad con la que se cuenta, para esto es importante considerar las restricciones propias del proceso de producción.

Un secuenciamiento adecuado genera un incremento de la utilización de las maquinas, la eficiencia, y por ende, un aumento de la capacidad de producción, esto a través de la disminución del *makespan* y del inventario de trabajo en proceso (WIP, por sus siglas en ingles), así lo indica Rodríguez en [3]. Es importante tener claro que el *makespan* es el tiempo que transcurre entre el inicio del procesamiento del primer trabajo y la finalización del procesamiento del último trabajo. Es decir, en este tiempo se procesan todos los trabajos [4]. Por otro lado, el WIP hace referencia a aquellos trabajos que ya no son materia prima sino, que han empezado a procesarse, pero no se han convertido en producto terminado, es decir, es un intermedio en la transformación de materia prima, como así lo indican [3], [5].

Como se puede notar, los beneficios del secuenciamiento son claros, lo que deja en evidencia la importancia de este, con respecto a las exigencias presentes en el mercado actual. Donde destacan los procesos de producción caracterizados por el *high mix low volume*, nuevas formas de comercio como el *e-commerce* y clientes que desean sus productos en poco tiempo; por lo que se crea la necesidad en las empresas de trabajar de la manera más precisa y eficiente posible, aspecto que está relacionado con la metodología justo a tiempo (JIT, por sus siglas en ingles), que, como menciona Alidaee [6], tiene como objetivo completar los trabajos lo más cerca posible de sus fechas de vencimiento. Para que el JIT funcione se necesita de la coordinación de toda la cadena de suministro, siendo los procesos de producción parte fundamental de esta; por ende, se enfatiza en la gran importancia de secuenciar adecuadamente los trabajos en una organización, logrando terminar en el momento justo, no antes y no después; de esta manera, no solo optimizar el proceso de producción como tal, sino que satisfacer las necesidades del mercado.

1.2.2 Penalidades por adelanto o atraso

Antes de que la producción con una metodología justo a tiempo fuera de especial interés para la operación de las empresas, el secuenciamiento o programación de la producción, se analizaba según indicadores (tales como tiempo de flujo promedio, retraso promedio, etc.) que no contemplaban de manera global el proceso en cuestión, por lo que los trabajos se optimizaban de forma individual y esto no permite grandes beneficios en la reducción del *makespan*, así lo explica Baker en [7]. Ahora, entendiendo la importancia del JIT y llevando a cabo una integración mucho más adecuada para el análisis de la programación o el secuenciamiento de los procesos de producción, se consideran las consecuencias de completar antes o completar tarde los trabajos. Aquellos trabajos que se completan anticipadamente deben mantenerse en inventario como producto terminado, lo cual, puede ser perjudicial para la empresa, ya que aumenta los costos de acarreo de inventario, y dependiendo del producto que sea, puede tener problemas con su vida útil. Por otro lado, los trabajos que se completan después de la fecha en la que deberían, pueden ocasionar problemas con los clientes como es el caso de mala reputación y pérdidas monetarias a partir de ventas no realizadas [8]. Con el fin de ajustar los tiempos de procesamiento de los trabajos para que estos sean completados lo más cerca posible de la fecha de vencimiento, existen diversos planteamientos, en [6] Alidaee propone, para evitar completar anticipadamente, insertar tiempos de inactividad entre trabajos consecutivos, de tal manera que un determinado trabajo se empiece a procesar justo en el momento en el que la duración del tiempo de su procesamiento termine lo más cerca posible de la fecha establecida; sin embargo, Alidaee también enfatiza en el análisis exhaustivo de cada proceso en particular, porque es posible que ese tiempo de inactividad, el cual puede traducirse en detener una máquina, por ejemplo; sea mucho más costoso que en lo que se incurre para acarrear el inventario de producto terminado, por lo que en ese caso, lo mejor sí sería terminar de procesar ese trabajo antes de la fecha de vencimiento. Como es evidente, los factores que influyen en evitar que existan costos extras por completar antes o después de la fecha de vencimiento, pueden ser múltiples; es por esta razón, que las investigaciones y propuestas de diversos modelos de secuenciamiento, se han convertido en un campo de estudio muy amplio. En estos modelos se consideran diversos factores para que el secuenciamiento que ese modelo proporciona reduzca al máximo los costos y optimice la producción [9]. A partir de lo anterior, las penalidades por adelanto o atraso son una forma en que se cuantifica e incorpora a los modelos de secuenciamiento de la producción, esos costos en los que se incurre por no terminar lo más cerca

posible de la fecha de vencimiento, así se indica en [8], [10]. Estas formas de penalidades y diferentes modelos de secuenciamiento se detallan más adelante.

1.2.3 Reglas de despacho

Antes de definir los modelos de secuenciamiento como tal, es importante analizar lo que corresponde a las reglas de despacho, ya que estas definen una primera prioridad de orden para llevar a cabo el procesamiento de los trabajos [11]. Es decir, las reglas de despacho determinan un secuenciamiento inicial, fundamentado en diversos criterios básicos, a partir de esto, algunos modelos toman como referencia ese primer ordenamiento para llevar a cabo la optimización del *makespan* y otros criterios de interés según lo que establece el modelo. Existen diversas reglas de despacho, entre ellas destaca el *earliest due date* (a partir de ahora EDD), donde se procesa primero el trabajo con fecha de entrega más próxima, en [12] se afirma que esta regla de despacho es adecuada para reducir la varianza por tardanza. Se tiene también el *shortest processing time* (SPT), es decir, se ejecuta primero la orden con el tiempo de procesamiento más corto, esta regla supera al EDD en los resultados de la tardanza media como en el porcentaje de trabajos retrasados, así se menciona en [12]. Una regla muy similar a SPT es la *first off first on* (FOFO), en esta también se procesa primero el trabajo con el menor tiempo de procesamiento, la diferencia es que aunque este trabajo no esté en cola, la máquina estaría inactiva esperando su ingreso [13].

Al Harkan [13] también presenta otras reglas de despacho como *longest processing time* (LPT por sus siglas en inglés), la cual ingresa primero la orden que tenga el tiempo de procesamiento mayor. También destaca la regla *job slack time* donde la orden con el tiempo mínimo de inactividad es procesada primero (JST). Además, se tiene la *critical ratio* (CR), en la cual el trabajo con una razón o indicador menor es procesado primero. Asimismo, la regla de despacho *random* que consiste en seleccionar un trabajo de forma aleatoria de un grupo de trabajos que se encuentran en cola. Otra regla importante corresponde a *first come, first served*, esta indica que la primera orden en entrar es la primera que se ingresa para su procesamiento (FCFS), en caso contrario, se tiene *last come, first served* que establece que la última en entrar será la primera en procesarse (LCFS). Existe otra regla que indica que se procesará de primero aquel trabajo que tenga menos flexibilidad (*least flexible job*, LFJ); es decir, el trabajo que tenga alguna limitación o condición especial según los requerimientos establecidos para ese proceso de producción.

Es importante indicar que además de las reglas explicadas, puede haber muchas más, ya que en las organizaciones se tiende a secuenciar según las características propias del

proceso de producción, por ejemplo, aspectos relacionados al producto y sus especificaciones, a las precedencias que este tenga al momento de procesarse, entre muchos otros aspectos. Es por tal razón que los modelos de secuenciamiento deben analizarse adecuadamente al momento de implementarlos, ya que estos se establecen para diferentes especificaciones, y por lo tanto, no son adecuados para todos los procesos de producción.

1.2.4 Modelos

Los modelos de secuenciación se categorizan a partir de diversos criterios de los cuales son de interés para este análisis: el número de máquinas, el tipo de penalidades y la fecha de entrega. Son de especial interés los modelos que contemplan penalidades por entrega tardía o adelanto, llamados *E/T Models*. Es común que en dichos modelos se utilice la siguiente parametrización [3]:

n : número de trabajos por secuenciar

j : trabajos

d_j : fecha de entregaj

P_j : tiempo de procesamientoj

S_j : tiempo de inicioj

C_j : tiempo de finalización j

E_j : penalización por adelanto j

T_j : penalización por atraso j

En algunas instancias, para los modelos E/T, la fecha de entrega d_j es dada y en otros casos es una decisión; algunos poseen una misma fecha de entrega, mientras que otros permiten fechas de entrega distintas. Adicionalmente en algunos casos se tienen penalizaciones solo por tardanza, por adelanto, por ambas, y hay algoritmos que permiten rechazar ciertos trabajos en cuyo caso también se incurre en una penalización.

Verma y Dessouky [14] analizan el problema de determinar un cronograma de trabajos con duraciones de tiempo unitario en una sola máquina donde todos los trabajos tienen tiempos de procesamiento idénticos y las fechas de vencimiento no son necesariamente múltiplos enteros de los tiempos de procesamiento. El algoritmo se formula como un problema de programación entera que minimiza los ponderados totales de las penalizaciones por adelanto y atraso con respecto a fechas de vencimiento arbitrarias.

Akjiratikar y Yenradee [15] brindan un algoritmo para un conjunto de n trabajos independientes que deben programarse de forma no preventiva en una máquina que puede manejar como máximo un trabajo a la vez. Una vez que finaliza el trabajo actual, el siguiente trabajo se inicia inmediatamente sin tiempo de inactividad. En este algoritmo, el costo de instalación es dependiente de la secuencia y se describe

mediante una matriz $SC = [SC_{ji}]$, donde SC_{ji} es el costo de cambiar del trabajo j al i . Se supone que el costo de instalación del primer trabajo, es decir, SC_{0i} es cero. El tiempo de preparación es independiente de la secuencia y se suma al tiempo de procesamiento. Cada trabajo está disponible en el momento cero y se conoce su fecha de vencimiento distinta d_j .

Hassin y Shani [16] generalizan un algoritmo para el modelo E/T de tiempo polinomial que calcula un programa óptimo para una secuencia dada de tareas, en una sola máquina, asumiendo que las tareas tienen distintos pesos de penalización E/T, tiempos de procesamiento distintos y fechas de vencimiento distintas. Además, brindan resultados para los problemas con tiempos de procesamiento iguales e introducen la idea de la no ejecución, o bien el rechazo, de trabajos en los modelos de programación E/T. Como consecuencia del rechazo, proponen penalizaciones por cada tarea no ejecutada. La noción de no ejecución de la tarea coincide con la filosofía JIT mencionada anteriormente, en que cada incumplimiento de los acuerdos debe ser sancionado.

En [17] los autores plantean un algoritmo para secuenciar n trabajos en una sola máquina que minimice las funciones objetivo-regulares. Tomando en cuenta los costos de configuración y los tiempos de configuración entre trabajos de familias distintas; con la posibilidad de rechazar algunos trabajos por los cuales se tiene una penalización de costo por abandono.

Valente y Alves [18] consideran el problema de la minimización de los costos de programación de una sola máquina con costos cuadráticos de anticipación y tardanza, y sin tiempo de inactividad de la máquina. Realizan los supuestos de que los trabajos son independientes, la máquina está disponible continuamente desde el momento cero en adelante, y no se permiten preferencias. Para su análisis proponen varias heurísticas de despacho y analizan su rendimiento diversas instancias. También proponen procedimientos heurísticos que abordan específicamente las penalizaciones por adelanto y atraso, así como la función de costo cuadrática. Analizan varios procedimientos de mejora que se aplican una vez que las heurísticas han generado una secuencia.

En [11] se brinda un algoritmo con las mismas características que el anterior, sin embargo, considera un costo fijo de penalización y en donde los costos de configuración y penalización son para trabajos de la misma familia. Para cada trabajo, se tiene una fecha a partir de la cual está disponible y una fecha límite para realizarlo.

El modelo E/T expuesto en [19] se diferencia del anterior porque agrega la posibilidad de tener tiempos de inactividad, siempre considerando que el trabajo puede estar temprano, a

tiempo, tarde o ser rechazado tomando en cuenta conjuntamente las penalizaciones por anticipación y tardanza, una configuración dependiente de los tiempos y costos de secuenciación y penalizaciones por rechazo.

Tavakkoli Moghaddam y Moslehi [20] presenta la secuencia óptima de un conjunto de trabajos para una sola máquina también con tiempos de inactividad. En este algoritmo la función objetivo es minimizar la suma del máximo adelanto y atraso. Dado que este problema trata de minimizar y disminuir los valores de precocidad y tardanza, los resultados pueden ser útiles para diferentes sistemas de producción como el *just in time*. Los autores investigan el caso especial para determinar la secuencia óptima, considerando una fecha de vencimiento común, e introducen la estructura de la solución óptima, usando algunas órdenes simples. En el caso general, se desarrollan las condiciones de vecindad y se determina el conjunto dominante para cualquier solución óptima. El método de *Branch and Bound* o bien, ramificación y acotación, se utiliza para resolver el problema y también se introducen las cotas superior e inferior adecuadas.

Bruno [21] también brinda un modelo con un algoritmo exacto y otro heurístico para una sola máquina, que permite la existencia de tiempos de inactividad con distintas ventanas de tiempo y tiempos de configuración dependientes de la secuencia.

Ribeiro y de Souza [22] consideran las ventas de tiempo con respecto a la fecha de entrega. Este algoritmo trata el problema de programación de una sola máquina con multas por adelanto y tardanza, considerando distintas ventanas de vencimiento y tiempo de configuración dependientes de la secuencia. Debido a su complejidad, los autores proponen un algoritmo genético adaptativo para resolverlo. Se utilizan cinco operadores de búsqueda para explorar el espacio de soluciones y la probabilidad de elección de cada operador depende del éxito en la búsqueda anterior.

Steiner y Zhang [23] consideraron el problema de secuenciación con penalizaciones por tardanza en una sola máquina en donde se pueda negociar la fecha de entrega. Al darse cuenta de que no se puede cumplir con la fecha de entrega original, el algoritmo permite que el proveedor asigne un plazo de entrega revisado (fecha de vencimiento) a cada trabajo, pero tiene que pagar una multa de costo por unidad de tiempo extendido. Si un trabajo termina tarde incluso con respecto al tiempo de entrega prometido revisado, entonces se debe pagar una multa por tardanza mayor. En este caso demostraron que este problema es equivalente a minimizar la morosidad total con rechazo respecto a los plazos originales y proporcionaron un algoritmo de tiempo pseudo-polinomial y un esquema de

aproximación de tiempo completamente polinomial para el problema.

Koulamas [24] indica que las negociaciones de la fecha de entrega suelen resultar en penalidades proporcionales al atraso en la entrega, por lo que considera el problema de programación de una sola máquina con tiempos de entrega revisados y penalizaciones por tardanza dependientes del trabajo. Demuestra que el algoritmo de programación dinámica exacto para el problema más simple con penalizaciones por tardanza independientes del trabajo propuesto en [23] puede ser extendido al problema de las penalizaciones por tardanza dependientes del trabajo. Cuando las multas por tardanza dependientes del trabajo están limitadas por una función polinomial dependiente del número de trabajos, el algoritmo de programación dinámica se puede convertir a un esquema de aproximación con tiempo completamente polinomiales implementando la técnica de “escalar la entrada”.

Baker [9] brinda un algoritmo del tipo *Branch and Bound* para el problema de programación estocástica de una sola máquina con el objetivo de minimizar costos por adelanto y atraso, en donde las fechas de vencimiento son decisiones. Los autores en [25] también desarrollan un algoritmo heurístico del tipo *Branch and Bound* pero en este caso para encontrar la solución óptima para un modelo E/T con máquinas paralelas. Proponen un modelo donde cada máquina está procesando continuamente y puede procesar solo un trabajo a la vez. Esto para n trabajos, los cuales están disponibles desde el momento cero, siempre con el objetivo de minimizar las penalizaciones por adelanto y atraso. El algoritmo toma en consideración los tiempos de alistado y tiempos de inactividad.

Zhang y Lu [26] también analizan el problema de programación de máquinas paralelas con fechas de lanzamiento y rechazo. Si un trabajo es rechazado se paga por la penalización, si es aceptado se procesa en una de las m máquinas paralelas idénticas. El objetivo es minimizar la suma de los *makespan* de los trabajos aceptados y la penalización total debido a los trabajos rechazados. Los autores brindan dos algoritmos. Uno de tiempo pseudopolinomial para cuando m es una constante fija, y un algoritmo de 2-aproximaciones para cuando m es arbitraria.

Un algoritmo que se diferencia de los demás mencionados es el que se presenta en [27]. Dicho método amplía la literatura sobre la programación de trabajos sujetos a fechas de entrega, al suponer que los trabajos pertenecen a agentes individuales que son racionales y actúan estratégicamente para maximizar sus utilidades. Los autores proponen un sistema de servidor único con clientes que envían sus trabajos para ser procesado en un taller de una sola máquina de acuerdo con un orden del tipo *first come first serve* (FCFS). Los trabajos que llegan

simultáneamente se ordenan aleatoriamente. Los clientes tienen una fecha de entrega d_j deseada para la cual sus trabajos ya deben estar procesados; pero la hora de inicio real depende de sus tiempos de llegada en relación con los demás, según lo dicta el régimen FCFS. Así resulta lo que los autores denominan un juego donde los jugadores son los clientes, las estrategias son los tiempos de llegada al sistema, y cada jugador busca minimizar sus esperadas penalizaciones por adelanto y tardanza. Se supone que el número de clientes es finito y cada uno tiene un solo trabajo que requiere procesamiento. Cuando algunos clientes llegan juntos, se ordenan de acuerdo con un sorteo aleatorio uniforme. Si todos los clientes tienen una fecha de vencimiento común o fechas de vencimiento cercanas, entonces llegar junto con otros clientes es potencialmente beneficioso porque se espera que cada cliente esté en el "medio", que puede estar más cerca de la fecha de vencimiento que llegar antes o después de todos los demás.

Los expuesto anteriormente constituyen los principales algoritmos presentes en la literatura con respecto a los modelos E/T.

1.3 Objetivo del trabajo

Según la bibliografía consultada, se determina que el secuenciamiento con penalidades por completar temprano o tarde es abordado por diversos modelos matemáticos. Los trabajos dentro de una secuencia poseen dos interpretaciones posibles, donde se entiende un trabajo como una operación y donde se entiende como una orden o pedido. En esta investigación se aborda la segunda interpretación.

Para la crítica se considera el supuesto de que dada un secuencia S , el pedido en la posición j debe ser terminado para poder iniciar el pedido en la posición $j+1$; es decir, dos pedidos no se pueden llevar a cabo de manera simultánea, de tal forma que se traslapen en su ejecución. Dicho supuesto teórico no refleja realmente lo que sucede en la práctica. Además, tener que esperar hasta que un pedido sea completado para poder iniciar el siguiente es un posible desperdicio de recursos.

Por estas razones se considera pertinente excluir este supuesto del modelo, de tal forma que se contemple la posibilidad de que el pedido j no debe concluirse por completo para poder iniciar el pedido $j+1$, es decir, se pueden traslapar parcialmente reduciendo la penalización total de la secuencia.

2. Materiales y métodos

El modelo en que se basa esta crítica es el desarrollado por Schaller [28], para el cual se tiene un secuenciamiento con penalización por entrega temprana y tardía, donde se obtiene una solución inicial que posteriormente se optimiza con la

inserción de tiempo ocioso. Esto con el objetivo de reducir la penalización en los trabajos que terminan temprano, donde al añadir tiempo ocioso se logra acercar la fecha de finalización del trabajo con la fecha de entrega.

La propuesta consiste en reducir, además, aquellos trabajos que poseen penalización con entrega tardía por medio del traslape de pedidos. Esto se realiza con un código de programación en el lenguaje Python, por medio de la plataforma Colab.

Además, se requiere tomar una serie de supuestos en torno al modelo matemático que se va a implementar. El modelo por implementar es un secuenciamiento con penalizaciones por entrega temprana o tardía, con penalización lineal en el primer caso y cuadrática en el segundo, debido a que castiga más las entregas tardías. Para ejecutar la propuesta se realiza un código de programación que contrasta la penalización total antes de aplicar la mejora y después.

Los supuestos que se consideran para llevar a cabo la propuesta se presentan a continuación:

- Se tiene n trabajos, todos disponibles en el tiempo cero.
- Los trabajos son independientes.
- El primer trabajo inicia en el tiempo cero.
- Se permite el tiempo ocioso.
- Los trabajos se pueden traslapar hasta cierto límite.

La propuesta consiste en realizar un secuenciamiento por medio de un método heurístico, actualmente mediante el modelo se obtiene una solución inicial, la cual posteriormente se optimiza al insertar tiempo ocioso y reducir así las penalizaciones por entrega temprana (este análisis ya lo incluye el modelo en estudio), posteriormente mediante la propuesta se plantea que los trabajos se traslapen, para reducir las penalizaciones por entrega tardía.

Para lograr que se traslapen los trabajos (pedidos) se plantea como una característica intrínseca de cada trabajo que un porcentaje de su tiempo de ejecución se pueda traslapar con otro. Es decir, el trabajo en la posición $j+1$ puede traslaparse con el trabajo j , de tal forma que un porcentaje de dichos trabajos se realice de manera simultánea, esto siempre y cuando el trabajo j permita este traslape y hasta que se minimice la penalización por entrega tardía del trabajo $j+1$. El traslape se da entre el inicio del trabajo $j+1$ y el final del trabajo j .

2.1 Parametrización

Para simular de manera más realista el modelo de secuenciamiento con penalización por entrega temprana o tardía, se integran al modelo desarrollado por [28] penalizaciones por entrega temprana (α_j) y por entrega tardía (β_j) exclusivas de cada pedido. Esto debido a que en una

empresa los pedidos usualmente se dirigen a distintos clientes, por lo que las repercusiones por no entregar en la fecha establecida van a depender de la forma en que se negocie con cada cliente. Con esto, se plantea la función objetivo de la siguiente forma (ecuación 1):

$$Z = \alpha_j * \max(d_j - C_j, 0) + \beta_j * \max(C_j - d_j, 0) \quad (1)$$

Esta función plantea una sumatoria de las penalizaciones por entrega temprana o tardía, donde los pedidos que terminan después de la fecha de entrega se castigan de forma más fuerte, considerando que esto puede dañar las relaciones con los clientes, la reputación de la empresa, entre otros [29]. Además, cabe destacar que cada pedido j puede tener únicamente alguna de las dos penalizaciones o ninguna.

Es importante resaltar que las variables que se toman en cuenta para este problema corresponden al número de trabajo como una identidad, la fecha de entrega, el tiempo de procesamiento, tiempo de inicio y tiempo de finalización. La simbología utilizada respectivamente es la que se presenta a continuación.

- d_j = fecha de entregaj
- P_j = tiempo de procesamientoj
- S_j = tiempo de inicioj
- C_j = tiempo de finalizacion' j
- para $j = (1, 2, \dots, n)$

Una vez definida la función objetivo, el primer paso para plantear el traslape de los pedidos es determinar si las características del pedido permiten que una parte del siguiente pedido se ejecute de forma simultánea. Considerando que en una línea de producción al ir terminando el proceso se dan actividades como transportes, empaques, operaciones finales, que no dependen de las primeras actividades del proceso, se establece un porcentaje para cada pedido en el que la línea de producción debe dedicarse exclusivamente a ese pedido. De esta manera, el complemento de dicho porcentaje determina cuanto tiempo se puede trabajar simultáneamente otro pedido. Para lo cual se establece la variable X_j , para $j = 1, 2, \dots, n$.

Una vez definida esta variable, al evaluar cada pedido primero se determina si existe una penalización por entrega tardía, para esto se compara el tiempo de finalización con su fecha de entrega. En caso de tener entrega tardía, es decir, cuando la fecha de finalización es mayor a la fecha de entrega, entonces es viable realizar el traslape, por lo que el siguiente paso consiste en determinar qué tanto traslape es pertinente o posible realizar. Esto considerando que el trabajo anterior permite un traslape hasta cierto límite o que dicho límite exceda el tiempo que se requiere para alcanzar la fecha de entrega (ecuaciones 2 y 3).

$$Si, C_j > d_j \quad (2)$$

\Rightarrow Es viable realizar traslape

$$M_{Tj} = C_j - d_j \quad (3)$$

Entonces, al verificar que el tiempo de finalización del pedido es mayor a la fecha de entrega, calculamos qué tanto es necesario traslapan para reducir la penalización al mínimo del pedido j . Conociendo este límite, lo contrastamos con la capacidad de traslape del pedido anterior, que definimos como T_{j-i} , es decir, verificamos si es posible hacer la reducción de la penalización al máximo o únicamente una parte de esta, para esto definimos lo siguiente (ecuaciones 4, 5 y 6):

$$T_{j-1} = P_{j-1} * (1 - X_{j-1}) \quad (4)$$

$$Red_j = \min(T_{j-1}, M_{Tj}) \quad (5)$$

$$Red_j = \min(P_{j-1} * (1 - X_{j-1}), C_j - d_j) \quad (6)$$

La variable T_{j-i} , se obtiene al multiplicar el tiempo de procesamiento de dicho trabajo por el complemento del porcentaje de tiempo que la línea debe dedicarse exclusivamente a ese pedido. Por ende, la interpretación de esta variable es el tiempo en que es posible traslapan ese pedido con el que sigue en la secuencia.

Seguidamente, la variable Red_j representa el tiempo que se traslapan el pedido evaluado con el anterior, tomando el valor mínimo entre T_{j-i} y M_{Tj} esto debido a que la primera limita el tiempo de traslape según lo que el pedido permita y la segunda es la máxima reducción que minimiza la penalización.

En caso de que T_{j-i} sea mayor a M_{Tj} esto implica que el trabajo anterior permite un traslape más grande del óptimo (el que minimiza la penalización) y por ende se traslapan M_{Tj} unidades de tiempo. Mientras que si T_{j-i} es menor a M_{Tj} eso implica que no es permitido traslapan la cantidad de tiempo ideal y por ende se traslapan todo lo posible sin lograr una reducción total de la penalización.

Una vez que se determina esta reducción máxima para el trabajo j , se modifica tanto el tiempo de inicio como el tiempo de finalización, acercándolo a su fecha de entrega y por ende reduciendo la penalización. La modificación de estos dos valores se realiza para cada trabajo que termina tarde y se expresa de la siguiente manera (ecuaciones 7, 8 y 9).

$$S'_j = S_j - Red_j \quad (7)$$

$$C'_j = C_j - Red_j \quad (8)$$

$$\{\forall j \in (1, 2, \dots, n) : C_j - d_j > 0\} \quad (9)$$

Con las reducciones determinadas para cada trabajo que finaliza tarde originalmente se calcula las nuevas

penalizaciones aplicando la propuesta con el fin de evaluar si efectivamente se presenta una mejora en la función objetivo.

Con esta propuesta, se plantea una metodología para reducir las penalizaciones por entrega temprana o tardía en una secuencia de pedidos, donde primeramente se utiliza una regla de despacho mediante la cual se obtiene una secuencia inicial. Seguidamente se procede con una primera optimización que inserta tiempo ocioso para reducir las penalizaciones por entrega temprana [28] y una segunda optimización en la que se traslapan los pedidos con el fin de reducir las penalizaciones por entrega tardía.

2.2 Simulación

Recapitulando, el problema que se plantea es el secuenciamiento de n trabajos de tal forma que se reduzcan al máximo las penalizaciones por entregas antes o después de la fecha de entrega. Para esto, inicialmente se utiliza alguna de las reglas de despacho disponibles para generar una secuencia inicial, en este caso se utiliza el criterio EDD (*earliest due date*), sin embargo, es importante resaltar que se puede implementar cualquier otra regla de despacho.

Seguidamente, mediante el trabajo realizado por [28] se inserta el tiempo ocioso en aquellos pedidos que tienen una penalización por entrega temprana como el primer paso de la optimización. El segundo paso consiste en determinar aquellos pedidos que se puede traslapar y modificar los tiempos de inicio y finalización para reducir las penalizaciones, para realizar esto se generó el siguiente código de programación.

- Secuenciamiento inicial
#Procedimiento EDD
#[Numero de trabajo, due date, tiempo de procesamiento, tiempo de inicio, tiempo de finalizacion, penalizacion unitaria, X_j] trabajos=[[1,34,12,0,0,1,0.75],
[2,15,8,0,0,1,0.82], [3,26,5,0,0,1,0.94],
[4,16,11,0,0,10.90],
[5,8,7,0,0,1,0.80]]
trabajos= sorted (trabajos, key=itemgetter (1))
#Definicion de tiempos de inicio y tiempo de finalizacion
x=0
trabajos [x][4]= trabajos [x][3]+ t [x] [2]
for i in trabajos:
if x < len(trabajos)-1:
trabajos [x+1][3]=trabajos [x][3]+ trabajos[x][2]
trabajos [x+1][4]=trabajos [x+1][3]+ trabajos [x+1][2]
x=x+1

Para representar los trabajos en el código de programación se realiza una lista de listas, donde cada elemento de la lista es un pedido que contiene datos sobre el número de pedido, su

fecha de entrega, tiempo de procesamiento, tiempo de inicio, tiempo de finalización, costo de la penalización y porcentaje de tiempo de procesamiento en que la línea de producción se debe dedicar a este pedido. En este paso se secuencian los trabajos según la fecha de entrega más próxima y se definen los tiempos de inicio y finalización de los pedidos de tal forma que cada pedido inicie uno tras otro.

- Primera Optimización
p= len(trabajos)
while(p>0):
if (trabajos [p-1][4] < trabajos[p-1][1]):
insertarIddle(p) p=p-1

En esta sección, se evalúa si es conveniente insertar tiempo ocioso antes de un pedido para lograr acercarlo a su fecha de entrega, el método utilizado es el desarrollado por [28]. El ciclo comienza analizando el último trabajo de la secuencia y prosigue con el anterior y el tras anterior y así sucesivamente hasta llegar al primero.

- Segunda Optimización
j=2
while (j<=n):
if (trabajos [j-1][4]> trabajos [j-1][1]):
Mtj=trabajos[j-1][4]- trabajos[j-1][1]
Tj-1=trabajos[j-2][2]* (1- trabajos[j-2][6])
Redj=min (Mt, T)
trabajos [j-1][3]=trabajos[j-1][3]-Redj
trabajos[j-1][4]=trabajos [j-1][4]-Redj
j=j+1

En este segundo paso de la optimización, se procede a evaluar cada trabajo a partir del que se ubica en la segunda posición de la secuencia para determinar si es conveniente realizar un traslape o no, siempre y cuando sea permitido. Para esto, se evalúa si el tiempo de finalización de dicho trabajo es mayor a su fecha de entrega, en caso de que sí, se calculan las variables M_{T_j} y T_{j-1} las cuales representan la reducción necesaria para llegar a la fecha de entrega del trabajo evaluado y el tiempo máximo que permite el pedido anterior que se traslape.

3. Resultados y discusión

Para validar que la penalización total por entregas tardías o tempranas de pedidos se reducen al aplicar la propuesta de optimización, se procede a ejecutar el programa en la plataforma Collab con distintos conjuntos de trabajos generados aleatoriamente por medio de Excel.

Para esto se plantean algunas limitaciones, tal como establecer una fecha de entrega máxima de 48 horas, considerando una jornada de ocho horas de lunes a sábado. Además, se cuenta con un tiempo de procesamiento máximo de un pedido es de 15 horas, las penalizaciones por unidad de tiempo están en el rango de \$1 a \$10, y por último, la variable X_j se maneja en un rango de 70 % a 100 %.

Los montos de las penalizaciones van a depender de la negociación con cada cliente, así como del tipo de producto y la urgencia de la entrega. El porcentaje de tiempo que se designa la línea exclusivamente a un pedido se mantiene alto considerando que las operaciones y actividades en las que se puede trabajar simultáneamente se dan hasta el final de proceso.

Estos parámetros se definieron de forma arbitraria debido a que se ejecuta el programa con distintas cantidades de pedidos por secuenciar ($n = 5, n = 8$ y $n = 12$) con el fin de comparar los resultados para cada cantidad en condiciones similares. Para cada corrida se aleatorizan todas las variables mencionadas anteriormente para cada trabajo y se analizan los resultados que se obtienen de la optimización.

Para evaluar la mejora en la penalización se calcula la diferencia entre el resultado en la primera optimización y la segunda optimización. Este cambio se documenta tanto en términos absolutos como porcentuales, con el fin de ilustrar de forma relativa el cambio en la penalización (ecuaciones 10 y 11).

$$I_{AB} = \Delta \text{Penalización} \quad (10)$$

$$I_{RE} = \frac{\Delta \text{Penalización}}{\text{Penalización}_{op1}} \quad (11)$$

De esta manera, el primer indicador muestra el cambio absoluto en términos de dinero entre la primera y segunda optimización y el segundo la muestra en términos relativos. El denominador del cambio porcentual corresponde a la penalización obtenida con la primera optimización de la secuencia.

Otra manera de evaluar los resultados es por medio de la reducción en el tiempo de finalización de la totalidad de los pedidos, lo cual resulta importante debido a que en la planificación y programación de las operaciones de una empresa es sumamente relevante conocer los tiempos requeridos para cumplir los pedidos.

Ante esto, se plantean dos indicadores análogos a los anteriores que ilustren la reducción absoluta y relativa en el tiempo total requerido para cumplir con el conjunto de pedidos. Dado que las penalizaciones unitarias se obtienen aleatoriamente (α_j y β_j) resulta importante analizar también el

cambio que se presenta en el tiempo requerido para cumplir con los pedidos (ecuaciones 12 y 13).

$$T_{AB} = \Delta C_n \quad (12)$$

$$T_{RE} = \frac{\Delta C_n}{C_{n_{op1}}} \quad (13)$$

El primer indicador calcula la reducción absoluta en el tiempo de finalización del último trabajo, mientras que el segundo la calcula en términos relativos, colocando en el denominador el tiempo de finalización del trabajo n en la primera optimización. Una vez definidos los indicadores se procede a analizar los resultados obtenidos con una muestra de 30 conjuntos de trabajos para cada cantidad de trabajos en la secuencia.

Esta cantidad de conjuntos de trabajo se elige debido al uso del teorema del límite central, el cual establece que para una muestra lo suficientemente grande ($n \geq 30$) de una variable de interés que se modele bajo cualquier distribución de probabilidad, la distribución de la media muestral será aproximadamente normal y la media será la misma que la de la variable de interés, mientras que la desviación típica será aproximadamente el error estándar [30].

En la tabla 1 se presentan los indicadores obtenidos para la simulación ejecutada. Cabe destacar que estos son los valores promedios de las 30 corridas para cada conjunto de trabajos.

Tabla 1. Resultados promedios obtenidos de la simulación.

Cantidad de trabajos (n)	T_{AB}	T_{RE}	I_{AB}	I_{RE}
5	0.97h	2.06%	\$366	14.51%
8	1.64h	2.51%	\$1493	16.11%
12	1.60h	1.66%	\$4656	5.83%

3.1 Reducción porcentual de la penalización por tardanza

Para el tamaño de conjunto igual a cinco pedidos (figura 1), la reducción porcentual se concentra en el rango entre cero y trece por ciento (17 de cada 30 observaciones), así como el rango entre trece y veintiséis por ciento (9 de 30 observaciones) como lo muestra la curva del diagrama de densidad. Incluso se llegaron a presentar reducciones en la penalización monetaria de hasta un 45 %.

Esto lleva a que la reducción en la penalización que se obtiene en promedio al aplicar la segunda optimización es de un 14.51 % lo cual indica una mejoría relevante en la penalización total para dicha secuencia.

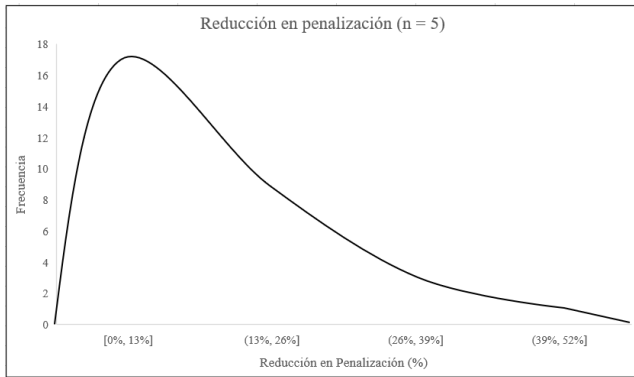


Figura 1. Gráfico de densidad I_{RE} ($n = 5$).

Análogamente, para el tamaño de conjunto igual a ocho pedidos (figura 2) las reducciones porcentuales se distribuyen un tanto más uniformemente entre 4.3 % hasta 22.6 %, con una cantidad similar de observaciones en dichos rangos presentados en el diagrama de densidad. En este caso, el valor máximo que se alcanzó fue de un 35 %. En promedio, los datos muestran una reducción del 16.11 % en la penalización total, por lo que igualmente representan una mejoría significativa para dicha secuencia.

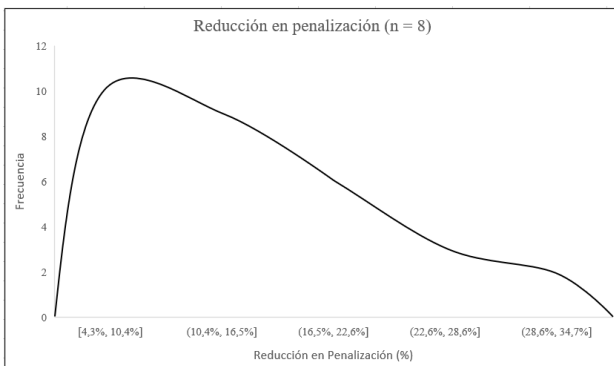


Figura 2. Gráfico de densidad I_{RE} ($n = 8$).

Para el caso del tamaño de subconjunto igual a doce pedidos (figura 3), se obtuvieron resultados un tanto distintos, debido a que las reducciones porcentuales resultaron ser más bajas que en los casos anteriores. Esto porque dicha cantidad de pedidos genera mayores entregas tardías, lo cual aumenta la penalización de forma muy importante.

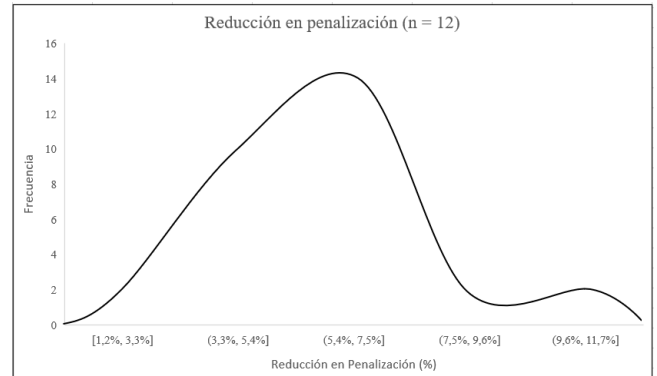


Figura 3. Gráfico de densidad I_{RE} ($n = 12$).

Aproximadamente la mitad de las observaciones, presentaron una reducción entre 5.4 % y 7.5 %, con un promedio de 5.83 % como resultado al aplicar la optimización, esto indica una mejora importante pero menor a los casos anteriores.

3.2 Reducción monetaria de la penalización por tardanza

El gráfico de cajas de la reducción monetaria para cada tamaño de conjunto de trabajos (figura 4) muestra que las reducciones son cada vez más importantes entre mayor sea la cantidad de trabajos. En promedio, se obtuvieron valores de 366 \$, 1493 \$ y 4656 \$ respectivamente para cada tamaño.

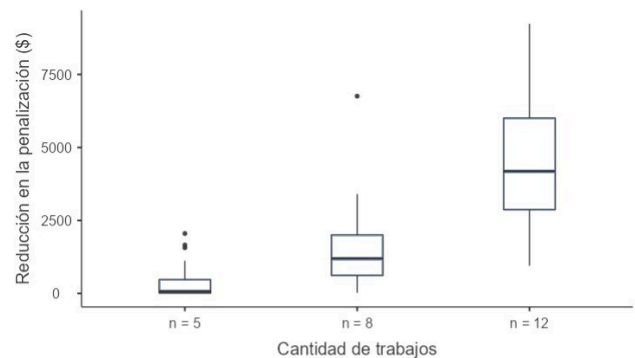


Figura 2. Gráfico de cajas para I_{AB} ($n = 5$, $n = 8$ y $n = 12$).

Esto se da debido a que entre mayor sea la cantidad de trabajos por secuenciar, se presenta un aumento en las penalizaciones por entrega tardía ya que las condiciones en las que se simula el programa establecen una fecha de entrega máxima de 48 horas. Además, de esto como dicha penalización es cuadrática la reducción es más significativa por cada unidad de tiempo que se traslapa alguno de los pedidos.

3.4 Reducción del tiempo de finalización

La reducción de tiempo de finalización de un set de pedidos, que es igual al tiempo de finalización del último pedido de dicho set, no mostró una mejoría contundente. Como lo muestra la figura 5 porcentualmente las reducciones fueron de 2.06 %, 2.51 % y 1.66 % respectivamente para cada tamaño de conjunto de trabajos.

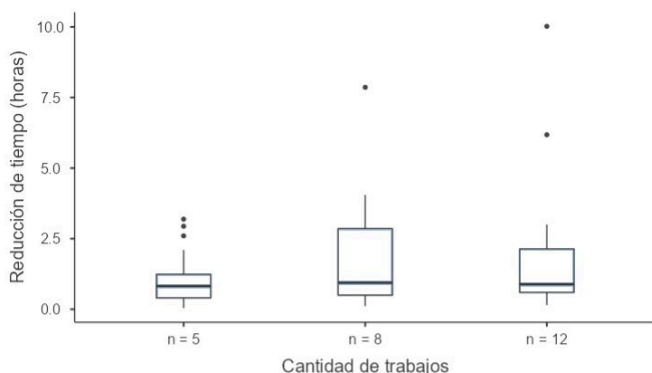


Figura 3. Gráfico de cajas para I_{AB} ($n = 5$, $n = 8$ y $n = 12$).

Este se debe a que, en la mayoría de las ocasiones, el traslape que permite realizar el penúltimo trabajo es muy pequeño, ya sea por un alto valor de la variable X_j o un bajo valor de la variable X_j . Es decir, que el pedido anterior requiere un que un alto porcentaje de su tiempo la línea se dedique exclusivamente a producirlo, o que su tiempo de procesamiento es corto y por ende el traslape que se permite realizar es poco.

Otro motivo por el que la reducción es baja es porque el algoritmo busca acercar el tiempo de finalización con la fecha de entrega de cada trabajo. Por lo que, si estas dos variables tienen valores similares para el último trabajo de la secuencia, la reducción en términos de tiempo será muy poca.

4. Conclusiones

Al aplicar la optimización que se propone se logra una reducción significativa en la penalización total para los tamaños de subconjunto igual a cinco y ocho (15% aproximadamente) y para el tamaño igual a doce (5.6%), lo que

implica que el algoritmo propuesto sí genera una mejora en la penalización que recibe la secuencia determinada.

La reducción en el tiempo de finalización de cada conjunto de trabajos no presentó una mejora significativa con el algoritmo presentado, esto debido a que lo que busca es acercar cada trabajo a su fecha de entrega con el propósito de reducir las penalizaciones, no una disminución en tiempo total requerido para completar los pedidos.

Dadas las condiciones de simulación, a mayor cantidad de pedidos por secuenciar, mayor es la reducción de la penalización en términos absolutos, lo cual se genera gracias a la ejecución simultánea de los pedidos que permite finalizar cada uno más cerca de su fecha de entrega.

Con esto, al tomar una secuencia inicial determinada por medio de un método heurístico es posible optimizar las penalizaciones por entrega temprana o tardía al insertar tiempo ocioso y procesar pedidos simultáneamente, según las restricciones de exclusividad de la línea de producción y tiempo de procesamiento de cada trabajo.

Dado que la optimización propuesta permite únicamente traslapar el final de un pedido con el inicio del siguiente, se propone como futura línea de investigación la posibilidad de traslapar pedidos desde un inicio hasta cierto límite, bajo el contexto en el que ambos pedidos compartan operaciones iniciales y posteriormente se deban procesar uno tras de otro.

Esto implicaría segmentar un pedido en varias partes, lo cual va en contra de uno de los supuestos que una gran cantidad de modelos sobre penalizaciones por entrega tardía o temprana asumen, el cual establece que un pedido o trabajo no se puede segmentar.

Una de las implicaciones que posee esta propuesta consiste en la generación de inventarios en proceso, donde al iniciar con dos pedidos simultáneamente, para posteriormente seguir procesando únicamente uno debido a la capacidad de la línea, se genera inventario en espera de ser procesado, lo que acarrea un nuevo costo en el modelo, el cual debe evaluarse qué tan factible es asumirlo.

Con esta nueva adición al modelo, se lograría representar de forma más realista la realidad de una línea de producción, así como la flexibilidad que posee para cumplir con los pedidos de los clientes, por lo que se recomienda ahondar en el tema y analizar los resultados que genere el modelo.

AGRADECIMIENTOS

Los autores desean agradecer a la Universidad de Costa Rica por ser la institución que hospeda y soporta el proyecto de investigación.

CONFLICTO DE INTERESES

Los autores declaran no tener algún conflicto de interés.

REFERENCIAS

- [1] B. L. Maccarthy and J. Liu, "Addressing the gap in scheduling research: A review of optimization and heuristic methods in production scheduling," *International Journal of Production Research*, vol. 31, no. 1, pp. 59–79, 1993
- [2] Z. Dragecevic, E. Vujasinovic, A. Hursa, I. Jimenez-perez, and M. Gil-calvo, "Mathematical model for production sequencing in a manufacturing company," 2019
- [3] J. C. P. Rodríguez, "Desarrollo de un algoritmo Heurístico para la calendarización del sistema N/M/G/Cmax mediante la asignación de grupos de reglas de despacho," Ph.D. dissertation, Tecnológico de Monterrey, 2003
- [4] E. S. Hornig, "Heurística GRASP para la minimización del makespan en máquinas paralelas no relacionadas con tiempos de preparación dependientes de la secuencia," vol. 25, pp. 524–534, 2017
- [5] D. Rossit, M. Frutos, F. Tohme, D. Bros, and D. Rossit, "Modelos mixtos enteros de subloteo para Producción flow-shop minimizando el makespan. Área temática: Gestión de Operaciones y Logística," no. 1, pp. 2–5, 2014
- [6] B. Alidaee, H. Li, H. Wang, and K. Womer, "Integer programming formulations in sequencing with total earliness and tardiness penalties, arbitrary due dates, and no idle time: A concise review and extension," *Omega*, vol. 103, p. 102446, 2021. [Online]. Available: <https://doi.org/10.1016/j.omega.2021.102446>
- [7] K. R. Baker and G. D. Scudder, "Sequencing with Earliness and Tardiness Penalties: A Review," no. August 2015, 1990
- [8] T. Keshavarz, M. Savelsbergh, and N. Salmasi, "A branch-and-bound algorithm for the single machine sequence-dependent group scheduling problem with earliness and tardiness penalties," vol. 39, pp. 6410–6424, 2015.
- [9] K. R. Baker, "Minimizing earliness and tardiness costs in stochastic scheduling," *European Journal of Operational Research*, vol. 236, no. 2, pp. 445–452, 2014. [Online]. Available: <http://dx.doi.org/10.1016/j.ejor.2013.12.011>
- [10] A. Khare and S. Agrawal, "Scheduling hybrid flowshop with sequence-dependent setup times and due windows to minimize total weighted earliness and tardiness," *Computers Industrial Engineering*, 2019. [Online]. Available: <https://doi.org/10.1016/j.cie.2019.06.057>
- [11] S. Thevenin, N. Zufferey, M. Widmer, S. Thevenin, N. Zufferey, M. Widmer, S. Thevenin, and N. Zufferey, "Tabu search for a single machine scheduling problem with rejected jobs , setups and deadlines T," 9 th International Conference of Modeling, Optimization and Simulation, 2012. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00728646>
- [12] H. Lødding and A. Piontek, "The surprising effectiveness of earliest operation due-date sequencing," *Production Planning and Control*, vol. 28, no. 5, pp. 459–471, 2017. [Online]. Available: <http://dx.doi.org/10.1080/09537287.2017.1302616>
- [13] I. M. Al-Harkan, "Algorithms for Sequencing and Scheduling," On merging sequencing and scheduling theory with genetic algorithms to solve stochastic job shops, p. 438, 2010.
- [14] S. Verma, "Single-Machine Scheduling of Unit-Time Jobs with Earliness and Tardiness Penalties 1 Introduction," pp. 1–16, 2000
- [15] C. Akjiratikarl, "Branch and Bound Approach for Single-Machine Sequencing with Early / Tardy Penalties and Sequence-Dependent Setup Cost," vol. 3, no. 2, pp. 100–115, 2004.
- [16] R. Hassin and M. Shani, "Machine scheduling with earliness, tardiness and non-execution penalties," vol. 32, pp. 683–705, 2005.
- [17] S. Thevenin, N. Zufferey, and M. Widmer, "Metaheuristics for a scheduling problem with rejection and tardiness penalties," pp. 89–105, 2015.
- [18] J. M. S. Valente and R. A. F. S. Alves, "Heuristics for the single machine scheduling problem with quadratic earliness and tardiness penalties," vol. 35, pp. 3696–3713, 2008.
- [19] S. Thevenin, N. Zufferey, and M. Widmer, "Order acceptance and scheduling with earliness and tardiness penalties," *Journal of Heuristics*, 2016.
- [20] R. Tavakkoli-moghaddam and G. Moslehi, "A branch-and-bound algorithm for a single machine sequencing to minimize the sum of maximum earliness and tardiness with idle insert," vol. 174, pp. 388–408, 2006.
- [21] Z. Ales, P. Yves, P. Michelon, and B. F. Rosa, Algorithms for job scheduling problems with distinct time windows and general earliness/tardiness penalties, 2016.
- [22] R. D. Souza, M. J. F. Souza, and M. Gomes, "An Adaptive Genetic Algorithm to Solve the Single Machine Scheduling Problem with Earliness and Tardiness Penalties," 2010.
- [23] G. Steiner and R. Zhang, "Revised Delivery-Time Quotation in Scheduling with Tardiness Penalties Revised Delivery-Time Quotation in Scheduling with Tardiness Penalties," no. July 2014, 2011.

- [24] P. Taylor, C. Koulamas, and C. Koulamas, “times and job-dependent tardiness penalties A note on the scheduling problem with revised delivery times and job-dependent tardiness penalties,” no. December, pp. 37–41, 2014.
- [25] J. Paulo, D. C. M. N. Jos^é, E. C. Arroyo, H. Mauricio, and M. V. Luciana, “Hybrid GRASP Heuristics to Solve an Unrelated Parallel Machine Scheduling Problem with Earliness and Tardiness Penalties,” *Electronic Notes in Theoretical Computer Science*, vol. 302, pp. 53–72, 2014. [Online]. Available: <http://dx.doi.org/10.1016/j.entcs.2014.01.02>.
- [26] L. Zhang and L. Lu, “Parallel-machine scheduling with release dates and rejection,” *4OR*, 2016.
- [27] A. Glazer, R. Hassin, and L. Ravner, “A strategic model of job arrivals to a single machine with earliness and tardiness penalties A strategic model of job arrivals to a single machine with earliness,” vol. 5854, no. October, pp. 0–29, 2017.
- [28] J. Schaller, “Single machine scheduling with early and quadratic tardy penalties,” *Computers and Industrial Engineering*, vol. 46, no. 3, pp. 511–532, 2004.
- [29] J. Elias, C. Arroyo, and S. Ottoni, “Multi-objective Variable Neighborhood Search Algorithms for a Single Machine Scheduling Problem with Distinct due Windows,” *Electronic Notes in Theoretical Computer Science*, vol. 281, pp. 5–19, 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.entcs.2011.11.022>.
- [30] M. Ruano and C. Rovira, “Teorema del límite central,” 2019.