

Optimización de la tardanza total en máquinas paralelas: caso de estudio en la industria del café.



Revista EIA
ISSN 1794-1237
e-ISSN 2463-0950
Año XIX/ Volumen 21/ Edición N.41
Enero - junio de 2024
Reia4102 pp. 1-20

Publicación científica semestral
Universidad EIA, Envigado, Colombia

Santiago López^{z1}
Alejandro Montoya¹
Juan Pablo Mesa¹
✉ Alejandro Uribe¹

1. Universidad EAFIT

PARA CITAR ESTE ARTÍCULO / TO REFERENCE THIS ARTICLE /

López, S.; Montoya, A.; Mesa, J.P. y Uribe, A.
Optimización de la tardanza total en máquinas paralelas: caso de estudio en la industria del café.
Revista EIA, 21(41), Reia4102.
pp. 1-20.
<https://doi.org/10.24050/reia.v21i41.1710>

✉ *Autor de correspondencia:*

Uribe, A
Maestría en Ingeniería
Universidad EAFIT

Correo electrónico:
auribev1@eafit.edu.co

Recibido: 07-06-2023
Aceptado: 09-11-2023
Disponible online: 01-01-2024

Resumen

La programación de producción es una de las tareas con mayor importancia en sistemas de manufactura ya que permite la asignación de trabajos con cierto número de recursos disponibles. En la industria de manufactura y de servicios, poder cumplir con los compromisos acordados con los diferentes clientes y con los plazos de producción puede generar mayores niveles de satisfacción y ventajas competitivas en el mercado. Por otra parte, una de las principales industrias a nivel mundial es la cafetera. El café es la segunda materia prima más negociada del mundo. Por lo tanto, en esta investigación se aborda la optimización de la tardanza total en la programación de trabajos en máquinas paralelas para un caso de estudio en una tostadora de café. Este es un problema NP-hard donde cada trabajo requiere del proceso de tostión y, además, cuenta con una fecha de entrega asociada. El tiempo de procesamiento de cada trabajo depende de la máquina que se designe y no todas las máquinas puede ejecutar todos los trabajos. Se presenta el modelo matemático para describir el problema mediante programación lineal entera mixta y se implementa en Python. Adicionalmente, se propone un modelo mate heurístico basado en la heurística ATC que permite reducir el tiempo de cómputo y obtener soluciones cercanas al óptimo. Este modelo fue evaluado con mil instancias de datos reales del proceso de tostión. Para la solución del problema se utilizó el optimizador Gurobi. El método exacto muestra altos tiempos de cómputo por lo que se propuso realizar primero la secuenciación de trabajos para luego asignarlos a las maquinas por medio de un modelo matemático simplificado. Al utilizar el modelo mate heurístico propuesto se logra reducir en promedio un 53.98% el tiempo de ejecución y con respecto a la tardanza de la solución, este modelo logra una mediana del 0.0% con respecto al optimo.

Palabras clave: Programación, Optimización, Máquinas paralelas no relacionadas, Retraso total, Modelo matemático, Heurística, Mateheurística, ATC, Café, Tostión.

Total tardiness optimization on parallel machines: case study in the coffee industry.

Abstract

Production scheduling is one of the most important tasks in manufacturing systems since it allows the allocation of jobs with a certain number of available resources. In the production and service industries, meeting the established commitments with the different customers and production deadlines can lead to higher levels of satisfaction and market competitive advantages. On the other hand, one of the main industries worldwide is the coffee industry. Indeed, coffee is the second most traded commodity in the world. Therefore, this research addresses the optimization of the total tardiness in the scheduling of jobs in parallel machines for a case study in a coffee roaster. This is an NP-hard problem where each job requires the roasting process and, in addition, has an associated delivery time. Each job's processing time depends on the machine that is designated and not all machines can run all jobs. We present the mathematical model to describe the problem using mixed integer linear programming and then it is implemented in Python. Additionally, a mate heuristic model based on the ATC heuristic is proposed in order to reduce the computation time and to reach solutions close to the optimum. This model was evaluated with one thousand instances of real data from the roasting process. The Gurobi optimizer was used to solve the problem. The exact method shows high computational times, so we proposed to first perform the sequencing of jobs and then assign them to the machines by means of a simplified mathematical model. By using the proposed mate heuristic model, an average reduction of 53.98% of the execution time is achieved and with respect to the solution delay, this model achieves a median of 0.0% with respect to the optimum.

Keywords: Scheduling, Optimization, Unrelated parallel machines, Total tardiness, Mathematical model, Heuristics, Mateheuristics, ATC, Coffee, Roasting.

1. Introducción

La programación de la producción es de gran importancia para la competitividad en la industria manufacturera y de servicios (B. Wang, Feng y Wang, 2023). Ser capaz de cumplir con los compromisos que se acuerden con los clientes y con los plazos de producción puede generar ventajas ante la competencia y satisfacción en los clientes (Zhao et al., 2021). Por otra parte, la industria cafetera es una de las principales a nivel mundial debido a que el café es la segunda materia prima más negociada del mundo luego del petróleo dada su gran popularidad (Costa et al., 2014; Ratanasanya et al., 2022). Es por esta razón que este trabajo trata sobre la programación de trabajos en máquinas paralelas para cumplir las demandas del proceso de tuestión del café. Este problema puede ser definido como N trabajos que deben de ser programados en M máquinas paralelas no relacionadas entre sí (Lenstra y Vakhania, 2023). Para el caso de estudio, cada trabajo requiere solamente del proceso de tuestión, además, la ejecución de los trabajos puede estar restringida en algunas máquinas. Por otra parte, cada máquina solo puede procesar un trabajo al tiempo y cada trabajo tiene una fecha de entrega.

El objetivo es encontrar una programación de los trabajos en donde se minimice la tardanza total. En la literatura esta clase de problema de programación es NP-hard y por esta razón se propone un efectivo modelo mate heurístico basado en la heurística ATC para obtener soluciones cercanas al óptimo con bajo tiempo de ejecución. La heurística ATC se usa para generar la secuencia de trabajos y luego realizar la asignación de estos mismos en las máquinas por medio de un modelo simplificado de programación entera mixta.

Este documento se organiza de la siguiente manera. En la sección 2 presenta la revisión de la literatura con respecto al problema y a los métodos propuestos para su solución. En la sección 3 se aborda la descripción del problema. La sección 4 aborda el modelo matemático que describe el problema. La sección 5 muestra el modelo matemático simplificado junto con la heurística ATC. Por último, en las secciones 6 y 7 se reportan los resultados de los experimentos y se mencionan las conclusiones respectivamente.

2. Revisión de la literatura

La programación de la producción o *scheduling*, es una de las tareas más relevantes realizadas en los sistemas de manufactura, responsable de la asignación de una serie de trabajos en unos recursos disponibles (Ruiz, Carlos García-Díaz y Maroto, 2007). El problema de programación de trabajos en máquinas paralelas no relacionadas ha sido altamente estudiado por muchos autores (Berthier et al., 2022; Wang S. et al., 2023; H. Wang, Li & Gong, 2023). Algunos de ellos han incluido en sus algoritmos los tiempos de alistamiento (Fang, Zhu y Mei, 2022; Heydar, Mardaneh y Loxton, 2022; Li et al., 2011; Sanati, Moslehi y Reisi-Nafchi, 2023; Vincent et al., 2016). La programación de trabajos en máquinas paralelas resulta ser útil en procesos de producción. Por ejemplo, el trabajo de Li et al. (2011) trata la programación de trabajos en máquinas paralelas minimizando la tardanza total. La heurística utilizada en esta investigación está basada en una regla de dominancia proporcionada por Azizoglu y Kirca (1998) que permite la reducción del número de trabajos a programar y adicionalmente, un método de recocido simulado para mejorar la solución durante la construcción del cronograma. El rendimiento del método propuesto se compara con el branch-and-bound de Yalaoui y Chu (2002) y con la relajación lagrangiana de Tanaka and Araki (2008). Se encontró que la heurística propuesta llega a soluciones cercanas al óptimo en un tiempo de cómputo mucho menor a los métodos con los que se compara.

Sanati et al. (2023) aborda el problema de programación de máquinas paralelas no relacionadas entre sí, con tiempos de alistamiento dependientes de la secuencia y diferentes tarifas de consumo de energía. Se consideran dos tiempos de alistamiento en este artículo: incluidos y no incluidos en el tiempo de procesamiento. Para cada uno de estos, se formuló un modelo de programación lineal entera mixta. Los resultados muestran que para el modelo que no incluye el tiempo de alistamiento en el tiempo de procesamiento, se encuentran soluciones de manera óptima con instancias de hasta 16 máquinas y 40 trabajos. Por otra parte, el modelo que si incluye el tiempo de alistamiento puede resolver el problema considerando hasta 20 máquinas y 20 trabajos. Adicionalmente, en este artículo implementaron un algoritmo heurístico de corrección y relajación

para resolver instancias a gran escala. Este fue desarrollado con base a la transformación del entorno de máquinas paralelas no relacionadas en algunos problemas de programación de una sola máquina. La heurística desarrollada para cada problema abordado resolvió desde 20 hasta 100 trabajos.

Bektur y Saraç (2019), estudia el problema de programación de trabajos en máquinas paralelas no relacionadas con tiempos de alistamiento dependientes de la secuencia. La función objetivo de este artículo busca minimizar la tardanza total ponderada. Los autores proponen un modelo de programación lineal entera mixta. Debido a la complejidad del problema, proponen algoritmos de búsqueda tabú (TS) y recocido simulado (SA) para reducir tiempos de ejecución. Los resultados obtenidos afirman que el algoritmo TS con memoria a largo plazo y el mecanismo de búsqueda de soluciones iniciales basado en la regla ATCS se desempeña mejor que los otros métodos heurísticos utilizados para comparar.

El trabajo de Diana, de Souza y Filho (2018), aborda el problema de minimización de tardanzas ponderadas totales en máquinas paralelas no relacionadas con tiempos de alistamiento dependientes de la secuencia y tiempos de operación del trabajo. Este problema consiste en programar un conjunto de trabajos reduciendo los costos de penalización causados por los retrasos en las fechas de vencimiento de los trabajos. Para solucionar esto, los autores proponen una metaheurística híbrida ILS-VND, donde se integra una heurística de búsqueda local Variable Neighborhood Descent (VND) y la metaheurística Iterated Local Search (ILS) con múltiples reinicios. El método propuesto ILS-VND supera en la mayoría de los casos los metaheurísticos IHM y ACO. Por último, el artículo de Monch (2008), presenta un método eficiente para minimizar la tardanza total ponderada (TWT) en la programación de máquinas paralelas no relacionadas. El método propuesto usa la heurística de colonia de hormigas artificiales para construir soluciones iterativas del problema de programación. Por otra parte, para el cálculo de la información en la heurística, utilizan la regla de despacho Costo Aparente de Tardanza (ATC).

En esta investigación no es factible resolver el problema por medio del modelo matemático exacto debido al alto tiempo de cómputo requerido. En la literatura esta clase de problema de programación es NP-hard, por lo tanto, muchos autores proponen heurísticas, meta y mate heurísticas para resolverlo. Este documento es novedoso debido a que aborda el problema en dos partes: secuenciación de los trabajos y asignación de estos mismos en máquinas paralelas por medio un modelo matemático simplificado.

3. Descripción del problema

En el proceso productivo de tosti3n, se cuenta con 7 tostadores y diferentes velocidades para suplir la demanda de dos procesos: empaque molido y extracci3n. La ejecuci3n de los trabajos puede estar restringida en algunos tostadores, es decir, no todas las m3quinas pueden ejecutar todos los trabajos. El flujo de los procesos de empaque molido y de las plantas de extracci3n se pueden ver en las Figuras 1 y 2.

El proceso de empaque molido empieza con el enfriamiento del caf3 tostado, luego este se muele e inicia la etapa de desgasificaci3n para que al final pueda ser empackado. El proceso de extracci3n inicia con la desgasificaci3n del caf3 tostado, luego se procede con la molienda y finalmente se extraen los s3lidos solubles del caf3. En ambos casos, el tiempo de cada etapa depende de la referencia con la que se est3 trabajando.

Por otra parte, en el proceso de empaque de caf3 molido se cuenta con 7 m3quinas de empaque, las cuales tienen diferentes velocidades seg3n la referencia. Los procesos de enfriamiento y desgasificaci3n se realizan en silos y tienen igual capacidad por m3quina de empaque.

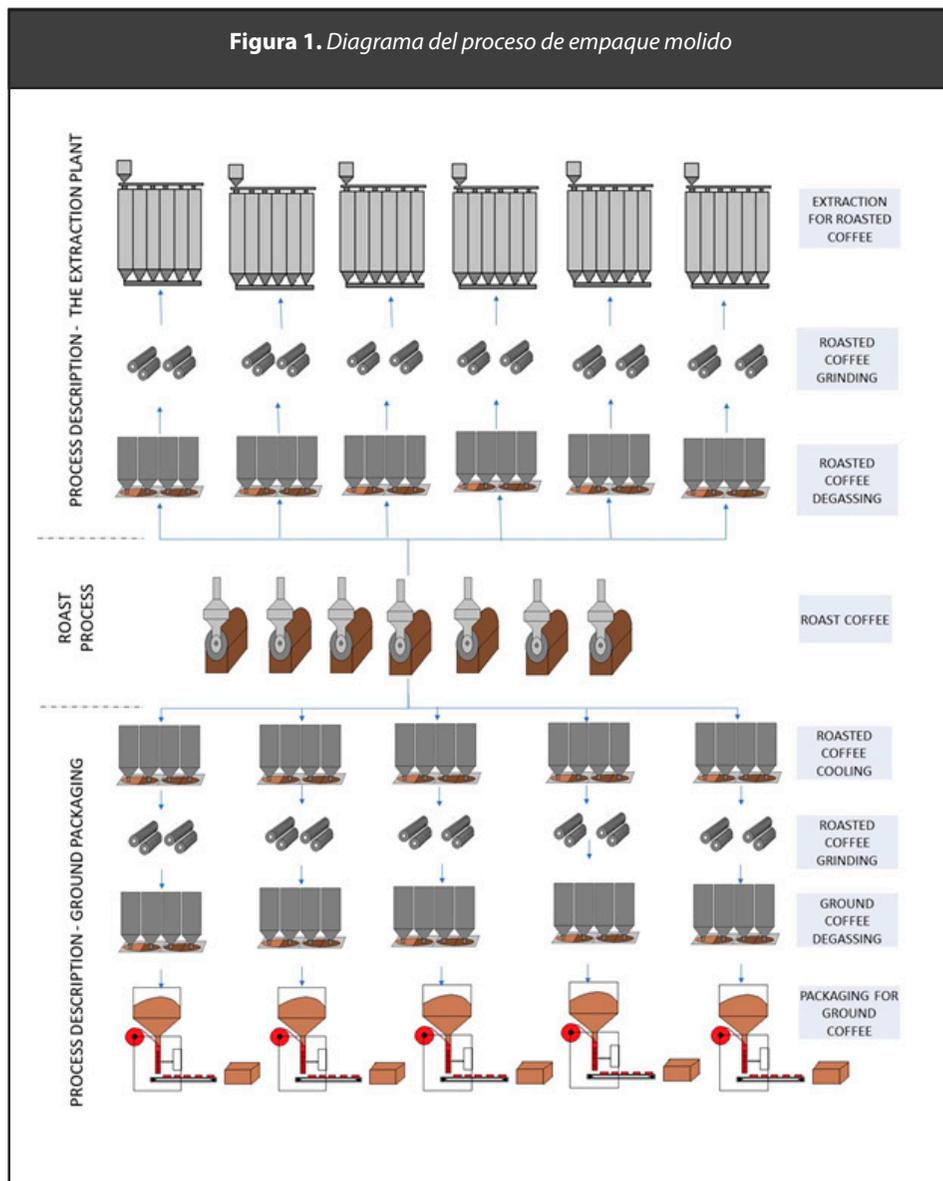


Figura 2. Diagrama del proceso de extracción

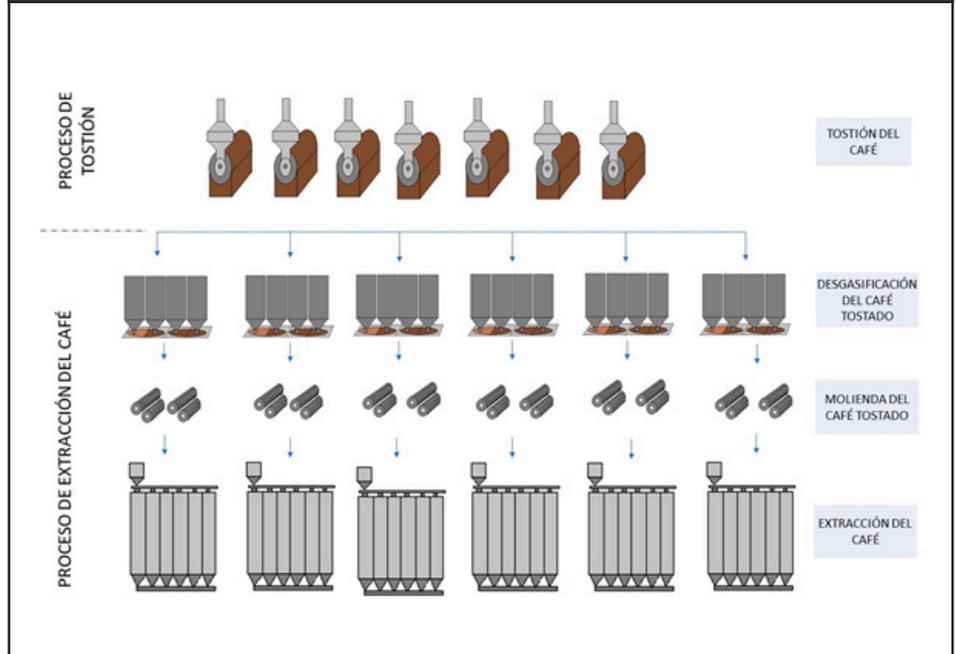
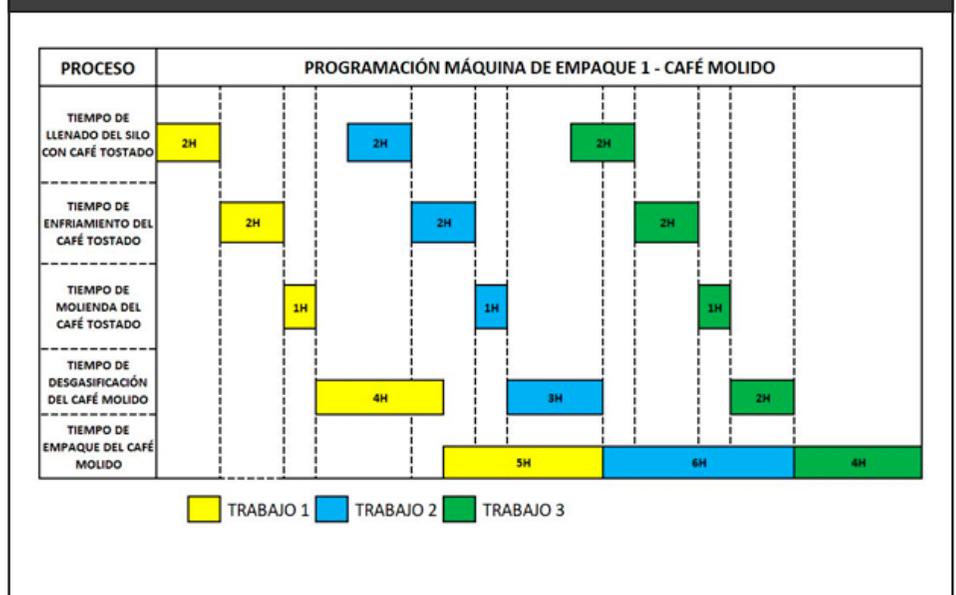
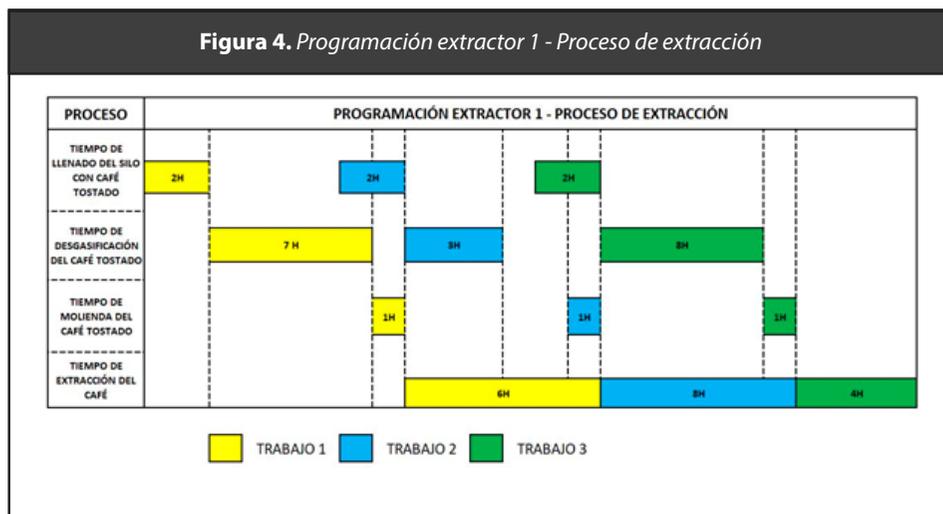


Figura 3. Programación máquina de empaque 1 - Café molido



En la Figura 3 se puede ver la programación de 3 trabajos que ocupan 15 horas de empaque continuo para la máquina 1. Cada trabajo tiene un tiempo máximo de inicio; es decir, los trabajos 1, 2 y 3 deben empezar 9, 8 y 7 horas antes de ser empacados. Si alguno se demora una hora más para producir, la máquina de empaque estará parada durante este tiempo.

Para el proceso de extracción se tienen 6 plantas diferentes. Además, se cuenta con diferentes velocidades, tiempos de molienda y desgasificación (dependiendo de la referencia con la que se esté trabajando). Es importante resaltar que los silos son el lugar donde el café tostado desgasifica.



En la Figura 4, se puede ver la programación de 3 trabajos que ocupan 18 horas de la planta de extracción 1. Cada trabajo tiene un tiempo máximo de inicio; es decir, los trabajos 1, 2 y 3 deben empezar 10, 6 y 11 horas antes de que la extracción inicie. De igual manera, si alguno se demora una hora más para producir, la planta de extracción estará parada por este mismo tiempo.

Si se combinan las demandas de los dos procesos (7 máquinas de empaque y 6 plantas de extracción), se obtiene una larga lista de trabajos que deben de ser ejecutados en el proceso de tostión con un tiempo máximo de cumplimiento de cada pedido. En la Tabla 1 se puede observar que, dependiendo del tostador que ejecute cada trabajo, se puede o no generar tardanza.

Tabla 1. Horas de procesamiento de cada trabajo por tostador

Trabajo	Cliente	Kg de cada trabajo	SKU	Tiempo entrega (hora)	T1	T2	T3	T4	T5	T6	T7
1	GP	3500	A	0	2	4	4	inf	2	3	1
2	GP	3500	A	0	2	4	4	inf	2	3	1
3	EP	1000	B	0	0,6	1,1	1,1	1,1	0,6	0,9	inf
4	GP	3300	A	1	1,9	3,8	3,8	inf	1,9	2,8	0,9
5	EP	500	C	1	0,3	0,6	0,6	0,6	0,3	0,4	inf
6	GP	2500	F	3	1,4	2,9	2,9	inf	1,4	2,1	0,7
7	EP	3000	D	3	1,7	3,4	3,4	3,4	1,7	2,6	inf
8	GP	3500	G	4	2	4	4	inf	2	3	1
9	GP	3500	G	4	2	4	4	inf	2	3	1
10	GP	2000	G	4	1,1	2,3	2,3	inf	1,1	1,7	0,6
11	EP	2500	D	6	1,4	2,9	2,9	2,9	1,4	2,1	inf
12	EP	3000	E	6	1,7	3,4	3,4	3,4	1,7	2,6	inf

En esta investigación se aborda la optimización de la tardanza total en la programación de trabajos en máquinas paralelas para poder cumplir las demandas de los procesos clientes. Este problema puede ser definido como N trabajos que deben de ser programados en M máquinas paralelas no relacionadas entre sí. El parámetro binario $h_{i,k}$ es igual a 1 si el trabajo $i \in N$ puede ser hecho por la máquina $k \in M$, 0 de otra forma. Por otra parte, cada máquina puede procesar un solo trabajo al tiempo. Cada trabajo $i \in N$ tiene un tiempo de procesamiento p_i y una fecha de entrega d_i . La variable de decisión $c_{i,k}$ es usada para definir el tiempo de finalización del trabajo $i \in N$ en la máquina $k \in M$. El trabajo $i \in N$ está retrasado si el tiempo de finalización $c_{i,k}$ es más grande que la fecha de entrega d_i y la tardanza t_i es medida por $c_{i,k} - d_i$. La tardanza de cada trabajo completado a tiempo es igual a 0, por lo tanto, la tardanza t_i de cada trabajo i es definida como el $\max(c_{i,k}) - d_i$, 0). El objetivo es encontrar una programación de los trabajos donde se minimice la tardanza total. Además, una solución factible para este problema debe de cumplir lo siguiente: cada trabajo puede ser

procesado al menos por una máquina; la máquina asignada por cada trabajo debe de cumplir con el parámetro $h_{i,k}=1$ y todos los trabajos deben ser programados.

4. Modelo matemático

A continuación, se presentan los conjuntos, parámetros, variables de decisión, función objetivo y restricciones.

4.1. Conjuntos

- N trabajos
- M máquinas.

El conjunto N tiene un trabajo ficticio denotado por N_0 . Las variables X_{0jk} y X_{i0k} son usadas para especificar que trabajos i y j serán procesados de primero y de último en la máquina k . Los tiempos de procesamiento y tiempos de entrega asociados al trabajo ficticio son iguales a cero ($p_{0k} = 0, d_0 = 0$).

4.2. Parámetros

- $p_{i,k}$: tiempo de procesamiento del trabajo $i \in N$, en la maquina $k \in M$.
- d_i : tiempo de entrega del trabajo $i \in N$.
- $h_{i,k}$: 1 si el trabajo $i \in N$ puede ser procesada en la maquina $k \in M$; 0 de otra manera.

4.3. Variables de decisión

- $x_{i,j,k}$: 1 si $i \in N$ es programado antes del trabajo $j \in N$, en la maquina $k \in M$; 0 de otra manera.
- t_i : tardanza del trabajo $i \in N$.
- $c_{i,k}$: tiempo de terminación del trabajo $i \in N$, en la maquina $k \in M$.

4.4. Función objetivo

$$\text{Min}(Z) = \sum_{i \in N} t_i \quad (1)$$

4.5. Sujeto a

$$\sum_{k \in M} \sum_{j \in N, i \neq j} x_{i,j,k} = 1, \forall i \in N \quad (2)$$

$$\sum_{k \in M} \sum_{i \in N, i \neq j} x_{i,j,k} = 1, \forall j \in N \quad (3)$$

$$\sum_{j \in N, i \neq j} x_{i,j,k} - \sum_{u \in N, i \neq u} x_{u,i,k} = 0, \forall i \in N, \forall k \in M \quad (4)$$

$$\sum_{j \in N} x_{0,j,k} \leq 1, \forall k \in M \quad (5)$$

$$\sum_{j \in N, i \neq j} x_{i,j,k} \leq h_{i,k}, \forall i \in N, \forall k \in M \quad (6)$$

$$t_i \leq c_{i,k} - d_i, \forall i \in N, \forall j \in N, \forall k \in M, i \neq j \quad (7)$$

$$c_{i,k} \geq p_{i,k}, \forall i \in N_0, \forall k \in M \quad (8)$$

$$c_{j,k} \geq c_{i,k} + p_{j,k} + w(x_{i,j,k} - 1), \forall i \in N, \forall j \in N, \forall k \in M, i \neq j \quad (9)$$

$$x_{i,j,k} \in \{0,1\}, \forall i \in N, \forall j \in N, \forall k \in M \quad (10)$$

$$t_i \geq 0, \forall i \in N \quad (11)$$

$$c_{i,k} \geq 0, \forall i \in N, \forall k \in M \quad (12)$$

La expresión (1) representa la minimización de la tardanza total. La expresión (2) garantiza que cada trabajo $i \in N$ tiene un trabajo $j \in N_0$ que le sigue. La expresión (3) garantiza que cada trabajo $j \in N$ es precedido por otro trabajo $i \in N_0$. La expresión (4) asegura que si un trabajo $i \in N$ es programado en la máquina $k \in M$, entonces tiene un predecesor y un sucesor $j \in N_0$ en una máquina $k \in M$. La expresión (5) asegura que solo un trabajo es programado de primero en cada máquina $k \in M$. La expresión (6) asegura si el trabajo $i \in N$ puede ser procesado en la máquina $k \in M$. La expresión (7) define la tardanza de cada trabajo $i \in N$. Por último, las expresiones (8) y (9) definen C_{ik} como el tiempo de finalización del trabajo $i \in N$, en la máquina $k \in M$, donde w es un número convenientemente grande.

5. Modelo mate heurístico

Además del modelo matemático explicado en la sección anterior, se propone un método aproximado para resolver el problema. Este consta de dos fases: secuenciación de los trabajos y asignación de los trabajos en las máquinas. Para la fase de secuenciación se utilizan 3 reglas heurísticas: la regla de despacho ATC, la secuenciación aleatoria y la regla EDD (earliest due date) que consiste en ordenar los trabajos de menor a mayor tiempo de entrega. Para la fase de asignación se utiliza un modelo matemático que parte de una secuenciación dada. Considerando que la secuenciación aleatoria y la regla EDD no presentan mayor complejidad, solo abordaremos el método ATC en la sección 5.1. Luego, explicaremos el modelo matemático de asignación en la sección 5.2.

5.1. Heurística ATC

La heurística ATC es una regla de despacho compuesta que tiende a programar primero los trabajos con índices de mayor prioridad. Esta se da por la combinación de otras dos reglas: holgura mínima (tiempo hasta la fecha de vencimiento) y tiempo de procesamiento más corto ponderado (Monch, 2008).

$$I_i(t) = w_i/p_i \exp\left(-\frac{(d_i - p_i - t)^+}{k\bar{p}}\right) \quad (13)$$

Donde:

- t : Tiempo para la toma de decisión.
- d_i : Tiempo de entrega del trabajo i .
- p_i : Tiempo de procesamiento del trabajo i .
- \bar{p} : Tiempo promedio de procesamiento de los trabajos restantes.
- k : Número constante.

5.2. Modelo matemático simplificado

5.2.1. Conjuntos

- N trabajos.
- M máquinas.

5.2.2. Parámetros

- $p_{i,k}$: tiempo de procesamiento del trabajo $i \in N$, en la máquina $k \in M$.
- d_i : tiempo de entrega del trabajo $i \in N$.
- $h_{i,k}$: 1 si el trabajo $i \in N$ puede ser procesado en la máquina $k \in M$; 0 de otra manera.

5.2.3. Variables de decisión

- $x_{i,j,k}$: 1 si $i \in N$ es programado en la máquina $k \in M$; 0 de otra manera.
- t_i : tardanza del trabajo $i \in N$.
- $c_{i,k}$: tiempo de terminación del trabajo $i \in N$, en la máquina $k \in M$.

5.2.4. Función objetivo

$$\text{Min}(Z) = \sum_{i \in N} t_i \quad (14)$$

5.2.5. Sujeto a

$$\sum_{k \in M} x_{i,j,k} = 1, \forall i \in N \quad (15)$$

$$x_{i,k} \leq h_{i,k}, \forall i \in N, \forall k \in M \quad (16)$$

$$t_0 \geq (c_{0,k} - d_0) + w(x_{0,k} - 1), \forall k \in M \quad (17)$$

$$t_i \geq c_{i,k} - d_i + w(x_{i,k} - 1), \forall i \in N, \forall k \in M, i \neq 0 \quad (18)$$

$$c_{i,k} = \sum_{i \in N} p_{i,k} x_{i,k}, \forall k \in M \quad (19)$$

$$x_{i,j,k} \in \{0,1\}, \forall i \in N, \forall j \in N, \forall k \in M \quad (20)$$

$$t_i \geq 0, \forall i \in N \quad (21)$$

$$c_{i,k} \geq 0, \forall i \in N, \forall k \in M \quad (22)$$

La expresión (14) representa la minimización de la tardanza total. La expresión (15) garantiza que cada trabajo $i \in N$ es programado. La expresión (16) asegura si el trabajo $i \in N$ puede ser programado en la máquina $k \in M$. Las expresiones (17) y (18) definen la tardanza de cada trabajo $i \in N$, donde w es un número convenientemente grande. Por último, la expresión (19) define c_{ik} como el tiempo de finalización del trabajo $i \in N$ en la máquina $k \in M$.

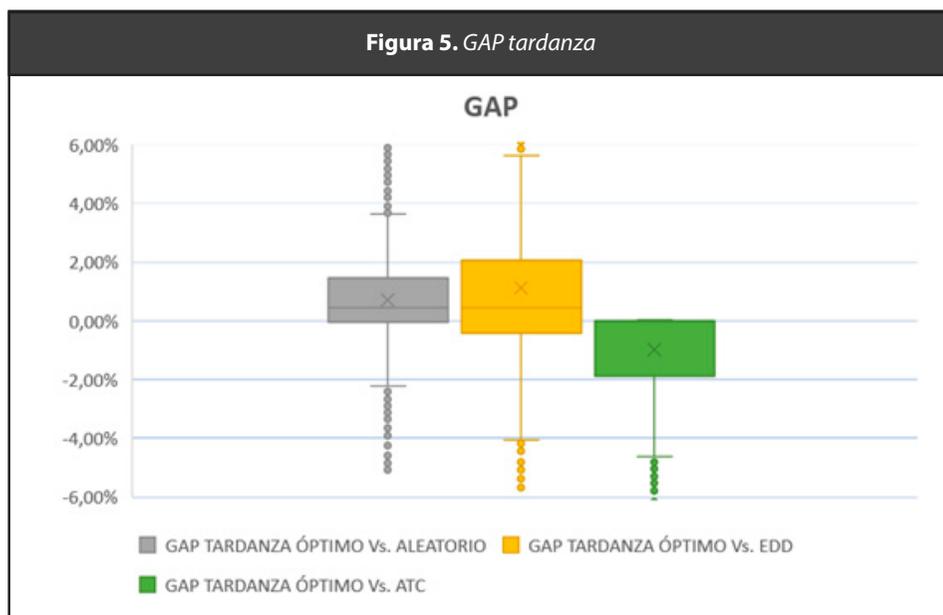
6. Resultados

En esta investigación se realizaron experimentos utilizando mil instancias con diferentes cantidades de trabajos, tomando valores aleatorios entre 9 y 21. Los parámetros de cada trabajo como el tiempo de procesamiento del trabajo p_{ik} con $i \in N$ en la máquina $k \in M$, el tiempo de entrega del trabajo d_j con $i \in N$ y el parámetro h_{ik} fueron obtenidos del proceso real.

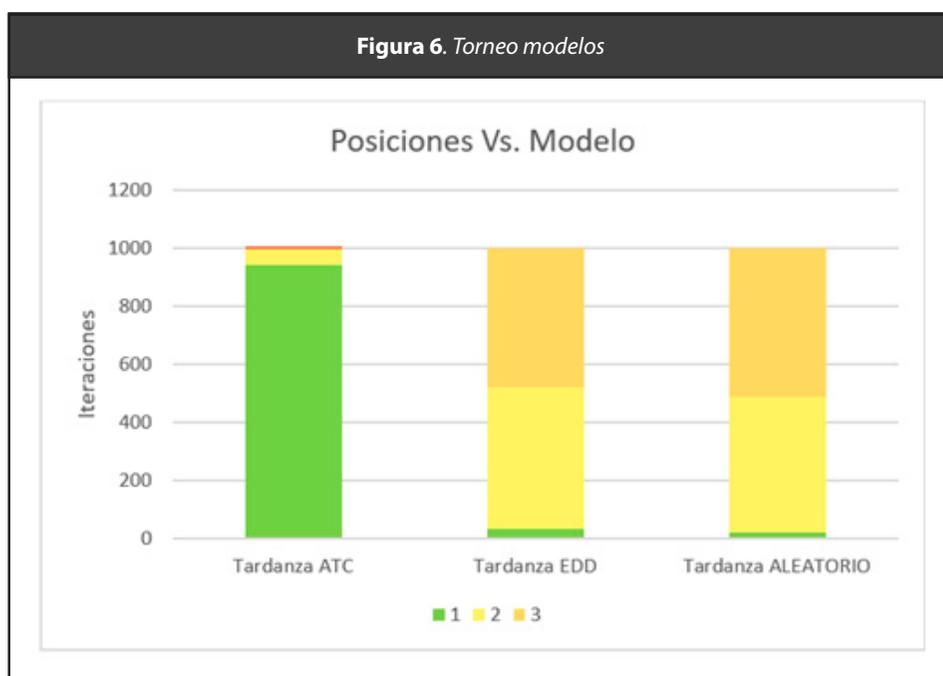
Las mil instancias se resolvieron con el método exacto explicado en la sección 4 y los modelos mate heurísticos explicados en la sección 5. Se estableció un tiempo máximo de computación de 1 hora por instancia para cada modelo. Para la solución de este problema se hizo un desarrollo en Python y se utilizó Gurobipy como solver. Como se puede apreciar en la Tabla 2 y en la Figura 5, la secuenciación por la heurística ATC es la que mejor GAP muestra con respecto al modelo óptimo; luego le siguen la regla EDD y el orden aleatorio.

Tabla 2. Estadísticos GAP tardanza óptimo vs. modelos

Métrica	ATC	Aleatorio	EDD
Promedio	-0,99%	0,69%	1,13%
Min	-6,72%	-5,09%	-5,67%
P 25%	-1,87%	-0,04%	-0,40%
Mediana	0,00%	0,46%	0,45%
P 75%	0,00%	1,44%	2,05%
Max	0,01%	12,91%	11,93%



En la Figura 6 se observa una clara preeminencia del enfoque de secuenciación ATC. Esta logra obtener la solución óptima más destacada en comparación con los demás modelos en el conjunto de mil instancias. A continuación, en términos de rendimiento, le sigue la aplicación de la regla EDD, dejando en último lugar el enfoque de secuenciación aleatoria.

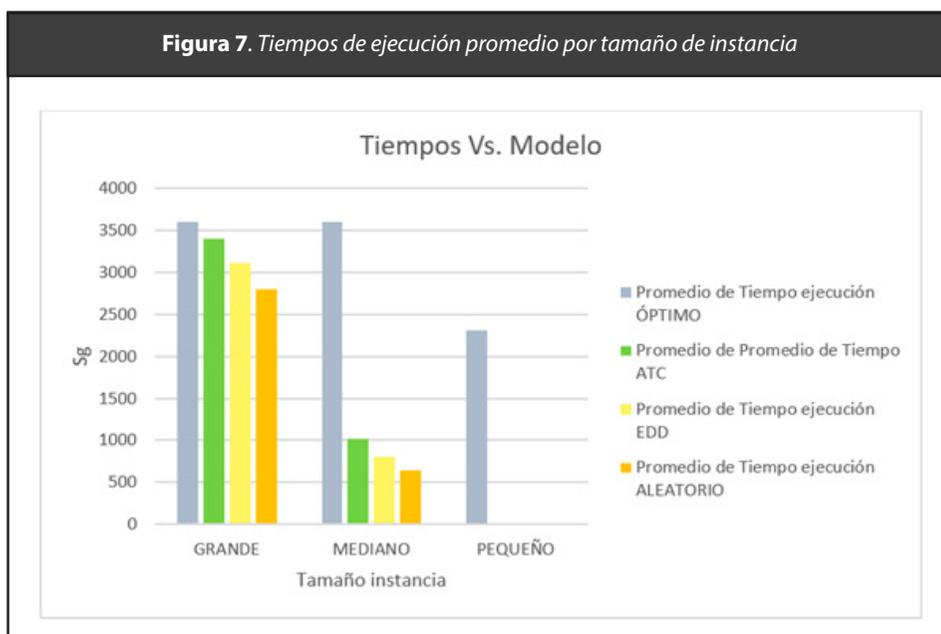


En la Tabla 3 se puede observar una reducción significativa del tiempo de cómputo para cualquiera de modelos propuestos. Para objeto de analisis, se categorizan los tamaños de las instancias en grande, entre 17 y 21 trabajos, mediano, entre 13 y 17 trabajos y pequeños, entre 9 y 13 trabajos. En la Figura 7 se observa que, en los tamaños medianos y pequeños, la reducción del tiempo promedio de cómputo es superior a los 2500 segundos por instancia y para los tamaños grandes la reducción es cercana a los 500 segundos por instancia.

Tabla 3. Estadísticos GAP tiempos óptimo vs. modelo

Métrica	ATC	Aleatorio	EDD
Promedio	-53,98%	-63,53%	-58,91%
Min	-99,93%	-99,94%	-99,92%
P 25%	-99,00%	-99,06%	-98,98%
Mediana	-84,54%	-93,73%	-90,62%
P 75%	0,02%	-0,05%	-0,01%
Max	0,33%	0,24%	0,45%

Figura 7. Tiempos de ejecución promedio por tamaño de instancia



7. Conclusiones

En esta investigación se aborda el problema de minimización de tardanza total en máquinas paralelas no relacionadas aplicado en una tostadora de café. Al ser este problema NP-hard, el método exacto explicado en la sección 4 no es viable dado el alto tiempo de cómputo. Es por esta razón que se propone dividir este problema en dos etapas: realizar la secuenciación de los trabajos y asignar los trabajos en las máquinas a través del modelo matemático simplificado explicado en la sección 5. Al aplicar la heurística ATC se logra encontrar un GAP con una mediana de la tardanza con respecto al óptimo de 0.0% y con respecto al tiempo de cómputo, se logra una reducción en promedio del 53.98%.

8. Referencias

- Azizoğlu, M.; & Kirca, Ö. (1998). Tardiness minimization on parallel machines. *International Journal of Production Economics*, 55(2), 163-168. [https://doi.org/10.1016/s0925-5273\(98\)00034-6](https://doi.org/10.1016/s0925-5273(98)00034-6).
- Bektur, G.; & Saraç, T. (2019). A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server. *Computers & Operations Research*, 103, 46-63. <https://doi.org/10.1016/j.cor.2018.10.010>
- Berthier, A.; Yalaoui, A.; Chehade, H.; Yalaoui, F.; Yalaoui, F.; & Bouillot, C. (2022). Unrelated parallel machines scheduling with dependent setup times in textile industry. *Computers & Industrial Engineering*, 174, 108736. <https://doi.org/10.1016/j.cie.2022.108736>
- Costa, A. S.; Alves, R. C.; Vinha, A. F.; Barreira, S. V. P.; Nunes, M. A.; Cunha, L. M.; & Oliveira, M. B. P. (2014). Optimization of antioxidants extraction from coffee silverskin, a roasting by-product, having in view a sustainable process. *Industrial Crops and Products*, 53, 350-357. <https://doi.org/10.1016/j.indcrop.2014.01.006>
- Diana, R. O. M.; De Souza, S. R.; & Filho, M. F. F. (2018). A variable neighborhood descent as ILS local search to the minimization of the total weighted tardiness on unrelated parallel machines and sequence dependent setup times. *Electronic Notes in Discrete Mathematics*, 66, 191-198. <https://doi.org/10.1016/j.endm.2018.03.025>
- Fang, W.; Zhu, H.; & Mei, Y. (2022). Hybrid meta-heuristics for the unrelated parallel machine scheduling problem with setup times. *Knowledge-Based Systems* 241:108193. <https://doi.org/10.1016/j.knosys.2022.108193>.
- Heydar, M.; Mardaneh, E.; & Loxton, R. (2022). Approximate dynamic programming for an energy-efficient parallel machine scheduling problem. *European Journal of Operational Research* 302(1):363–80. <https://doi.org/10.1016/j.ejor.2021.12.041>

- Lenstra, J. K.; & Vakhania, N. (2023). On the complexity of scheduling unrelated parallel machines with limited preemptions. *Operations Research Letters* 51(2):187–89. <https://doi.org/10.1016/j.orl.2023.02.004>.
- Li, K.; Shi, Y.; Yang, S.; & Cheng, B. (2011). Parallel machine scheduling problem to minimize the makespan with resource dependent processing times. *Applied Soft Computing* 11(8):5551–57. <https://doi.org/10.1016/j.asoc.2011.05.005>.
- Monch, L. (2008). Heuristics to Minimize Total Weighted Tardiness of Jobs on Unrelated Parallel Machines. *2008 IEEE International Conference on Automation Science and Engineering* 572–77. doi: 10.1109/COASE.2008.4626531.
- Ratanasanya, S.; Chindapan, N.; Polvichai, J.; Sirinaovakul, B.; & Devahastin, S. (2022). Model-based optimization of coffee roasting process: model development, prediction, optimization and application to upgrading of robusta coffee beans. *Journal of Food Engineering* 318:110888. doi: <https://doi.org/10.1016/j.jfoodeng.2021.110888>.
- Ruíz, R.; García-Díaz, J. C.; & Álvarez, C. M. (2007). Considering scheduling and preventive maintenance in the flowshop sequencing problem. *Computers & Operations Research* 34(11):3314–30. doi: <https://doi.org/10.1016/j.cor.2005.12.007>.
- Sanati, H.; Moslehi, G.; & Reisi-Nafchi, M. (2023). Unrelated parallel machine energy-efficient scheduling considering sequence-dependent setup times and time-of-use electricity tariffs. *EURO Journal on Computational Optimization* 11:100052. doi: <https://doi.org/10.1016/j.ejco.2022.100052>.
- Tanaka, S.; & Araki, M. (2008). A branch-and-bound algorithm with lagrangian relaxation to minimize total tardiness on identical parallel machines. *International Journal of Production Economics* 113(1):446–58. doi: <https://doi.org/10.1016/j.ijpe.2007.10.006>.
- Vincent, B. G.; Duhamel, C.; Ren, L.; & Tchernev, N. (2016). An efficient heuristic for scheduling on identical parallel machines to minimize total tardiness. *IFAC-PapersOnLine* 49(12):1737–42. <https://doi.org/10.1016/j.ifacol.2016.07.833>
- Wang, B.; Feng, K.; & Wang, X. (2023). Bi-objective scenario-guided swarm intelligent algorithms based on reinforcement learning for robust unrelated parallel machines scheduling with setup times. *Swarm and Evolutionary Computation* 80:101321. <https://doi.org/10.1016/j.swevo.2023.101321>
- Wang, H.; Li, R.; & Gong, W. (2023). Minimizing tardiness and makeSpan for distributed heterogeneous unrelated parallel machine scheduling by knowledge and Pareto-based memetic algorithm. *Egyptian Informatics Journal* 24(3):100383. <https://doi.org/10.1016/j.eij.2023.05.008>
- Wang, S.; Wu, R.; Chu, F.; & Yu, J. (2023). An exact decomposition method for unrelated parallel machine scheduling with order acceptance and setup times. *Computers & Industrial Engineering* 175:108899. <https://doi.org/10.1016/j.cie.2022.108899>
- Yalaoui, F.; & Chen, C. (2002). Parallel machine scheduling to minimize total tardiness. *International Journal of Production Economics* 76(3):265–79. [https://doi.org/10.1016/S0925-5273\(01\)00175-X](https://doi.org/10.1016/S0925-5273(01)00175-X)
- Zhao, Y.; Xu, X.; Xu, E.; & Niu, B. (2021). Stochastic customer order scheduling on heterogeneous parallel machines with resource allocation consideration. *Computers & Industrial Engineering* 160:107539. <https://doi.org/10.1016/j.cie.2021.107539>