

El problema de secuenciamiento parcialmente flexible, multiobjetivo, basado en un algoritmo genético de selección natural

Multi-objective partially flexible job-shop scheduling problem, based on a genetic natural selection algorithm

Samantha K. Jones Martínez¹, Alice M. Castillo Corrales¹, Justin A. Arce Obando^{1*}, Monserrat Álvarez González¹,
Daniel R. Rivera Leaver¹

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

Fecha de recepción: 1 de julio de 2022. **Fecha de aceptación:** 15 de enero de 2023.

***Autor de correspondencia:** justin.arceobando@ucr.ac.cr

Resumen. En el presente trabajo se buscó resolver un problema de secuenciación parcialmente flexible, energéticamente eficiente, con tiempos de preparación dependientes de la secuencia, penalizaciones de costos por finalización temprana y tardía y mantenimiento preventivo. Estas variables y objetivos fueron seleccionados porque se considera que estos son los que tienen mayor impacto en el día a día de las empresas. Para la solución se propone un algoritmo genético compatible con las variables propuestas y capaz de obtener una solución de manera eficiente. Para comprobar el funcionamiento de la propuesta, se prueba el algoritmo con una simulación que combina datos de otros trabajos con información ficticia. El algoritmo encontró una solución para las tres funciones objetivo propuestas a partir de la información de cinco generaciones de datos. El resultado fue un cronograma óptimo, la duración total y el costo de los envíos tardíos y los adelantos que no se pudieron evitar.

Palabras clave. Adelantos, algoritmo genético, atrasos, duración, eficiencia energética, JSSP, mantenimiento preventivo, PFJSP, SDST, taller de trabajo.

Abstract. In the present work, we seek to solve a partially flexible scheduling problem, energetically efficient, with sequence dependent setup time, cost penalties for both early and late completion, and preventive maintenance. These variables and objectives were selected because it is considered that these are the ones that have the greatest impact on the day-to-day of companies. For the solution, a genetic algorithm compatible with the proposed variables and capable of obtaining a solution efficiently is proposed. To check the operation of the proposal, the algorithm is tested with a simulation that combines data from other works with fictitious information. The algorithm found a solution for the three proposed objective functions from the information of five generations of data. The result was an optimal schedule, the total makespan and the cost of tardiness and earliness that could not be avoided.

Keywords. Earliness, genetic algorithm, tardiness, makespan, energy efficient, JSSP, preventive maintenance, PFJSP, SDST, job shop.

1. Introducción

La planificación de la fabricación es un elemento esencial de la gestión de la cadena de suministro; en la fabricación inteligente, la programación eficiente de los pedidos de los clientes juega un papel importante para maximizar la productividad y las ganancias [1]. El problema de secuenciamiento de la producción, es uno de los campos más importantes en la fabricación inteligente, donde cada trabajo consta de un conjunto específico de operaciones, que deben

procesarse de acuerdo con un orden determinado [2]. En muchas industrias, así como señala [2], entre las que se encuentran la manufactura de químicos, impresión, fabricación de productos farmacéuticos y automóviles, entre otras; los tiempos de *set up* no solo se requieren a menudo entre trabajos, sino que también dependen en gran medida del trabajo en sí y del trabajo anterior que se ejecutó en la misma máquina, generando una dependencia en la secuencia.

Al plantear el problema de secuenciamiento con tiempos de alisto dependientes de este en un escenario realista, podemos encontrar múltiples variables dentro del sistema, las cuales delimitan restricciones para la solución planteada. En este documento las variables a considerar son: la flexibilidad de la maquinaria, la presencia de penalizaciones producto de los adelantos o los atrasos en entregas del producto, el mantenimiento periódico de la maquinaria y la necesidad de hacer uso eficiente de la energía.

En este sentido, [3] Delimita que una planta es parcialmente flexible si se presenta por lo menos una máquina que no sea capaz de realizar todas las operaciones necesarias, lo cual es común de encontrar en la mayoría de las industrias. Por su parte, las penalizaciones por atraso y adelanto son comunes en la industria, el primero se encuentra ligado a penalizaciones relacionadas a la incapacidad para cumplir con una fecha de entrega, mientras que los costos con adelantos se encuentran ligados con el mantenimiento de inventario [4].

Además, [5] señala que recientemente, el aumento del consumo de energía se ha convertido en una importante preocupación a nivel global, de forma que la demanda total de consumo de energía se ha duplicado en los últimos 40 años y se espera que se vuelva a duplicar para el 2030; por lo que la eficiencia energética se ha vuelto una prioridad en la industria. Así mismo, con el fin de verificar el adecuado funcionamiento de la maquinaria, es usual la aplicación de mantenimiento preventivo a las máquinas [6], por lo que, como se observa, todas estas variables son relevantes para el desarrollo es un escenario más realista, de un problema de secuenciamiento con tiempos de alisto dependientes del secuenciamiento.

El algoritmo genético (GA) es una optimización estocástica técnica inspirada en la naturaleza y es similar al proceso de evolución en el mundo biológico donde el más apto sobrevivir para reproducirse. De esta forma se podrá delimitar una serie de soluciones para el algoritmo con el que se cuenta, y se elegirán aquellas que cumplen con los objetivos planteados, de la mejor forma posible, de manera que las mejores soluciones se van combinando hasta delimitar la solución óptima [7]. En este caso, nuestros objetivos consisten en disminuir el consumo energético, disminuir el *makespan*, al mismo tiempo que se evita la falta o la retención de inventario; considerando los múltiples factores mencionados anteriormente.

Este artículo científico contará con otras 5 secciones: la revisión bibliográfica del problema de secuenciamiento y las múltiples variables y temas que se abordarán en esta investigación, una crítica al desarrollo que se le ha dado a la investigación de este tema, de forma que se delimite un nuevo enfoque para la investigación; la propuesta al enfoque delimitado, un ejemplo que permita ejemplificar el algoritmo

propuesto, un espacio para sugerir futuras líneas de investigación de importancia para abordar en este tema y las conclusiones recabadas de todo este proceso.

2. Materiales y Métodos Metodología

El SDST-FJSP, por sus siglas en inglés, es definido por [2] de la siguiente manera: este problema consiste en realizar "n" trabajos en "m" máquinas. Donde cada trabajo consiste en una secuencia de operaciones que debe realizarse para completar dicho trabajo. La ejecución de cada operación requiere una máquina del conjunto de máquinas disponibles y los tiempos de configuración dependen de la secuencia de trabajos. Como señala [3] la mayoría de los enfoques brindados a la resolución de este problema son metaheurísticos, por su parte [3] le brinda un enfoque de Pareto proporciona una alternativa a optimización multiobjetivo. Mientras que tanto [2] y [7] obtiene resultados exitosos por medio de la aplicación de algoritmos genéticos.

En cuanto a la flexibilidad del sistema, [7] y [3] señalan que esta puede categorizarse de dos formas. Presentándose flexibilidad total cuando cada operación puede ser procesado en cualquiera de las máquinas y parcial cuando cada operación puede ser realizado solo en un subconjunto de máquinas; ambos señalan que esto aumenta el nivel de complejidad del problema a tratar, debido a las restricciones existentes dentro del procesamiento.

En la actualidad cada vez es más evidente la influencia de más de un factor en el problema de secuenciamiento, normalmente se investiga sobre los talleres flexibles como factor principal, sin embargo, otros factores como la disminución del *makespan*, la eficiencia energética, el mantenimiento previo, entre otros, no es común verlos incorporados en un solo algoritmo [8].

En la literatura existen diferentes autores que han aplicado algoritmos híbridos de dos o más factores para la solución del problema de secuenciamiento con tiempos de alisto dependientes del secuenciamiento, [8] explora el problema del taller con dos suposiciones: tiempos de preparación dependientes de la secuencia y restricciones de disponibilidad de la máquina. Propusieron un método de hibridación del algoritmo SA y de tipo electromagnético como mecanismo para minimizar la tardanza ponderada total.

En lo que respecta al efecto de los retrasos y adelantos en la producción, en [9] se presenta una solución al JSSP con énfasis en la minimización del *makespan* y el retraso total mediante el algoritmo de colonia de abejas artificial; en esta investigación se llega a la conclusión de que mediante la metaheurística se puede obtener un procedimiento único que permita resolver problemas con casos grandes que tienen un

mayor número de objetivos. De igual forma, [10] analiza el atraso en la producción; no obstante, este considera un costo por penalización y un máximo permitido de retraso. También [11] desarrolla una solución para un JSSP multiobjetivo mediante el algoritmo genético; los autores toman en consideración los objetivos de tiempo y coste, incluyendo el tiempo de finalización, el retraso total, el plazo de entrega, el coste de producción y la pérdida de máquinas; sin embargo, dicha investigación no toma en cuenta la flexibilidad de las máquinas con la que cuentan las empresas hoy en día, ya sea de forma parcial o total.

La eficiencia energética en el modelado del problema de secuenciamiento, de acuerdo con la bibliografía consultada se aborda principalmente desde la perspectiva de control de emisiones bajo un programa de sostenibilidad o regulaciones ambientales externas con un límite de consumo, es decir, una restricción a nivel matemático limitándose a seleccionar entre las mejores soluciones para los objetivos planteados, las que se encuentren bajo este nivel [12] [13]. Sin embargo, la inclusión de esta variable como objetivo, en búsqueda de una solución que minimice el consumo energético total, se realiza mayoritariamente en problemas de secuenciamiento bi-objetivo, o en planteamientos multivariable donde únicamente se plantea el costo eléctrico en un costo total de producción [14] y no se analiza integralmente como parte de un planeamiento productivo que permita aprovechar al máximo los recursos disponibles mientras se reducen emisiones y costos energéticos, en conjunto con la optimización del resto de variables. [15]

En cuanto al mantenimiento preventivo, [6] define tres políticas para abordarlos: la primera es cuando se planean los mantenimientos siempre entre los mismo intervalos de tiempo, la segunda es llamada como el periodo óptimo de mantenimiento donde se considera que la probabilidad de una falla se puede modelar bajo alguna distribución y se planean los mantenimientos preventivos de tal manera que se maximiza el tiempo de operación, la última política es similar a la anterior con la diferencia de que la probabilidad de una falla aumenta con el tiempo. A la hora de realizar este tipo de modelos es necesario conocer el comportamiento de la maquinaria para poder modelar su comportamiento, así mismo, los comportamientos pueden variar radicalmente de una industria a otra.

Adicionalmente, [16] señala el creciente interés de las empresas por la producción *just in time* (JIT), la cual corresponde a la planificación de la producción de forma que el producto no se produzca antes o después, sino en el momento preciso de manera que se minimicen los gastos operativos.

2.1 Crítica

A pesar de que el problema del secuenciamiento ha sido muy estudiado a lo largo de los años, incorporando múltiples variables al mismo, como lo han hecho [6], [8], [17] y [15]; los cuales abordan este problema tomando en consideración distintos factores que pueden afectar la resolución del mismo, la mayoría de estos autores se limitan a abordar dentro de su problema únicamente una o dos variables, lo cual limita la capacidad de generar un algoritmo que se apegue a la realidad vivida diariamente en un ciclo productivo.

Debido a todo lo anteriormente mencionado es que esta investigación busca abordar el problema del secuenciamiento con tiempos de alisto dependientes de este, bajo un enfoque multiobjetivo, de forma que se permita disminuir la duración del *makespan*, al mismo tiempo que se disminuye el consumo energético y se minimizan las penalizaciones producto de las entregas tardías y el almacenamiento de inventario. Al mismo tiempo, el algoritmo a generar es multivariable, ya que toma en consideración la flexibilidad parcial brindada por la maquinaria y el mantenimiento preventivo que esta necesita, también considerando que existen tiempos de alisto dependientes del secuenciamiento. Se plantea el uso de un algoritmo genético para delimitar la metodología a implementar para solucionar el problema, como lo han hecho múltiples autores entre los que se encuentran: [2], [7], [11], [18] y [19], de forma que se consiga un algoritmo capaz de apegarse a la realidad esperada en una planta productiva.

2.2 Propuesta

2.1.1 Planteamiento matemático

Se plantea una propuesta de solución para el problema, para el cual se proponen las siguientes funciones objetivo. En primer lugar, se busca minimizar el *makespan* (ecuaciones 1 y 2).

$$\text{Min} (\text{Makespan}) = \sum_{i=1}^n TCT_i \quad (1)$$

Con:

$$TCT_i = \sum_{j=1}^m TC_{ij} + \sum_{j=1}^m ST_{ij} + \sum_{j=1}^m TE_{ij} \quad (2)$$

Donde:

TCT_i = Tiempo de ciclo del trabajo i .

m = Cantidad total de trabajos.

n = Cantidad total de operaciones del trabajo i .

TC_{ij} = Tiempo C_{ijk} del trabajo i , operación j en la máquina k seleccionada.

ST_{ij} = *Setup time* del trabajo i , operación j en la máquina k seleccionada.

TE_{ij} = Tiempo de espera del trabajo i , operación j .

También se requiere una combinación que minimice el costo total en penalizaciones por atrasos y adelantos (ecuación 3 y 4).

$$\text{Min} (\text{CostoTotal}_{(\text{atrasos/adelantos})}) = \sum_{i=1}^n P_i \quad (3)$$

Con:

$$P_i = TCT_i * \text{Penalización} \quad (4)$$

Donde:

TCT_i = Tiempo de ciclo del trabajo i .

P_i = Costo total en penalizaciones por atrasos y adelantos.

n = Cantidad total de operaciones del trabajo i .

penalización = penalización por adelanto o atraso, dependiendo del tiempo total de ciclo y el *due time* de cada operación.

Finalmente, se establece el objetivo de minimizar el consumo eléctrico de cada una de las máquinas (ecuaciones 5 y 6).

$$\text{Min} (\text{ConsumoEléctrico}) = \sum_{k=1}^M C_k \quad (5)$$

Con:

$$C_k = (X_k - H_k) * PN_k \quad (6)$$

Donde:

C_k = Consumo eléctrico total de la máquina k .

H_k = Hora en la que se enciende la máquina k siendo

$H = 0$ el momento que se enciende la primera máquina.

X_k = Tiempo que la máquina k ha estado encendida.

PN_k = Potencia nominal de la máquina k .

M = Cantidad total de máquinas.

Además, se cuentan con restricciones en cuanto a las soluciones, tiempo de actividad continua de las máquinas, y el uso de las máquinas para realizar las operaciones. En cuanto a las soluciones se tiene el siguiente formato (ecuación 7):

$$\begin{bmatrix} & M_1 & M_2 & \dots & M_k \\ O_{11} & S_{111} & S_{112} & \dots & S_{11k} \\ O_{12} & S_{121} & S_{122} & \dots & S_{12k} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ O_{ij} & S_{ij1} & S_{ij2} & \dots & S_{ijk} \end{bmatrix} \quad (7)$$

Esta solución cuenta con restricciones que indican que solo se puede asignar una máquina para realizar la operación ij , y la máquina seleccionada realizará la operación por completo. También, debido a que se está trabajando con flexibilidad parcial, algunas combinaciones serán imposibles; es decir, según lo establezca el caso la máquina k no podrá realizar la

operación ij ($S_{ijk} = 0$). Estas restricciones se expresan de la siguiente manera (ecuación 8):

$$\begin{aligned} S_{ij1} + S_{ij2} + \dots + S_{ijk} &= 1 & (8) \\ S_{ijk} &\in [0,1]; \text{ cuando sea posible la combinación.} \\ S_{ijk} &= 0; \text{ cuando la máquina no pueda realizar } O_{ij}. \\ S_{ijk} &\in Z^+ \end{aligned}$$

Donde:

O_{ij} = Operaciones j del trabajo i .

M_k = Máquina k .

S_{ijk} = cuando su valor es 1 indica que la operación ij se realizará en la máquina k .

Z^+ = Números enteros positivos.

Asimismo, la solución puede variar el orden en el que se procesan cada una de las operaciones, lo que va a afectar positiva o negativamente al *makespan*. No obstante, existe una restricción en cuanto al orden en la que se procesan las operaciones de un mismo trabajo. La restricción anterior se expresa de la siguiente forma (ecuación 9):

$$i_{ijk} < i_{ij-1} \quad (9)$$

Es decir, la posición en el cromosoma, y por ende el orden de procesamiento, de la operación ij debe ser posterior a la posición de la operación previa ($j-1$). Por último, se cuenta con una restricción en cuanto al tiempo que puede estar una máquina en funcionamiento continuo antes de realizarle mantenimiento preventivo. Para ello se define que (ecuación 10):

$$TA_k \leq TM_k \quad (10)$$

Donde:

TA_k = Tiempo de procesamiento que ha transcurrido desde que se encendió la máquina k o desde el último mantenimiento.

TM_k = Tiempo entre mantenimientos establecido para la máquina k .

DM_k = Duración del mantenimiento preventivo en la máquina k .

En esta última restricción cabe mencionar que una vez iniciada una operación la máquina no se va a detener hasta terminar, por lo que si una máquina excede el tiempo entre mantenimientos durante una operación se deberá esperar que se finalice para proceder con el mantenimiento respectivo. Cada vez que una máquina k alcance el tiempo entre mantenimientos, se realizará el mantenimiento preventivo de duración

DM_k , el cual se tomará en cuenta en el tiempo total que la máquina k ha estado encendida y en el tiempo de espera de la siguiente operación que sea procesada en dicha máquina.

2.1.2 Algoritmo genético

Por otro lado, para solucionar el problema planteado es necesario utilizar un algoritmo que es capaz de escoger la mejor solución posible de forma rápida y eficiente; por lo tanto, se define el uso de un algoritmo genético, debido a que es un método ampliamente utilizado en la solución en este tipo de problemas, además que esta opción permite trabajar con distintos tipos de variables, como listas o enteros.

Como lo señala [20], un algoritmo genético es un algoritmo evolutivo que toma como base la Teoría de la evolución darwiniana, la cual propone que, a partir de una población inicial, los individuos más aptos sobreviven y generan descendencia y los menos aptos perecen por lo que tienen menos probabilidades de reproducirse, y el ciclo se repite con la descendencia.

Con base en la misma teoría de la evolución, el algoritmo genético realiza las siguientes etapas [21]:

- **Población inicial:** lo primero que hace el algoritmo es generar una población inicial de forma aleatoria. Para crear una población inicial es necesario determinar el tamaño de la población, el formato del cromosoma, los valores posibles para los genes y las restricciones para que una solución sea válida según el problema. Cabe mencionar que el gen es la unidad mínima de información, cromosoma es el nombre que se le dan a una posible solución dentro del vocabulario del algoritmo genético; además, estos cromosomas son los que va a conformar la población a analizar. En este caso, en específico, se definieron poblaciones con un tamaño de cuatro cromosomas. Cada una de las posibles soluciones que brinde el algoritmo deben cumplir con las restricciones mencionadas en la sección de planteamiento matemático (orden y flexibilidad parcial). También se determinó que no hay prioridad en cuanto a los trabajos, por lo que los genes pueden tomar cualquier posición mientras cumpla con las demás restricciones; cada gen va a contener el número del trabajo, el número de la operación i y la máquina asignada a la operación ij ; y el formato del cromosoma es de la siguiente manera (ecuación 11):

$$[[i1, j1, k] [i2, j2, k] [\dots] [i9, j9, k]] \quad (11)$$

- **Fitness function (evaluación):** en esta etapa se evalúan cada una de las soluciones contenidas en la población, ya sea la

población inicial o las siguientes generaciones. A cada cromosoma se le va a asociar un valor fitness, el cual sirve de referencia para que al comparar distintos cromosomas el algoritmo identifique el mejor. En el caso planteado, al tratarse de un algoritmo genético multiobjetivo, cada solución va a ser evaluado con base en las tres funciones objetivo, lo que va a dar como resultado vectorial (*makespan*, penalización, consumo). Para determinar el valor *fitness* de una solución se va a calcular la distancia del resultado vectorial al origen (0,0,0). Por lo tanto, entre más cerca se encuentre una solución del origen, menor será su valor *fitness* y será una mejor solución al problema.

- **Selección:** una vez formada la población, se deben seleccionar los mejores cromosomas, que serán denominados "padres" para pasar sus genes a la siguiente generación; es decir, las mejores soluciones de la presente población serán usadas de base para generar la siguiente población de soluciones y el resto serán desechadas.
- **Crossover:** para formar la siguiente población, se busca que los resultados sean distintos y mejores que los anteriores, por lo que se debe de establecer un punto de crossover. Los cromosomas hijos va a ser una mezcla de genes donde los genes antes del punto definido van a corresponder al primer padre y el resto de los genes serán los del otro padre.
- **Mutación:** finalmente, para evitar que la solución se limite a una solución óptima local, se realizan cambios en algunos genes de los cromosomas hijos. Los genes y los valores son de carácter aleatorio. Terminada la fase de mutación, se establece una nueva población conformada por los cromosomas padres, los cromosomas hijos toman el lugar de las soluciones desechadas en la etapa de selección y se repite el ciclo empezando desde la función fitness. Este ciclo termina cuando se completan la cantidad de iteraciones definidas o cuando la respuesta se estabiliza y llega a una solución óptima.

Debido a la complejidad de la propuesta planteada, cada vez que se ponga a funcionar el algoritmo para encontrar una solución óptima, las limitaciones del problema conducirán al código a brindar una solución óptima local en pocas iteraciones; por lo tanto, la mayoría de las veces se obtendrá una respuesta óptima distinta que depende en su mayoría de la población inicial generada.

3. Resultados y discusión

3.1 Planteamiento del escenario

Con el fin de ejemplificar el funcionamiento del algoritmo genético generado, y explicado en la sección anterior, se decidió simular el problema que desarrolla [2] en el lenguaje

de programación Python. Es importante considerar que, el problema expuesto por [2], corresponde a un problema de secuenciamiento que busca minimizar el *makespan* dentro de un sistema parcialmente flexible y que cuenta con tiempos de alisto que dependen del orden de la secuencia, debido a esto, es que únicamente los datos de las tablas 1 y 2 fueron extraídas de este artículo para formar parte de nuestra simulación.

Tabla 1. Tiempos de ciclo de las operaciones

Job	Operación	M ₁	M ₂	M ₃
J ₁	O11	4	-	5
	O12	-	3	4
	O13	6	5	-
J ₂	O21	3	-	4
	O22	4	5	-
	O23	-	4	7
J ₃	O31	5	3	-
	O32	-	4	-
	O33	4	5	3

Tabla 2. Setup times dependientes del secuenciamiento

Machine	M ₁			M ₂			M ₃		
Jobs	J ₁	J ₂	J ₃	J ₁	J ₂	J ₃	J ₁	J ₂	J ₃
Dummy	3	2	1	2	4	1	4	3	2
J ₁	0	1	3	0	2	4	0	2	3
J ₂	1	0	2	4	0	3	2	0	3
J ₃	1	3	0	2	1	0	3	2	0

Con el fin de tomar en cuenta el resto de las variables que abarca el modelo matemático planteado, fue necesario la incorporación de datos ficticios. Primeramente, se definieron los tiempos de entrega, los cuales son necesarios para delimitar las penalizaciones por el incumplimiento de estos, ya sea por adelanto o por atraso, esta información se encuentra disponible en la tabla 3.

Tabla 3. Tiempo de entrega acordado para los trabajos

Job	Due date (horas)	Penalización por adelanto	Penalización por atraso
J ₁	20	\$1	\$16
J ₂	15	\$8	\$14
J ₃	25	\$4	\$250

Posteriormente se definieron los tiempos entre mantenimientos y la duración de estos para cada máquina (tabla 4).

Tabla 4. Tiempos entre mantenimiento y duración del mantenimiento de las máquinas.

Máquina	M ₁	M ₂	M ₃
Tiempo entre mantenimientos (h)	6	4	7
Duración del mantenimiento (h)	3	4	2

Por último, se establecieron las potencias nominales para cada máquina, estas se encuentran en la tabla 5.

Tabla 5. Potencia nominal de las máquinas

Máquina	M ₁	M ₂	M ₃
Potencia nominal (kW)	2	1	3

Ya con todas las variables definidas se pudo realizar una simulación que permitiera solucionar el problema de secuenciamiento considerando que el sistema es parcialmente flexible, la maquinaria es sometida a mantenimiento preventivo, y se cuenta con tiempos de alisto dependientes del secuenciamiento; al mismo tiempo que se busca la disminución de los gastos por las penalizaciones de entrega temprana o tardía de producto terminado, minimizar el consumo de recurso energético de la maquinaria y disminuir el *makespan*.

3.2 Supuestos

Es importante considerar que para el desarrollo de la simulación propuesta se realizaron los siguientes supuestos:

- Las penalizaciones por adelanto o atraso se delimitan por trabajo realizado, no por operación.
- Las penalizaciones por adelanto o atraso para cada trabajo son independientes entre sí, es decir que, si tengo un atraso tanto en J1 y J2, los valores no cambian, se mantienen los delimitados en la tabla 4.
- No se toma en consideración la posible ocurrencia de eventos extraordinarios, como lo puede ser el averío de una de las máquinas o un corte de electricidad en la planta.
- La maquinaria y todo el proceso productivo se encuentra automatizado, por lo que, el factor humano no es contemplado.
- Los tiempos de alisto, operación y mantenimiento; no presentan variación con el paso del tiempo.
- Todos los trabajos poseen la misma cantidad de operaciones.
- Cada operación se procesa en una sola máquina a la vez.
- Cada máquina es capaz de procesar una única operación a la vez.
- El tiempo de procesamiento de cada operación depende de la máquina y las máquinas son independientes entre sí.
- Cada operación, una vez iniciada, debe completarse sin interrupción.

3.3 Resultados de la simulación

Como se explicó en la sección anterior, el algoritmo genético desarrollado para encontrar una solución óptima para las funciones objetivo expresadas en las ecuaciones 1, 3 y 5, fue evolucionando de acuerdo a los valores de la población generada. Debido a la recursividad del algoritmo desarrollado, lo cual permite establecer una separación de aquellas soluciones que mejor se adaptan a la optimización deseada, se obtuvieron cinco generaciones, lo cual brinda en su totalidad una población que cuenta con 5,460 cromosomas. Con los

5,460 cromosomas generados, se logró desarrollar 1,365 soluciones, las cuales pueden observarse en la figura 1. En esta figura la solución óptima se ve representada por una estrella; también, se observa que aquellas respuestas que presentaron un mejor valor *fitness* se encuentran diferenciadas con el color rojo, mientras que aquellas con los valores más pequeños se encuentran en tonos azulados. El valor *fitness* de la respuesta se delimitó que correspondía a 143.35; de forma que este fue la distancia mínima que el algoritmo pudo encontrar entre el origen (0,0,0) y todas las respuestas que presentaban viabilidad para solucionar el problema de secuenciamiento, minimizando el *makespan*, el consumo eléctrico y las penalizaciones por no controlar la salida de inventario.

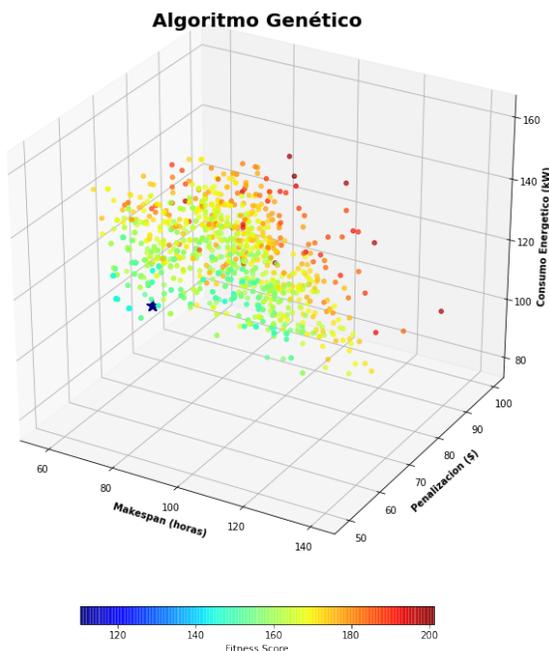


Figura 11. Resultados de la comprobación del algoritmo genético.

La solución óptima está compuesta por los vectores de la matriz de la siguiente ecuación, de forma que sí se logra comprobar que el algoritmo es capaz de encontrar el secuenciamiento óptimo delimitado para las tres máquinas y las nueve operaciones que conforman a los tres trabajos a desarrollarse en su totalidad (ecuación 12).

$$\begin{bmatrix} [3,1,2] & [2,1,1] & [3,2,2] \\ [2,2,1] & [1,1,1] & [3,3,2] \\ [1,2,2] & [2,3,3] & [1,3,2] \end{bmatrix} \quad (12)$$

Donde:

Cada vector presenta la forma $[i,j,k]$, siendo 'i' el número del trabajo, 'j' el número de operación, y 'k' el número de máquina.

Sin embargo, estos vectores solo nos brindan tres valores, el número del trabajo a llevarse a cabo, el número de la operación a desarrollar, y el número de máquina en la cual debe llevarse a cabo, en ese respectivo orden. Para facilitar el entendimiento del secuenciamiento propuesto por la solución óptima encontrada por el algoritmo, se realizó un diagrama de Gantt que representa gráficamente el secuenciamiento presente en cada máquina de acuerdo con la hora. En este se incluye el *set up* time de las máquinas, para aquellas operaciones que lo requieren, y el tiempo de mantenimiento que cada máquina requiere. Este diagrama corresponde a la figura 2, la cual se encuentra en el apéndice del documento.

Por medio de la aplicación del secuenciamiento planteado por el algoritmo genético generado, y tomando en consideración las múltiples variables que se abordaron para el desarrollo de este, de forma que se apegara más a un escenario real del entorno productivo; se obtuvo que en total la línea de producción contaría con un *makespan* de 70 horas, es imposible evitar las penalizaciones por falta o exceso de inventario, y estas serían de 68 dólares, además de contar con un consumo de 105 kW. Este resultado se expone gráficamente en la figura 1.

Además, podemos observar que tanto la máquina 1 como la 2, son las que presentarían un mayor ciclo de trabajo, especialmente la máquina 2, la cual debe estar disponible por 23 horas y contar con tres ciclos de mantenimiento; por lo cual, sería necesario contar con turnos tanto diurnos como nocturnos para hacer funcionar la línea de forma óptima. Al mismo tiempo, se muestra coherencia en el resultado brindado, ya que, la mayoría de la producción se centra en la máquina 2, esto es consecuencia del hecho de que esta máquina es la que presenta mayor flexibilidad, al mismo tiempo que un menor consumo energético en comparación a la máquina 1 y la máquina 3. Por otra parte, la máquina 3, al ser menos flexible y presentar un mayor consumo energético, es la que menos cantidad de horas se usa para producción.

Es importante recalcar que, debido a las múltiples variables que se incorporaron dentro del algoritmo planteado, y la complejidad de este, no se vio factible realizar una comparación entre los resultados obtenidos, con aquellos expuesto por [2], ya que, el objetivo de este se limitaba a disminuir el *makespan* y únicamente contemplaba la existencia de *set up times* dependientes y un sistema parcialmente flexible. A pesar de esto, la complejidad del algoritmo

planteado permitiría que este se reestructurara fácilmente para abordar problemas de secuenciamiento más simples que el delimitado por nuestro equipo.

4. Conclusiones

La presente propuesta ofrece una solución para la problemática según las variables y condiciones planteadas, sin embargo, esta se puede adaptar a gran variedad de situaciones dependiendo de las necesidades específicas que pueda poseer cada usuario o empresa que desee aplicar soluciones similares. Así mismo es importante considerar que para cada solución encontrada es necesario un análisis posterior para corroborar de que esta sea realmente compatible en el marco de la industria para la que se requiere y en las condiciones propias de la empresa tomando en cuenta la distribución de los turnos y la capacidad de producción.

La modelación matemática utilizada permite integrar las características de eficiencia energética, *makespan* y penalizaciones por atraso y adelanto desde una aproximación multiobjetivo, lo que se ajusta de mejor manera a la realidad de la producción y no reduce la solución a una simple ecuación de costo.

Por otra parte, al utilizar un algoritmo genético multigeneracional se reduce la probabilidad de que la solución encontrada se encuentre en un mínimo local. Esto se debe al funcionamiento de este tipo de algoritmos que suele mutar por lo que no es común que las soluciones caigan en este tipo de secuencias. Con esto en cuenta, se recomienda realizar varias corridas para cada problema ya que es posible que de estas se obtengan resultados distintos por lo que se considera de gran interés el estudio de distintas soluciones con el fin de encontrar algún patrón o una conexión en común entre las secuencias que permita expandir el análisis sobre estas.

Al mismo tiempo es importante recalcar que la propuesta planteada en este documento pretende adaptarse de la mejor manera posible a los diferentes escenarios que puedan enfrentar las empresas, con el fin de brindar la solución óptima a partir de las condiciones del proceso.

Sin embargo, existen variables que fueron consideradas en el algoritmo que podrían ser más específicas, en este caso se toma la eficiencia energética como variable a partir de la evaluación de la potencia de cada máquina, no obstante, sería de gran utilidad para un proceso productivo, considerar las tarifas eléctricas ya que estas aumentan o disminuyen dependiendo de la hora en la que se esté produciendo, de manera que el algoritmo sea capaz de sugerir la hora a la que debe dar inicio la producción y si esta debe detenerse en algún momento para minimizar el consumo eléctrico.

También se considera un lapso fijo de mantenimiento por máquina, pero esos tiempos en una situación real no son iguales y van a depender de las averías o el mantenimiento que la máquina necesite, por lo que se sugiere modelar los tiempos por medio de una distribución normal para obtener un promedio de datos y su desviación.

Actualmente se tiene como supuesto que todos los trabajos poseen una misma cantidad de operaciones, pero sería de gran interés desarrollar el algoritmo con el fin de considerar trabajos con distintas operaciones y, además, se podría ampliar el algoritmo con mayor cantidad de máquinas y trabajos para evaluar su desempeño.

AGRADECIMIENTOS

Agradecemos a la Ing. Sofía Castrillo por la guía y retroalimentación brindada a lo largo del desarrollo de esta investigación.

CONFLICTO DE INTERESES

Los autores declaran no tener algún conflicto de interés.

REFERENCIAS

- [1] Lei He, Mathijs de Weerd y Neil Yorke-Smith. "Time/sequence-dependent scheduling: the design and evaluation of a general purpose tabu-based adaptive large neighbourhood search algorithm". En: Journal of Intelligent Manufacturing 31 (4 abr. de 2020), págs. 1051-1078. ISSN: 15728145. DOI: 10.1007/s10845-019-01518-4.
- [2] Ameni Azzouz, Meriem Ennigrou y Lamjed Ben Said. "Flexible job-shop scheduling problem with sequence-dependent setup times using genetic algorithm". En: vol. 2. SciTePress, 2016, págs. 47-53. ISBN: 9789897581878. DOI: 10.5220/0005821900470053.
- [3] Tsung Che Chiang y Hsiao Jou Lin. "A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling". En: International Journal of Production Economics 141 (1 ene. de 2013), págs. 87-98. ISSN: 09255273. DOI: 10.1016/j.ijpe.2012.03.034.
- [4] Michele Lombardi y Michela Milano. Optimal methods for resource allocation and scheduling: A cross-disciplinary survey. Ene. de 2012. DOI: 10.1007/s10601-011-9115-6.
- [5] Hadi Mokhtari y Aliakbar Hasani. "An energy-efficient multi-objective optimization for flexible job-shop scheduling problem". En: Computers and Chemical Engineering 104 (2017), págs. 339-352. ISSN: 00981354. DOI: 10.1016/j.compchemeng.2017.05.004.
- [6] K. Naboureh y E. Safari. "Integrating the sequence dependent setup time open shop problem and preventive maintenance policies". En: Decision Science Letters (2016), págs. 535-550. ISSN: 19295804. DOI: 10.5267/j.dsl.2016.4.002.

- [7] S. I. Ao, Len. Gelman, David W. L. Hukins, Andrew. Hunter, Alexander. Korsunsky e International Association of Engineers. World Congress on Engineering: WCE 2013 : 3-5 July, 2013, Imperial College London, London, U.K. 2013, pág. 2249. ISBN: 9789881925107.
- [8] Ming Li y Deming Lei. “An imperialist competitive algorithm with feedback for energy-efficient flexible job shop scheduling with transportation and sequence-dependent setup times”. En: Engineering Applications of Artificial Intelligence 103 (2021), pág. 104307. DOI: 10.1016/j.engappai.2021.104307.URL: https://doi.org/10.1016/j.engappai.2021.104307.
- [9] Abyson Scaria, Kiran George y Jobin Sebastian. “An Artificial Bee Colony Approach for Multi-objective Job Shop Scheduling”. En: Procedia Technology 25 (2016), págs. 1030-1037. ISSN: 22120173. DOI: 10. 1016 / J. PROTCY.2016.08.203. URL: www.sciencedirect.com.
- [10] Jae Gon Kim, Hong Bae Jun, June Young Bang, Jong Ho Shin y Seong Hoon Choi. “Minimizing tardiness penalty costs in job shop scheduling under maximum allowable tardiness”. En: Processes 8 (11 nov. de 2020), págs. 1-15. ISSN: 22279717. DOI: 10.3390/pr8111398.
- [11] Si Chen Liu, Zong Gan Chen, Zhi Hui Zhan, Sang Woon Jeon, Sam Kwong y Jun Zhang. “Many-Objective Job-Shop Scheduling: A Multiple Populations for Multiple Objectives-Based Genetic Algorithm Approach”. En: IEEE Transactions on Cybernetics (2021). ISSN: 21682275. DOI: 10.1109/TCYB.2021.3102642.
- [12] Jiong Guo y Deming Lei. “An imperialist competitive algorithm for energy-efficient flexible job shop scheduling; An imperialist competitive algorithm for energy-efficient flexible job shop scheduling”. En: (2021). DOI: 10.1109/CCDC52312.2021.9601714.
- [13] Deming Lei, Ming Li y Ling Wang. “A Two-Phase Meta-Heuristic for Multiobjective Flexible Job Shop Scheduling Problem with Total Energy Consumption Threshold”. En: IEEE Transactions on Cybernetics 49.3 (2019), págs. 1097-1109. ISSN: 21682267. DOI: 10.1109/TCYB.2018.2796119.
- [14] Mingyang Zhang, Jihong Yan, Yanling Zhang y Shenyi Yan. “Optimization for energy-efficient flexible flow shop scheduling under time of use electricity tariffs”. En: Procedia CIRP 80 (2019), págs. 251-256. ISSN:22128271. DOI: 10. 1016 / j . procir. 2019 . 01 . 062. URL: https://doi.org/10.1016/j.procir.2019.01.062.
- [15] Jiong Guo y Deming Lei. “An imperialist competitive algorithm for energy-efficient flexible job shop scheduling”. En: Proceedings of the 33rd Chinese Control and Decision Conference, CCDC 2021 (2021),págs. 5145-5150. DOI: 10. 1109 / CCDC52312 . 2021 .9601714.
- [16] Saheed Akande y Ganiyu Ajisegiri. “Three classes of Earliness-Tardiness (E/T) scheduling problems”. En: Fuel Communications 10 (mar. de 2022), pág. 100047. ISSN: 26660520. DOI: 10. 1016 / J. JFUECO. 2021 .100047. URL: https://doi.org/10.1016/j.jfueco.2021.100047.
- [17] W. Bouazza, Y. Sallel y B. Beldjilali. “A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect”. En: vol. 50. Elsevier B.V., jul. de 2017, págs. 15890-15895. DOI: 10.1016/j.ifacol.2017.08.2354.
- [18] E. Salazar-Hornig y S.J.C. Medina. “Makespan Minimization for The Identical Machine Parallel Shop with Sequence Dependent Setup Times Using a Genetic Algorithm”. En: Ingeniería, Investigación y Tecnología 14 (1 2013), págs. 43-51. ISSN: 14057743.
- [19] Fatima Abderrabi, Matthieu Godichaud, Alice Yalaoui, Farouk Yalaoui, Lionel Amodeo, Ardian Qerimi y Eric Thivet. “Flexible Job Shop Scheduling Problem with Sequence Dependent Setup Time and Job Splitting:Hospital Catering Case Study”. En: Applied Sciences 11 (4 feb. de 2021), pág. 1504. ISSN: 2076-3417. DOI: 10.3390/app11041504.
- [20] Juan Andrés Ambuludi Olmedo. “Aplicación de Algoritmos Genéticos para la optimización de problemas combinatorios”. En: (2018).
- [21] Samuel de Greiff y Juan Carlos Rivera. “Investment portfolio optimization with transaction costs through a multiobjective genetic algorithm: An applied case to the Colombian Stock Exchange”. En: Estudios Gerenciales 34 (146 2018), págs. 74-87. ISSN: 26656744. DOI: 10.18046/J.ESTGER.2018.146.2812.

ANEXOS

Anexo 1. Diagrama de Gantt del secuenciamiento recomendado.

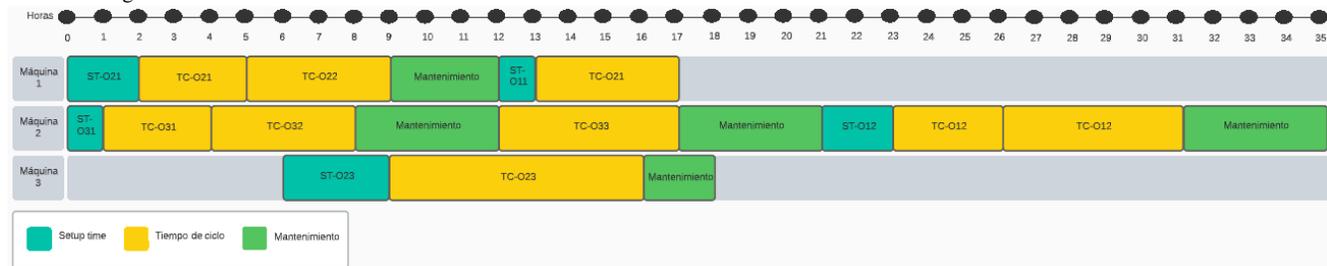


Figura 2. Diagrama de Gantt del secuenciamiento recomendado.