

Un Algoritmo Híbrido para la solución del Problema Job Shop Considerando Eficiencia Energética¹

Fabián Alexander Torres Cárdenas², Lina Mayerly Lozano Suárez³, Víctor Alfonso Sanabria Ruiz⁴

Resumen

Introducción: La presente investigación aborda un problema bi-objetivo, el clásico de la programación de la producción de sistema tipo taller *Job Shop Scheduling* con la extensión de eficiencia energética el cual busca minimizar el makespan y el consumo de energía. En este, se manejan maquinas fijas por operaciones, se producen variaciones en el consumo de energía y tiempo de procesamiento dependientes de la velocidad, lo cual se asemeja a ambientes reales presentes en las industrias. Este problema es considerado NP-Hard debido a su complejidad computacional. **Objetivo:** El objetivo de esta investigación es diseñar un algoritmo híbrido

para la solución con el fin de minimizar el tiempo computacional y establecer la mejor secuencia y asignación de velocidades buscando reducir el tiempo de finalización de la última tarea (Makespan) y el consumo de energía. **Materiales y métodos:** El enfoque de investigación es cuantitativo de tipo Investigación aplicada tecnológica dado a que sirve para generar conocimientos que se puedan poner en práctica en el sector productivo, con el fin de impulsar un impacto positivo en la vida cotidiana. **Resultados:** El algoritmo propuesto se evaluó con problemas benchmarking de la literatura logrando un buen rendimiento en términos computacionales. **Conclusiones:** El algoritmo Recocido Simulado mejoró los resultados

1 Artículo original derivado del proyecto de investigación “Un algoritmo híbrido para programación de la producción con eficiencia energética en un sistema tipo taller” de la Universidad de Investigación y Desarrollo-UDI, ejecutado entre agosto de 2021 a noviembre del 2021; Grupo de Investigación Sinergia; Financiado por la Universidad de Investigación y Desarrollo-UDI.

2 Magíster en Ingeniería Industrial. Universidad Industrial de Santander, Docente tiempo completo y miembro del grupo Sinergia de la Universidad de Investigación y Desarrollo-UDI, correo: ftorres4@udi.edu.co
ORCID 0000-0003-1781-5380

3 Magíster en Ingeniería Industrial. Universidad Industrial de Santander, Docente tiempo completo y miembro del grupo Sinergia de la Universidad de Investigación y Desarrollo-UDI, Mail llozano7@udi.edu.com /

4 Magíster en Ingeniería Industrial. Universidad Industrial de Santander, Docente tiempo completo y miembro del grupo Sinergia de la Universidad de Investigación y Desarrollo-UDI, Mail vsanabria1@udi.edu.co
ORCID 0000-0002-8065-8894

Autor para Correspondencia: ftorres4@udi.edu.co

Recibido: 03/12/2021 Aceptado: 01/12/2022

*Los autores declaran que no tienen conflicto de interés

obtenidos por el algoritmo genético siendo la unión de los dos una buena alternativa para la solución del problema.

Palabras clave: Eficiencia Energética, Job Shop, Algoritmo Híbrido, Algoritmo Genético; Algoritmo Recocido Simulado.

A Hybrid Algorithm for Solution of the Job Shop Problem Considering Energy Efficiency

Abstract

Introduction: The present research addresses a bi-objective problem, the classic Job-Shop Scheduling problem with the extension of energy efficiency which seeks to minimize the brand and energy consumption. In this, fixed machines are handled by operations, there are variations in energy consumption and processing time depending on speed, which resembles real environments present in industries. This problem is considered NP-Hard due to its computational complexity. **Objective.** The objective of this research is to design a hybrid algorithm for the solution to minimize computational time and establish the best sequence and speed

assignment seeking to reduce the completion time of the last task (Makespan) and energy consumption. **Materials and methods.** The research approach is quantitative of the Technological Applied Research type since it serves to generate knowledge that can be put into practice in the productive sector, to promote a positive impact on daily life. **Results.** The proposed algorithm was evaluated with benchmarking problems in the literature, achieving good performance in computational terms. **Conclusions.** The Simulated Annealing algorithm improved the results obtained by the genetic algorithm, being the union of the two a good alternative for solving the problem.

Keywords: Energy efficiency; Job Shop; Hybrid Algorithm, Genetic algorithm; Simulated Annealed Algorithm.

Um Algoritmo Híbrido para A Solução do Problema do Job Shop Considerando a Eficiência de Energia

Resumo

Introdução: A presente pesquisa aborda um problema bi-objetivo, o clássico problema Job-Shop Scheduling com extensão de eficiência energética que busca minimizar o consumo de marca e energia. Nela, máquinas fixas são movimentadas

por operações, há variações no consumo de energia e no tempo de processamento em função da velocidade, o que se assemelha a ambientes reais presentes nas indústrias. Este problema é considerado NP-Difícil devido a sua complexidade computacional. **Objetivo.** O objetivo desta pesquisa é projetar um algoritmo híbrido para a solução de minimizar o tempo computacional e estabelecer a melhor sequência e atribuição de velocidade buscando reduzir o tempo de conclusão da última tarefa (Makespan) e o

consumo de energía. **Materiais e métodos.** A abordagem da pesquisa é quantitativa do tipo Pesquisa Tecnológica Aplicada, pois serve para gerar conhecimento que pode ser colocado em prática no setor produtivo, para promover um impacto positivo na vida cotidiana. **Resultados.** O algoritmo proposto foi avaliado com problemas de benchmarking na literatura, alcançando bom desempenho

em termos computacionais. **Conclusões.** O algoritmo Simulated Annealing melhorou os resultados obtidos pelo algoritmo genético, sendo a união dos dois uma boa alternativa para resolução do problema.

Palavras-chave: Eficiência energética; Loja de trabalho; Algoritmo híbrido, algoritmo genético; Algoritmo Simulado Recozido.

Introducción

Constantemente las empresas están en la búsqueda del mejoramiento continuo con el propósito de satisfacer al cliente y, en vista de que existe un amplio entorno competitivo se ha incrementado la necesidad de ofrecer productos de calidad a buen precio, de igual manera, es un reto poder mantener o aumentar su participación en el mercado en la actualidad; por esta razón, el hacer uso óptimo de los recursos disponibles se vuelve parte esencial. Así mismo, si se traslada esta situación a una empresa manufacturera con diferentes líneas de productos y bajos volúmenes de producción, la optimización de los recursos se encontraría relacionada con la programación de trabajos, en la cual se involucraría la disponibilidad de las máquinas y su capacidad de procesamiento, lo cual se puede representar como un entorno productivo “*Job Shop*” (Li et al., 2021).

Entre las extensiones del problema *Job Shop Scheduling* (JSP) se encuentra aquel que involucra la eficiencia energética, también conocido como el problema *Job Shop Scheduling* con Eficiencia Energética (*Energy-Efficient Job Shop Scheduling Problem*,

EJSP), el cual, aunque ha sido poco estudiado despierta gran interés, tanto así que, desde el 2007 se han realizado diferentes trabajos entorno a este, incluyendo modelos integrales de servicios públicos, que consideran el consumo de energía, el costo y el riesgo para las tres partes, el proveedor, el consumidor y el operador (Cheng et al., 2013); una de las preocupaciones más comunes en las industrias manufacturas es el aumento del precio de la energía y la tendencia actual de sostenibilidad ya que el tener poco control sobre esta se afecta directamente el costo de producción y el medio ambiente.

Desde el 2007 año en el que empezó a explorarse el EJSP, se han planteado modelos matemáticos bajo la programación lineal entera mixta, solucionados debido a la naturaleza NP-hard del problema a través de técnicas heurísticas y metaheurísticas.

Usualmente, en cada *Job Shop Problem* JSP se consideran variables como el volumen de la producción, los costos, la calidad y variedad del producto, entre otras, sin embargo, el panorama respecto a la problemática del medio ambiente ha cambiado los últimos años, dando prioridad a la protección del

planeta reduciendo el impacto ambiental utilizando diferentes métodos de cambio (Hidalgo & Perez, 2016);(Carlos Sornoza-Bravo & Felipe Sabando-Piguabe, 2021). Por esto, la eficiencia energética se ha consolidado como una variable importante a nivel mundial, debido al interés del ahorro de energía en los Problemas de Job Shop Scheduling, se han realizado varias investigaciones sobre esta temática (H. Zhang et al., 2022); (Ham et al., 2021); (Wang et al., 2021)

Las metodologías utilizadas para resolver el Problema *Job Shop* con Eficiencia Energética (EJSP), se encuentran clasificadas como técnicas aproximadas las cuales se conocen como: métodos heurísticos y metaheurísticos, de estas se pueden destacar el Algoritmo Genético (Genetic Algorithm, GA), Optimización de Colonias de Hormigas (Ant Colony Optimization, ACO), Búsqueda Tabú (Tabú Search, TS), Optimización de Enjambre por Partículas (Particle Swarm Optimization, PSO), Recocido Simulado (Simulated Annealing, SA) (Coca, Castrillón, Ruiz, Mateo-Sanz, & Jiménez, 2019); (Mouzon & Yildirim, 2008).

Liu et al., (2014) plantean un modelo para la reducción del consumo de energía para el clásico JSP, que busca minimizar el consumo total de electricidad y la tardanza ponderada total, para ello emplean el Algoritmo Genético de Clasificación No Dominante (Non-dominated Sorting Genetic Algorithm, NSGA-II) como la solución para obtener el frente de Pareto, este planteamiento se basó en un taller de trabajo 10*10 modificado para mostrar la efectividad del algoritmo.

Dai et al., (2013) propusieron un modelo de optimización con el fin de minimizar el consumo de energía y el Makespan para el problema flexible Flow shop (FFS) con restricciones de transporte, luego, desarrollaron un algoritmo genético mejorado (Modified Genetic Algorithm, MGA) para obtener soluciones de Pareto y finalmente, evaluaron el rendimiento del modelo y el algoritmo propuesto, como resultado, pudieron resolver el problema de manera efectiva y eficiente; en el 2013 Dai et al., abordan un modelo matemático de programación de enteros mixtos para el JSP de manera que el tiempo de inactividad total del cronograma dado de las máquinas herramientas que trabajan a diferentes velocidades de corte cambien con el objetivo de minimizar el consumo de energía, seguidamente, utilizan un Algoritmo Híbrido Genético con Recocido Simulado (Hybrid Genetic Simulated Annealing Algorithm, HGSA) para explorar la solución óptima con el fin de mejorar la asignación de velocidades y finalmente, la efectividad del enfoque se realiza en instancias de pequeño y gran tamaño, respectivamente.

May et al., (2015) centraron sus esfuerzos en estudiar los efectos de las políticas de programación dirigidas al mejoramiento de la productividad y el desempeño ambiental de los talleres de trabajo, para ello propusieron un Algoritmo Genético (Genetic Algorithm, GA) verde que les permitió evaluar el problema con objetivos múltiples relacionados con la sostenibilidad entre los que se encuentran la reducción del consumo de energía y el Makespan.

Coca et al., (2019) establecieron un enfoque para programar sistemas *Job shop*, en el que evaluaron variables ambientales, sociales y económicas, estructurando la metodología en catorce pasos que cubren aspectos tales como diseño del método multi-objetivo, fundamentos de los algoritmos genéticos y comparación con otros métodos tales como reglas de programación o el método del Cuello de Botella Móvil (CBM).

Xu et al., (2016) presentaron un modelo conjunto de objetivos múltiples de consumo de energía y eficiencia de producción relacionado con las condiciones múltiples de los servicios de equipos de fabricación. Para resolver el problema de objetivos múltiples, proponen un algoritmo mejorado de abejas basado en Pareto (Enhanced Pareto-based Bees Algorithm, EPBA), el método propuesto es evaluado y muestra un mejor rendimiento en escenarios estáticos y dinámicos en comparación con los algoritmos de optimización existentes; durante ese mismo año Zhang y Xhiong (2016) propusieron un algoritmo genético (Genetic Algorithm, GA) incorporado con dos estrategias de desarrollo local con los objetivos de mejorar la calidad de la solución y minimizar el consumo de energía en un modelo típico de programación de *Job Shop* basado en un marco de escala de velocidad de la máquina.

Teniendo en cuenta el creciente interés por abordar el EJSP y su importancia, se plantea un algoritmo híbrido genético en el que se plantea aprovechar la búsqueda por exploración del algoritmo genético y la búsqueda local que ofrece el algoritmo Recocido Simulado lo cual permite potenciar los resultados del algoritmo genético.

Materiales y métodos

El enfoque de investigación es cuantitativo de tipo Investigación aplicada tecnológica dado a que sirve para generar conocimientos que se puedan poner en práctica en el sector productivo, con el fin de impulsar un impacto positivo en la vida cotidiana. La metodología con la que se abordó la presente investigación se adapta de la propuesta por Hillier y Lieberman (2010) para el estudio de la Investigación de Operaciones compuesta de 4 Fases: (1) Definición del problema y recolección de datos relevantes, (2) Formular el modelo matemático que representa el problema, (3) Desarrollar un algoritmo de solución y (4) Validación del algoritmo de solución.

En la fase de definición del problema y recolección de datos relevantes está compuesta por dos etapas: la primera es la definición del problema en la cual se realiza una revisión de literatura con el fin de establecer los estudios relacionados con el problema de la programación de la producción considerando eficiencia energética para posteriormente definir el planteamiento del problema y los artículos base para la selección de los datos para el desarrollo de los ejemplos numéricos.

Una vez establecido el problema se procede hacer la recolección de datos relevantes que sirvan de insumo para el modelo. Partiendo de lo anterior se plantea el siguiente problema de investigación compuesto por 2 subproblemas: la permutación de la operación y la selección de la velocidad, para esto se consideraron los siguientes supuestos: No se pueden

procesar trabajos simultáneamente en más de una máquina. Una máquina solo puede ejecutar una operación a la vez. No se aceptan preferencias una vez inicie el trabajo. La configuración y el breakdown no son considerados. Durante el procesamiento de una operación no es permitido hacer ajustes a la velocidad de la máquina. Cada máquina se detendrá solo cuando haya terminado todos los trabajos asignados. Durante los tiempos de espera por trabajo, la máquina estará en modo stand-by con un costo de consumo de energía por unidad de tiempo SE_k .

De acuerdo con Jiang et al., (2019) el *Job Shop con Eficiencia Energética* se puede definir de la siguiente manera: un conjunto de n trabajos $\{j_1, j_2, j_3, \dots, j_n\}$ debe ser procesado por un conjunto m de máquinas $\{r_1, r_2, r_3, \dots, r_m\}$. Cada trabajo j_i consiste en una secuencia de tareas $f_i(\theta_{i1}, \theta_{i2}, \dots, \theta_{if_i})$. Cada tarea θ_{i1} tiene un único requisito de máquina $r\theta_{i1}$ y aún debe determinarse un tiempo de inicio. Cada máquina puede trabajar con diferentes velocidades, la combinación de duración y energía se presenta mediante una tupla $\{p\theta_{i1}, e\theta_{i1}\}$. Un cronograma factible es una asignación completa de tareas de tiempos de inicio que satisface las siguientes restricciones: 1. Las tareas de cada trabajo se programan secuencialmente. 2. Cada máquina puede procesar como máximo una tarea en cualquier momento. 3. No se permite la preferencia. El objetivo es encontrar una programación de operaciones factible que minimice el tiempo de finalización de todas las tareas.

Existen i trabajos y k máquinas en el taller de trabajo, donde cada trabajo consta de o operaciones para ser procesadas. Cada máquina puede trabajar a niveles de velocidades ajustables representados por $v = \{v_1, v_2, \dots, v_d\}$. Se supone que existe un tiempo de procesamiento básico q_{ik} , donde el trabajo i está siendo procesado en la máquina k . Si la velocidad v_d es seleccionada por el trabajo i cuando se está procesando en la máquina k , el tiempo de procesamiento puede definirse como $P_{ikd} = \frac{q_{ik}}{v_d}$, en cuanto al costo de consumo de energía por unidad de tiempo se puede definir por E_{kD} . Si $v_{d'} > v_d$, entonces $E_{kd'} \times P_{ikd'} > E_{kd} \times P_{ikd}$ se cumple. Es decir, si una máquina trabaja a altas velocidades, el tiempo de procesamiento disminuirá, pero el costo de energía aumentará.

Una vez definido el problema, se diseña una representación de esta situación, que permita su análisis y solución. La forma convencional en que la investigación de operaciones logra este objetivo es mediante la construcción de un modelo matemático que represente la esencia del problema. Para esto se realizan las siguientes actividades: comprender las bases teóricas del problema *Job Shop con Eficiencia Energética* con el fin de determinar las variables de decisión, supuestos, parámetros, los objetivos y restricciones correspondientes al problema, definir el modelo matemático que represente el problema de investigación.

A_{kd} : Es el consumo de energía de la máquina k a una velocidad d .

O'_{ik} : Operación de procesamiento del trabajo i en la máquina k .

- C_{ik} : Tiempo de finalización de O'_{ik} P_{ikd} : Tiempo de procesamiento del trabajo i en la máquina k con una velocidad d .
- S_{ik} : Hora de inicio de O'_{ik} D_{ik} : Número de niveles ajustables de la máquina k .
- C_k : Tiempo de finalización de la máquina k .
- W_k : Carga de trabajo total de la máquina k .
- C_{max} : Makespan Q : Constante positiva muy grande.
- η : Costo relacionado al tiempo de finalización por unidad de tiempo. P_{ik} : Operación predecesora de O'_{ik} en el trabajo i .
- CP_{ik} : Tiempo de finalización P_{ik}

La ecuación (1) representa la función objetivo.

$$\min F = \sum_{i=1}^n \sum_{k=1}^m \sum_{d=1}^{D_k} A_{ikd} P_{ikd} x_{ikd} + C_{max} \quad (1)$$

La ecuación (2) representa la restricción que asegura que la velocidad de una máquina no pueda ser cambiada durante el procesamiento de una operación.

$$\sum_{d=1}^{D_k} x_{ikd} = 1, i = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (2)$$

La ecuación (3) representa la restricción que asegura la precedencia entre operaciones de un trabajo.

$$C_{ik} - \sum_{d=1}^{D_k} P_{ikd} x_{ikd} \geq CP_{ik}, i = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (3)$$

La ecuación (4) representa la restricción que garantiza que cada máquina pueda procesar una operación a la vez.

$$C_{lk} - C_{ik} + Q(1 - z_{ilk}) \geq \sum_{d=1}^{D_k} P_{lkd} x_{lkd}, i, l = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (4)$$

La ecuación (5) representa la restricción que garantiza que cada máquina pueda procesar una operación a la vez.

$$W_k = \sum_{i=1}^n \sum_{d=1}^{D_k} P_{ikd} x_{ikd}, k = 1, 2, \dots, m \quad (5)$$

La ecuación (6) representa la restricción de carga de trabajo de cada máquina

$$C_k = \max\{C_{ik}\}, i = 1, 2, \dots, n \quad (6)$$

La ecuación (7) representa la restricción de no negatividad.

$$C_{ik} \geq 0, i = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (7)$$

La ecuación (8) representa restricción lógica o binaria.

$$x_{ikd} = \{1, \text{si el trabajo } i \text{ es procesado en la máquina } k \text{ con velocidad } d. \\ 0, \text{de lo contrario. } i = 1, 2, \dots, n; k = 1, 2, \dots, m; d = 1, 2, \dots, D_k \quad (8)$$

La ecuación (9) representa restricción lógica o binaria.

$$z_{ilk} = \{1, \text{si el trabajo } i \text{ es procesado en la máquina } k \text{ antes del trabajo } l. \\ 0, \text{de lo contrario. } i, l = 1, 2, \dots, n; k = 1, 2, \dots, m; k = 1, 2, \dots, m \quad (9)$$

Fuente: Adaptado de Jiang et al., 2019

Algoritmo de solución. Una vez formulado el modelo matemático del problema en estudio, la siguiente etapa de un trabajo de Investigación de Operaciones consiste en desarrollar un procedimiento, por lo general en computadora, para obtener una solución a partir de este modelo en este caso se plantea diseñar un Algoritmo Híbrido Genético el cual se programó en el software matemático Matlab ®. Para el diseño del Algoritmo Híbrido Genético se tomó como base el Algoritmo Genético y se definió un algoritmo de búsqueda local como lo es recocido simulado, en este caso el algoritmo recocido simulado se ubicará al final de la última operación del algoritmo genético en

el que por medio de búsqueda local mejorar la población y de esta manera lograr buenos resultados en menos tiempo.

Algoritmo genético. La primera aparición de los Algoritmos Genéticos fue por medio del autor Jhon Holland (1975) al utilizar GA para la investigación formal de la evolución natural de los seres vivos y los mecanismos de adaptación natural, también, este término ha sido aplicado en diferentes áreas como la matemática, la ingeniería, la biología y ciencias sociales. Se basa principalmente en partir de una población inicial la cual está conformada por un número determinados de individuos, estos individuos son denominados

cromosomas y son la representación en forma vectorial del problema, a cada uno de estos individuos se les calcula la función de adaptación (fitness) y de acuerdo con este valor los individuos mejor adaptados son seleccionados para ser cruzados, es decir mezclar su información genética para generar la siguiente generación de individuos. Finalmente existe la probabilidad a lo largo de las generaciones que ocurra una mutación es decir un cambio espontáneo en algún individuo de la población.

La definición de función de aptitud (fitness) es simplemente el recíproco del valor de la función objetivo. El objetivo es encontrar una solución que minimice la amplitud multiobjetivo y el consumo energético. Siguiendo las reglas de NWAUF, nuestra función de aptitud $F(i)$ (10) es una combinación convexa entre los valores normalizados de makespan y el consumo de energía de la solución y tomando de referencia (Escamilla et al., 2016).

$$F(i) = \lambda * NormalMakespan(i) + (1 - \lambda) * NormalEnergia(i) \quad (10)$$

$$NormalMakespan(i) = \frac{Makespan(i)}{MaxMakespan} \quad (11)$$

$$NormalEnergia(i) = \frac{ConsumoEnergia(i)}{MaxconsumoEnergia} \quad (12)$$

Fuente: Adaptado de Escamilla et al., 2016

donde $\lambda \in [0, 1]$. NormalMakespan (11) es el makespan dividido por el valor de makespan máximo en la ejecución de un algoritmo genético cuando el valor de λ es igual a 0 (MaxMakespan). NormalEnergy (12) se calcula sumando la energía utilizada en la ejecución de todas las tareas, dividida la energía máxima (MaxEnergy). MaxEnergy es la suma de la energía necesaria para ejecutar todas las tareas a la máxima velocidad. Una vez configurado el parámetro λ para la función fitness (10), la población inicial se puede generar de una manera específica. Así, para $\lambda = 0$, la función objetivo solo se

enfoca en reducir el consumo de energía ($F = NormalEnergy$), por lo que la población inicial se puede generar aleatoriamente para ordenar las tareas, pero las velocidades correspondientes se fijan al valor más bajo. De la misma manera, si $\lambda = 1$, la función objetivo sólo está enfocada a reducir la marca ($F = NormalMakespan$), por lo que la población inicial también se puede generar aleatoriamente para ordenar las tareas, pero las velocidades correspondientes se fijan al valor más alto. Para $\lambda \in [0, 1]$, la velocidad de cada tarea puede ser generada apropiadamente.

En la **Figura 2** se visualiza la representación del problema consiste en una matriz en el que las fila 1 representa las operaciones de los trabajos donde cada repetición de cada número representa una operación es decir que si el trabajo 1 tiene 5 operaciones dentro del vector operaciones se van a

encontrar 5 números 1 y así con los demás trabajos en el ejemplo presentado en la imagen se evidencia que son 3 trabajos con 5 operaciones cada 1. En el vector inferior se asignan las velocidades con las que van a ser procesadas cada operación de cada trabajo.

Vector de operaciones (Op)	2	2	1	1	3	2	2	3	1	3	3	3	2	1	1
Vector de velocidades (V)	1	2	3	2	1	3	2	3	1	2	1	3	3	3	1

Figura 1. Representación de la solución adaptada al problema

Fuente: Adaptado de Escamilla et al., 2016

Selección por torneo. Con el fin de garantizar la aleatoriedad a la hora de seleccionar a los mejores individuos de la población se seleccionan a número un número k de individuos de la población los cuales van a competir en un torneo y el mejor va a ser seleccionado, este proceso se repite el número de individuos de la población, es decir, el número de torneo que se realizan van a ser igual al número de individuos de la población.

Operador de Cruce POX (Precedence Preserving Order-based). Este operador fue desarrollado por Lee y Yamakawa

(1998). El cual genera dos hijos a partir de dos padres respetando las restricciones de precedencia. Se selecciona el padre 1 y el padre 2, determinar para cada padre, el conjunto de trabajos que serán heredados a su correspondiente hijo. Para el padre 1, los trabajos {3,1} y para el padre 2, el trabajo 2. Los trabajos del padre 1, toman la misma posición en el hijo 1, igualmente para el trabajo del padre 2, toma la misma posición en el hijo 2. Luego se eliminan del padre 1 los trabajos obtenidos del padre 2 y se coloca el resultado en el primer hijo según el mismo orden de aparición y viceversa para el segundo hijo. El proceso se muestra en la **Figura 3**.

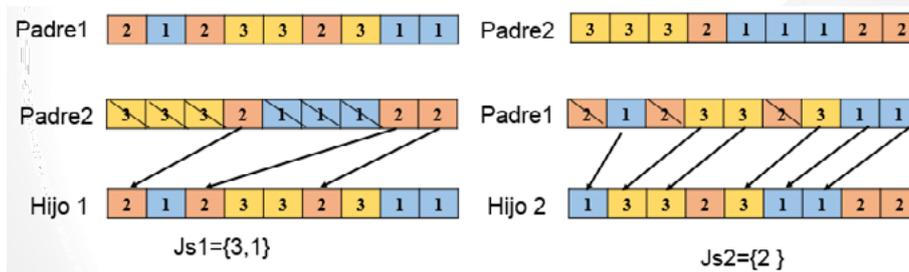


Figura 2. Operador de cruce POX

Fuente: Adaptado de Lee y Yamakawa, 1998

Operador de mutación PPS (Precedence Preserving Shift). Tomando como referencia el operador PPS que utilizaron He, Weng, y Fujimura (2017) en su investigación. El operador consiste en seleccionar un gen del vector de secuenciación de manera aleatoria, luego, el gen se desplaza a otra posición, mientras que la operación que este gen representa en su trabajo no cambia. En la **Figura 4** se selecciona el tercer 3, que representa la tercera operación del trabajo

3, por lo tanto solo se puede desplazar entre el rango del segundo 3 y el tercer 3, los cuales representan la segunda y tercera operación del trabajo 3. Como mencionan los autores He et.al, para este caso si el 3 se selecciona fuera de este rango, antes del segundo 3, entonces el 3 se convertiría en el segundo 3, o en el primer 3, y ya no representaría la tercera operación del trabajo 3, esta situación influiría mucho en la estructura del cromosoma original.

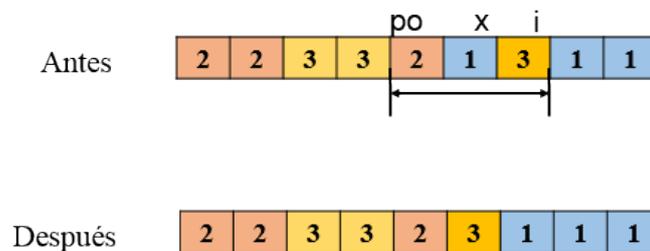


Figura 3. Operador de mutación PPS

Fuente: Adaptado de He, Weng, y Fujimura, 2017

Recocido Simulado. El recocido simulado o el Simulated Annealing (SA) es un algoritmo de búsqueda local que fue concebido por S. Kirkpatrick, C. D., Gelatt Jr, y M. P.

Vecchi (1982), consiste principalmente en emular un proceso térmico del temple de los metales, en el que a temperaturas altas las partículas se mueven en busca de

puntos de mínima energía cambiando su posición, el cual mediante un proceso de temple va reduciendo su temperatura de forma controlada, cristalizando el metal limitando el movimiento de las partículas. En este caso se genera una solución inicial del problema un vector solución de forma aleatoria, posteriormente se le va generar un vecindario el cual consiste en realizar un leve cambio en la estructura de la solución inicial, generando un entorno de las posibles soluciones que proporcionan puntos con funciones objetivos más altas o bajas, toma como referencia que si la solución comparada tiene una función objetivo

menor esta automáticamente se convierte en la nueva mejor solución, sin embargo, si la función objetivo de la nueva solución comparada es peor se tiene una probabilidad de aceptación definida por la ecuación de Boltzmann (13) relacionada principalmente con la temperatura y dependiendo de este valor se aceptan soluciones peores en el proceso, esto le permite al recocido simulado escapar de óptimos locales siendo una ventaja significativa en la búsqueda de la mejor solución. En la **Figura 6** se muestra el diagrama del flujo del Algoritmo Recocido Simulado.

$$Probabilidad\ aceptación = exp\ exp\left(\frac{-F(y) - F(x)}{T}\right) \quad (13)$$

Estructura de vecindario. La estructura de vecindario consiste en seleccionar aleatoriamente 2 operaciones e intercambiar la posición en la **Figura 5** se muestra el proceso-

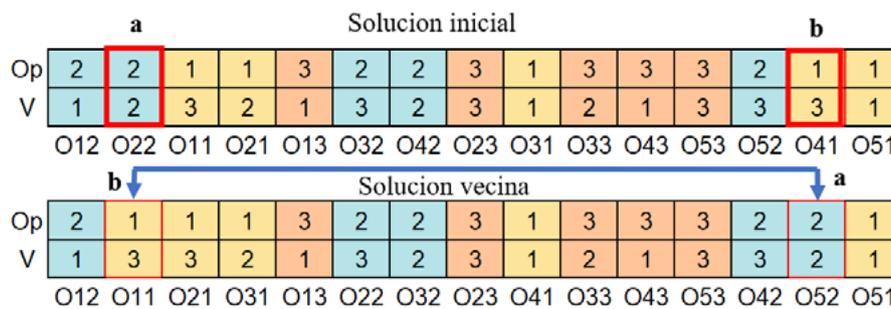


Figura 4. Generación de una solución vecina

Fuente: Elaborado por autores

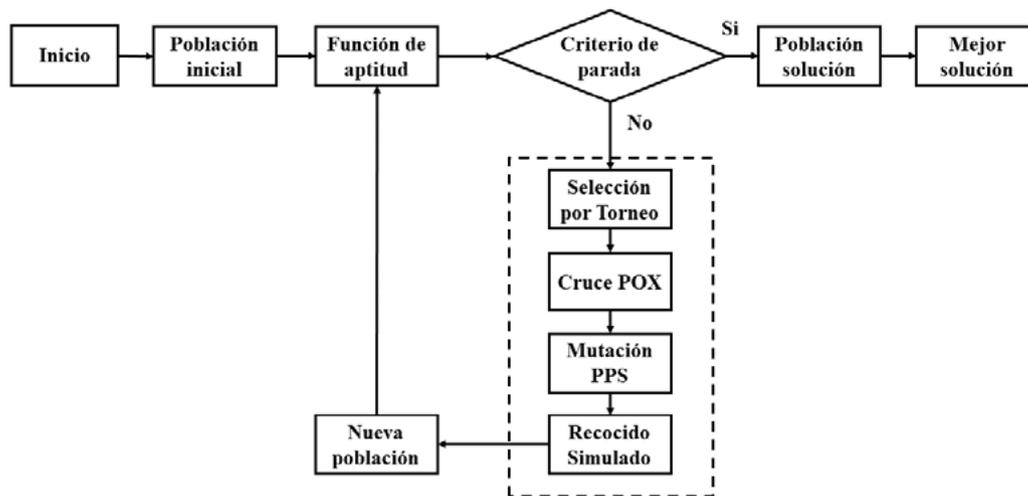


Figura 5. Diagrama de flujo Algoritmo Híbrido Genético propuesto

Fuente: Elaborado por autores

Resultados

Se observa en forma de cuadro comparativo los resultados obtenidos en la presente investigación a través del Algoritmo Híbrido Genético y los obtenidos por Escamilla J., Salido, Giret y Barber (2016) por medio de la evaluación del Algoritmo Genético (AG) y un solucionador comercial IBM optimizador de CP (CP Optimizer) el cual incorpora técnicas de propagación de restricciones de gran potencia y un método de búsqueda de vecindario adaptativo. Se evaluaron instancias de diferentes tamaños con el fin de evaluar el rendimiento del algoritmo propuesto, se dividieron en tres tamaños de instancias pequeñas medianas

y grandes. Estas instancias Benchmarking de la literatura fueron proporcionadas por el estudio de Escamilla J., Salido, Giret y Barber (2016), las cuales están conformadas por 10 problemas cada una, es decir se van a evaluar 10 problemas diferentes de cada tamaño con el fin de tener un panorama amplio en la evaluación del algoritmo propuesto. Cada problema es evaluado variando de 0 a 1 con un paso de 0.1 lo cual permite ver el comportamiento de cuándo se va cambiando el peso de los objetivos. En las siguientes tablas se muestran los resultados promedios de cada tamaño de problemas para las variables Makespan, Consumo de energía y la función Normal que representa el balance de los 2 objetivos.

Tabla 1. Comparación de resultados Algoritmo Híbrido Genético Vs CP Optimizer y AG, para instancia Agnetis 3,5,10.

Algoritmo Híbrido Genético (AHG)				CP Optimizer			Algoritmo Genético (AG)		
λ	k	E	F.F	Mk	E	F.F	Mk	E	F
0	60,7	90,3	0,59	71,4	84,4	0,53	65,8	84,4	0,55
0,1	52	90,2	0,57	65,2	84,5	0,55	65,2	84,5	0,55
0,2	50	89,9	0,56	64,4	84,7	0,55	64,4	84,7	0,55
0,3	49,1	90,4	0,54	63,2	85,2	0,55	63,2	85,2	0,55
0,4	46,8	92,6	0,53	59,7	88,1	0,55	59,7	88,1	0,55
0,5	43,3	99,1	0,51	53,9	94,3	0,54	54,2	93,9	0,54
0,6	42,9	101,5	0,49	48,4	104,2	0,52	48,9	103,2	0,52
0,7	39,5	108,2	0,45	45,3	111,9	0,50	45	112,7	0,50
0,8	36,9	120,7	0,41	42,2	123,4	0,46	42,2	123,5	0,46
0,9	36,5	126,1	0,37	41	133,2	0,41	41	133,7	0,41
1	35,6	132,4	0,31	41	143,1	0,36	41	145,3	0,36

Nota: Mk: makespan, E: energía y F.F: función fitness

Fuente: Elaborado por autores

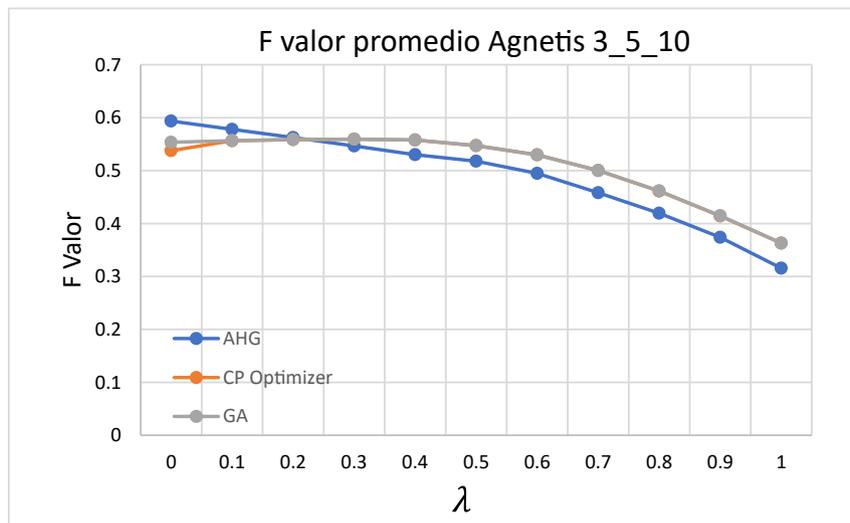


Figura 7. Gráfica F Valor Vs λ Algoritmo Híbrido Genético, CP Optimizer, Algoritmo genético

Fuente: Elaborado por autores

Tabla 2. Comparación de resultados Recocido Simulado Vs CP Optimizer y AG, para instancia Agnetis 3,7,100

Algoritmo Híbrido Genético (AHG)			CP Optimizer			Algoritmo Genético (AG)			
λ	Mk	E	F	Mk	E	F	Mk	E	F
0	843,9	1496,7	0,51	1088,4	1571,4	0,53	1006,3	1571,4	0,53
0,1	772,7	1539,7	0,52	999,3	1572,6	0,54	999,3	1572,6	0,54
0,2	753,1	1529,9	0,51	987,2	1576,5	0,54	987,2	1576,5	0,54
0,3	743,7	1544,8	0,50	922,2	1613,3	0,55	926,9	1610	0,55
0,4	710,8	1576,8	0,49	885,9	1649,2	0,55	891	1642,9	0,55
0,5	683,5	1649,1	0,49	838,8	1716	0,54	847,5	1704,8	0,51
0,6	642,6	1753,7	0,47	779,1	1859,3	0,53	782,4	1845,7	0,53
0,7	606,4	1876,1	0,45	708,5	2068,4	0,51	703,9	2099,5	0,51
0,8	573,2	2038,8	0,41	651,8	2346	0,47	642,4	2418,9	0,47
0,9	566,8	2098,4	0,38	626	2560,7	0,42	626	2573,3	0,42
1	553,5	2242,2	0,33	625,9	2664,1	0,37	626,9	2773,4	0,37

Nota: Mk: makespan, E: energía y F.F: función fitness

Fuente: Elaborado por autores

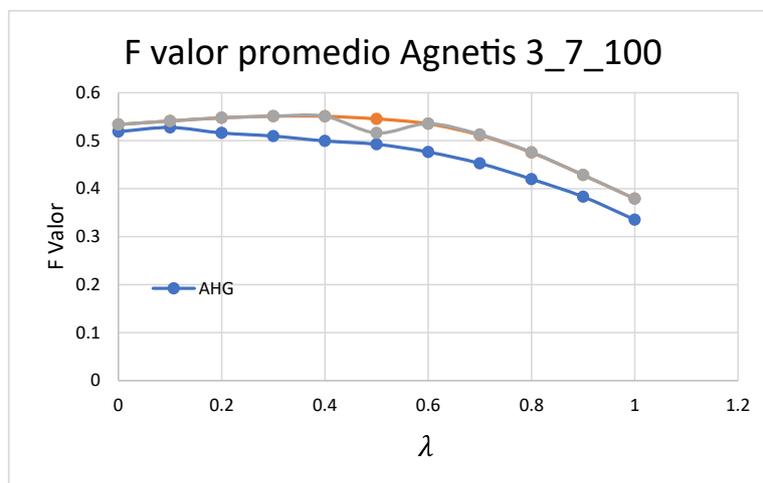


Figura 8. Gráfica F Valor Vs Algoritmo Híbrido Genético, CP Optimizer, Algoritmo genético

Fuente: Elaborado por autores

Tabla 4. Comparación de resultados Recocido Simulado Vs CP Optimizer y AG, para instancia de Instancia 3,25,100

λ	Algoritmo Híbrido Genético (AHG)			CP Optimizer			Algoritmo Genético (AG)		
	Mk	E	F	Mk	E	F	Mk	E	F
0	2393,4	4309,4	0,61	3160	3827,1	0,53	3096	3829	0,53
0,1	2219	4462,1	0,62	2768,1	3827,6	0,53	2781,7	3827,9	0,53
0,2	2098,9	4543,2	0,60	2719,3	3842,5	0,54	2764,8	3845,5	0,54
0,3	2064	4501,5	0,58	2597,9	3904,6	0,54	2657,3	3920,2	0,54
0,4	1967,2	4703,3	0,57	2480,7	4005,8	0,54	2495,3	4068,7	0,54
0,5	1910,3	4760,3	0,54	2342	4181,6	0,53	2317,1	4257,3	0,53
0,6	1839,7	4912,6	0,51	2147	4548,6	0,52	2118,3	4617,4	0,52
0,7	1766,1	5173,4	0,48	1935,5	5075,6	0,49	1943,7	5097,4	0,49
0,8	1732,1	5295,5	0,44	1806,2	5666	0,45	1791,4	5726,2	0,45
0,9	1700,6	5550,7	0,40	1725,9	6251,3	0,40	1732,2	6311	0,41
1	1681,9	5665,2	0,35	1673,4	6732,2	0,34	1711,8	6797,2	0,35

Nota: Mk: makespan, E: energía y F.F: función fitness

Fuente: Elaborado por autores

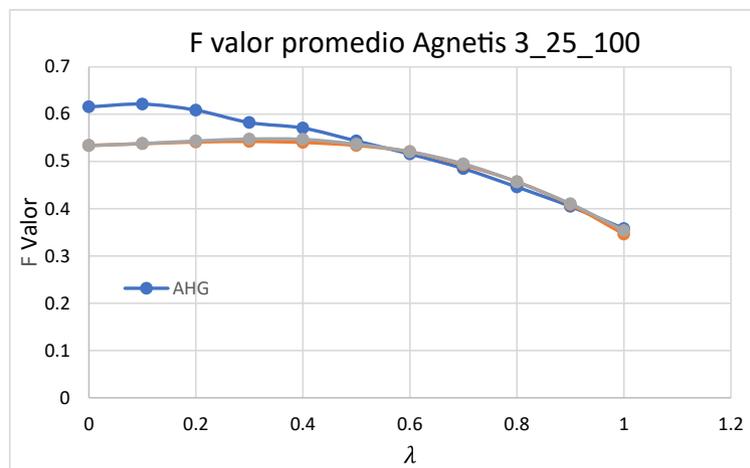


Figura 9. Gráfica F Valor Vs λ Algoritmo Híbrido Genético, CP Optimizer, Algoritmo genético

Fuente: Elaborado por autores

Discusión

Las nuevas tendencias en la producción hacen que las empresas busquen un nuevo enfoque más sostenible, haciendo que las líneas que ya se tenían en la programación de operaciones en sistemas tipo taller *Job shop* se ajusten ya no solo es interés de minimizar el tiempo de programación de la producción sino que también se busquen alternativas de solución en el que se incluya además la reducción del consumo de energía realizando una correcta asignación de velocidades a las máquinas con el fin de encontrar un equilibrio entre el consumo de energía y el tiempo de operación debido a que a mayor velocidad mayor consumo de energía y menor tiempo de operación. Se evidencia que hay un alto interés en la comunidad científica en este nuevo estudio de la adaptación de algoritmos a esta nueva problemática con el fin de analizar el rendimiento y establecer nuevas alternativas de solución.

El problema tratado en esta investigación se establece como la programación del taller *Job Shop* con eficiencia energética en el que las máquinas pueden consumir diferentes cantidades de energía para procesar tareas a diferentes velocidades. Es una extensión del clásico problema de programación del taller *Job Shop*, donde cada operación debe ser ejecutada por una máquina y esta máquina tiene la posibilidad de trabajar a diferentes velocidades. En este artículo se presenta un Algoritmo Híbrido Genético conformado por un Algoritmo Genético y

un Algoritmo complementario Recocido Simulado para modelar y resolver este problema. Se aprovechó la búsqueda global por exploración del Algoritmo Genético y la búsqueda local para mejorar las soluciones a partir del algoritmo Recocido Simulado.

Se realizó un estudio comparativo para analizar el comportamiento de nuestro Algoritmo Híbrido Genético frente a un conocido solucionador: IBM ILOG CPLEX CP Optimizer y el Algoritmo Genético propuesto por Escamilla J., Salido, Giret & Barber, (2016). La evaluación muestra que el algoritmo propuesto tuvo un mejor comportamiento al de CP Optimizer y el Algoritmo Genético propuesto por Escamilla J., Salido, Giret & Barber, (2016).

Conclusiones

El algoritmo propuesto puede ser útil para ser aplicado en problemas de programación a gran escala. Se pudieron evaluar diferentes tamaños de problemas en el que se permitió ajustar el peso de cada objetivo con el fin de brindar diversidad de soluciones siendo flexible para la toma de decisiones. Para futuras investigaciones se puede evaluar a cada algoritmo por separado con el fin de determinar el rendimiento independiente al igual que realizar híbrido con otros algoritmos de búsqueda local como el algoritmo de búsqueda Tabú o el algoritmo Búsqueda de vecindario variable VNS y explorar los nuevos algoritmos alternativos con el fin de evaluar su rendimiento.

Referencias

- Carlos Sornoza-Bravo, J. I., & Felipe Sabando-Piguabe, L. I. (2021). Estado del arte de la gestión energética en la industria Ciencias técnicas y aplicadas. *Dominio de Las Ciencias*, 7(3), 778–802. <https://doi.org/10.23857/dc.v7i3.1966>
- Cheng, Y., Tao, F., Liu, Y., Zhao, D., Zhang, L., & Xu, L. (2013). Energy-aware resource service scheduling based on utility evaluation in cloud manufacturing system. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 227(12), 1901–1915.
- Coca, G., Castrillón, O. D., Ruiz, S., Mateo-Sanz, J. M., & Jiménez, L. (2019). Sustainable evaluation of environmental and occupational risks scheduling flexible job shop manufacturing systems. *Journal of Cleaner Production*, 209, 146–168.
- Dai, M., Tang, D., Giret, A., Salido, M. A., & Li, W. D. (2013). Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, 29(5), 418–429.
- Escamilla, J., Salido, M. A., Giret, A., & Barber, F. (2016). A metaheuristic technique for energy-efficiency in job-shop scheduling. *The Knowledge Engineering Review*, 31(5), 475–485.
- Ham, A., Park, M. J., & Kim, K. M. (2021). Energy-Aware Flexible Job Shop Scheduling Using Mixed Integer Programming and Constraint Programming. *Mathematical Problems in Engineering*, 2021. <https://doi.org/10.1155/2021/8035806>
- He, Y., Weng, W., & Fujimura, S. (2017). Improvements to genetic algorithm for flexible job shop scheduling with overlapping in operations. *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, 791–796.
- Hidalgo, D. H., & Perez, Y. (2016). Eficiencia energética, competitividad empresarial y economía verde. *Revista Publicando*, 3(9), 447–466. <https://doi.org/10.1016/j.rser.2016.05.090>
- Hillier, F. S., & Lieberman, G. J. (2010). *Introducción a la investigación de operaciones*. Mc Graw Hill.
- Jiang, T., Zhang, C., & Sun, Q. M. (2019). Green Job Shop Scheduling Problem with Discrete Whale Optimization Algorithm. *IEEE Access*, 7, 43153–43166. <https://doi.org/10.1109/ACCESS.2019.2908200>
- Kirkpatrick, S. (1982). Optimization by Simulated Annealing. *Science*, 220, 671.
- Lee, K.-M., Yamakawa, T., & Lee, K.-M. (1998). A genetic algorithm for general machine scheduling problems. *1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No. 98EX111)*, 2, 60–66.

- Li, W., He, L., & Cao, Y. (2021). Many-objective evolutionary algorithm with reference point-based fuzzy correlation entropy for energy-efficient job shop scheduling with limited workers. *IEEE Transactions on Cybernetics*.
- Liu, Y., Dong, H., Lohse, N., Petrovic, S., & Gindy, N. (2014). An investigation into minimising total energy consumption and total weighted tardiness in job shops. *Journal of Cleaner Production*, *65*, 87–96.
- May, G., Stahl, B., Taisch, M., & Prabhu, V. (2015). Multi-objective genetic algorithm for energy-efficient job shop scheduling. *International Journal of Production Research*, *53*(23), 7071–7089.
- Mouzon, G., & Yildirim, M. B. (2008). A framework to minimise total energy consumption and total tardiness on a single machine. *International Journal of Sustainable Engineering*, *1*(2), 105–116.
- Wang, J., Liu, Y., Ren, S., Wang, C., & Wang, W. (2021). Evolutionary game based real-time scheduling for energy-efficient distributed and flexible job shop. *Journal of Cleaner Production*, *293*. <https://doi.org/10.1016/j.jclepro.2021.126093>
- Xu, W., Shao, L., Yao, B., Zhou, Z., & Pham, D. T. (2016). Perception data-driven optimization of manufacturing equipment service scheduling in sustainable manufacturing. *Journal of Manufacturing Systems*, *41*, 86–101.
- Zhang, H., Xu, G., Pan, R., & Ge, H. (2022). A novel heuristic method for the energy-efficient flexible job-shop scheduling problem with sequence-dependent set-up and transportation time. *Engineering Optimization*, *54*(10), 1646–1667.
- Zhang, R., & Chiong, R. (2016). Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *Journal of Cleaner Production*, *112*, 3361–3375.