

**Solución al “Problema Job Shop con Eficiencia Energética” Mediante la Metaheurística
Recocido Simulado**

Angie Carolina Ruiz Osorio y Camila Andrea Toro Correa

Trabajo de grado para optar por el Título de Ingeniería Industrial

Director:

**Carlos Eduardo Díaz Bohórquez
Magister en Ingeniería Industrial**

Codirector:

**Fabián Alexander Torres Cárdenas
Magister en Ingeniería Industrial**

**Universidad Industrial de Santander
Facultad de Ingenierías Físico-mecánicas
Escuela de Estudios Industriales y Empresariales
Bucaramanga**

2021

A Dios por guiarme en el camino correcto,
A mi mamá por siempre confiar en mí y ser mi apoyo,
A mis dos papas por el apoyo y ejemplo que me brindan,
A mis amigos y amigas por cada momento vivido,
A mi pareja por confiar en mi aun cuando yo no lo hacía,
A Coco que me daba calma,
A mi amiga y compañera de proyecto por su optimismo y apoyo,
A nuestro director y codirector por su guía y apoyo.

Angie Carolina Ruiz Osorio

A Dios por dame la sabiduría para culminar esta etapa de mi vida.
A mi mamá por amarme, apoyarme siempre y creer en mí y en todas las decisiones que he
tomado en mi vida,
A mi mamá Alicia por el amor y por el hogar que mi brindó en esta etapa académica de mi vida,
A mi papá por su apoyo y amor,
A mi esposo y a la familia de León por apoyarme en esta última etapa académica y permitirme
pertenecer a esta familia,
A mi compañera de proyecto de grado y amiga Angie Ruiz por apoyarme y animarme cuando
más lo necesitaba,
A nuestro director Profesor Carlos Díaz y Co director Fabián Torres por su apoyo para la
realización de este proyecto de grado.

Camila Andrea Toro Correa

Tabla de contenido

Introducción	12
Cumplimiento de Objetivos	14
1. Generalidades del proyecto	15
1.1 Objetivos.....	15
1.2 Planteamiento del Problema	16
1.3 Justificación del proyecto	17
2. Revisión de la Literatura	18
2.1. Análisis Preliminar de la Literatura	18
2.1.1 Enfoque Mono-Objetivo.....	19
2.1.2 Enfoque Multi-Objetivo..	20
3. Marco de Antecedentes	26
4. Marco Teórico	27
4.1 Optimización Combinatoria.....	27
4.2 Clases de complejidad computacional	28
4.2.1 Escala para medir la complejidad.....	29
4.3 Espacios de Soluciones	29
4.3.1 Planificación Semiactiva.	29
4.3.2 Planificación Activa	30
4.3.3 Planificación densa.....	30
4.4 Clases de Programación.....	30
4.4.1 Schedule Activo.....	30
4.4.2 Schedule Semiactivo	30
4.4.3 Schedule sin retraso (Non-Delay).	30
4.5 Problemas de programación de operaciones.....	30
4.5.1 Patrones de Flujo.	32

4.5.2 Job Shop con Eficiencia energética	34
4.6 Métodos Exactos.....	35
4.7 Métodos Aproximados.....	35
4.7.2 Métodos Metaheurísticos.....	36
5 Recocido Simulado	38
5.2 Elementos del SA.....	39
5.3 Parámetros de entrada del Algoritmo	40
5.3.1 Temperatura inicial	40
5.3.2 Temperatura final.	41
5.3.3 Estructura de generación de Vecindario.....	41
5.3.4 Velocidad de enfriamiento.	41
5.3.5 Tasa de disminución de Temperatura.....	41
5.3.6 Función de variación de Temperatura..	41
6 Problema Job Shop con Eficiencia Energética- EJSP	42
6.1 Descripción del problema	42
6.2 Formulación.....	43
6.2.1 Conjuntos del modelo.....	43
6.2.2 Parámetros del modelo	43
6.2.3 Variables del modelo	43
6.2.4 Función objetivo.....	44
6.2.5 Restricciones.....	44
6.2.6 Ejemplo.....	45
6.3 Desarrollo del algoritmo	47
6.3.4 Solución inicial.....	47
6.3.5 Makespan.....	48

6.3.6 Consumo de Energía.....	49
6.3.7 Parámetros de entrada del Recocido Simulado.	50
6.4 Función Fitness	51
6.5 Diagrama de flujo del Recocido Simulado	53
7. Validación del algoritmo Propuesto.....	55
7.1 Análisis de Varianza	56
7.2 Análisis de Varianza para la instancia ejemplo 3,7,10	58
7.3 Análisis de Varianza para la instancia Ejemplo 50,20,20.....	60
8. Resultados	63
9. Conclusiones	71
10. Recomendaciones	72
Referencias Bibliográficas	73

Lista de Tablas

Tabla 1. Cumplimiento de objetivos.....	14
Tabla 2. Funciones de Variación de Temperatura.....	41
Tabla 3 Número de operaciones.....	46
Tabla 4 Matriz de máquinas disponibles.....	46
Tabla 5 Matriz de velocidades.....	46
Tabla 6 Matriz de procesamiento.....	47
Tabla 7 Matriz de energía usada.....	47
Tabla 8 Solución inicial para el ejemplo 3,3,5.....	47
Tabla 9 Instancias usadas.....	55
Tabla 10 Factores y niveles del diseño experimental para las pequeñas instancias.....	55
Tabla 11 Análisis de Varianza para la instancia Ejemplo 3,3,5.....	56
Tabla 12 Análisis de Varianza para la instancia ejemplo 3,7,10.....	58
Tabla 13 Factores y niveles del diseño experimental para las medianas instancias.....	60
Tabla 14 Análisis de Varianza para la instancia ejemplo 50,20,20 de Watson.....	61
Tabla 15 Comparación de resultados Recocido Simulado Vs CP Optimizer y AG, para instancia Ejemplo 3,3,5.....	64
Tabla 16 GAP entre resultados de Recocido Simulado Vs CP Optimizer y AG para la instancia Ejemplo 3,3,5.....	64
Tabla 17 Comparación de resultados Recocido Simulado Vs CP Optimizer y AG, para instancia Ejemplo 3,7,10.....	65
Tabla 18 GAP entre resultados de Recocido Simulado Vs CP Optimizer y AG para la instancia Ejemplo 3,7,10.....	66
Tabla 19 Comparación de resultados Recocido Simulado Vs CP Optimizer y AG, para instancia de Watson Ejemplo 50,20,20.....	66
Tabla 20 GAP entre resultados de Recocido Simulado Vs CP Optimizer y AG para la instancia Watson Ejemplo 50,20,20.....	67
Tabla 21 Comparación de resultados Recocido Simulado Vs CP Optimizer y AG, para instancia de Watson Ejemplo 100,20,20.....	68

Tabla 22 GAP entre resultados de Recocido Simulado Vs CP Optimizer y AG para la instancia
Watson Ejemplo 100,20,20..... 69

Lista de Figuras

Figura 1. Identificación de problemas	28
Figura 2. Clasificación del Scheduling Problem según Hagar, Haddock y Heragu, 1995.	31
Figura 3. Representación en forma de grafo dirigido de un JSP.	33
Figura 4. Representación del diagrama de Gantt en un problema JSP.	33
Figura 5. Analogía de los parámetros termodinámicos y de optimización.....	38
Figura 6. Explicación de la estructura del vector solución inicial.	48
Figura 7. Diagrama de Gantt sin adelantar operaciones.	49
Figura 8. Diagrama de Gantt con adelanto operaciones.	49
Figura 9 Ejemplo de la estructura de vecindario	50
Figura 10. Diagrama de flujo del Recocido Simulado para las pequeñas y medianas instancias.	53
Figura 11. Diagrama de flujo del Recocido Simulado para las grandes instancias.	54
Figura 12. Diagrama de Pareto de efecto estandarizados para la instancia Ejemplo 3,3,5.....	57
Figura 13. Gráfica de efectos principales para la instancia Ejemplo 3,3,5	57
Figura 14. Diagrama de Pareto de efecto estandarizados para la instancia Ejemplo 3,7,10.....	59
Figura 15. Gráfica de efectos principales para la instancia Ejemplo 3,7,10	59
Figura 16. Diagrama de Pareto de efecto estandarizados para la instancia Ejemplo 50,20,20.....	61
Figura 17. Gráfica de efectos principales para la instancia Ejemplo 50,20,20.	62
Figura 18. Comportamiento de los valores promedio de la Función Fitness para las instancias pequeñas,.....	69
Figura 19. Comportamiento de los valores promedio de la Función Fitness para las instancias grandes	70

Lista de Apéndices

Ver Apéndices adjuntos y pueden ser visualizados en la base de datos de la Biblioteca UIS

Apéndice A. Revisión de Bibliométrica

Apéndice B. Código Vector Solución Inicial

Apéndice C. Código de Makespan adaptado

Apéndice D. Código del cálculo del vector de Energía

Apéndice E. Código del cálculo de Energía total

Apéndice F. Código de la Estructura del Vecindario

Apéndice G. Código de formulación de la Función Fitness

Apéndice H. Código Recocido Simulado

Apéndice I. Diseño de experimentos para la instancia Ejemplo 3,3,5

Apéndice J. Diseño de experimentos para la instancia Ejemplo 3,7,10

Apéndice K. Diseño de experimentos para la instancia de Watson Ejemplo 50,20,20

Apéndice L. Lista de resultados de la corrida individuales y cálculo de promedios

Apéndice M. Instancias usadas

Apéndice N. Artículo

Resumen

Título: Solución al problema “Job Shop con eficiencia energética” mediante la metaheurística del Recocido Simulado*

Autores:

Ruiz Osorio, Angie Carolina y Toro Correa, Camila Andrea**

Palabras claves:

Job Shop, Recocido Simulado, Eficiencia energética, Makespan

Descripción:

La presente investigación aborda un problema bi-objetivo, el clásico problema de Programación del taller de trabajo 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, por ejemplo, carpintería, taller de maquinaria mecánico, imprenta comercial y otras industrias que fabrican productos personalizados en lotes pequeños. Este problema es considerado NP-Hard debido a su complejidad computacional dado que el tiempo de cómputo que se requiere para resolver uno de estos problemas se incrementa conforme crece el tamaño del problema.

Para la solución de este problema se adaptó un algoritmo al problema Job Shop con Eficiencia Energética (EJSP), agregando la metaheurística del Recocido Simulado, vecindarios para la creación de soluciones aleatorias, cálculo del consumo de energía y finalmente la función fitness encontrada en la literatura para normalizar la función bi-objetivo de este problema.

Por medio de un diseño de experimentos 2^k , se determinaron los valores correspondientes a Tamaño de vecindario, Temperatura inicial y decremento de temperatura, estos fueron utilizados con el fin de identificar la influencia de cada factor en la función objetivo, buscando así valores mínimos en el algoritmo.

El algoritmo se ejecutó más de 400 veces, es decir, más de 100 por tamaño de instancia, los resultados fueron comparados con los obtenidos por los creadores de las instancias (Escamilla J., Salido, Giret y Barber).

*Proyecto de grado

**Facultad de Ingenierías Físico Mecánicas. Escuela de Estudios Industriales y Empresariales. Programa de Ingeniería Industrial. Director Magister en Ingeniería Industrial Carlos Eduardo Díaz Bohórquez

Abstract

Title: Solution to "Job Shop Problem with Energy Efficiency" Using Metaheuristic Simulated Annealing*

Authors:

Ruiz Osorio, Angie Carolina y Toro Correa, Camila Andrea**

Keywords:

Job-Shop, Simulated Annealing, Energy Efficiency, Makespan

The current research addresses a bi-objective problem, the classic Job Shop Scheduling problem with the extension of energy efficiency which seeks to minimize makespan 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, for example, carpentry, mechanical machine shop, printing press. commercial and other industries that manufacture custom products in small batches. This problem is considered NP-Hard due to its computational complexity since the computation time required to solve one of these problems increases as the size of the problem grows.

To solve this problem, an algorithm was adapted to the Energy Efficiency Job Shop (EJSP) problem, adding the Simulated Annealing metaheuristics, neighborhoods for the creation of random solutions, energy consumption calculation and finally the fitness function found in the literature. to normalize the bi-objective function of this problem.

Through a 2^k design of experiments, the values corresponding to neighborhood size, initial temperature and temperature decrease were determined, these were used to identify the influence of each factor on the objective function, thus looking for minimum values in the algorithm.

The algorithm was executed more than 400 times, that is, more than 100 per instance size, the results were compared with those obtained by the creators of the instances (Escamilla J., Salido, Giret and Barber).

*Bachelor thesis

**Faculty of of Physical-Mechanical Engineering, School of Industrial and Management Studies. Industrial Engineering Program. Director Carlos Eduardo Díaz, Master in industrial engineering.

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 esta manera, es un reto poder mantener o aumentar su participación en el mercado en la actualidad buscando fortalezas competitivas; 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”.

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 manufacturadas es el aumento del precio de la energía y la tendencia actual de sostenibilidad (Liu, Dong, Lohse, Petrovic y Gindy, 2014), 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 (Abedi , Chiong, Noman y Zhang, 2017), con las que se pretende, dar solución a un solo objetivo como la optimización en el consumo de energía o a varios, entre los que se incluyen el tiempo de finalización del último trabajo para salir del sistema, también conocido como el Makespan, y el consumo total de energía, como es el caso del trabajo realizado por Mouzon, Yildirim, & Twomey (2007), el cual es considerado como uno de los pioneros en el campo de la programación de operaciones enfocado a eficiencia

energetica, debido a que durante este trabajo, los autores demostraron que el consumo de energia puede ser significativamente menor si las máquinas precedidas por aquellas con restricciones de cuello de botella se apagan cuando el tiempo de llegada entre operaciones es considerable. De igual manera, para cumplir con el objetivo de su trabajo, los autores desarrollaron un conjunto de reglas de despacho (Vallejos Cifuentes P. E., 2017).

Teniendo en cuenta el creciente interés por abordar el EJSP y su importancia, se plantea un algoritmo de Recocido Simulado para su solución que permita mejorar la calidad de las soluciones.

Cumplimiento de Objetivos

Tabla 1.

Cumplimiento de objetivos.

Objetivo	Cumplimiento	Pág.
Realizar una revisión de literatura sobre el problema Job shop, así como los diferentes enfoques propuestos por distintos autores relacionado con la eficiencia energética	Capítulo 2 y Apéndice A	18
Diseñar un algoritmo Recocido Simulado y programar en MATLAB, como solución al problema Job shop con eficiencia energética.	Capítulo 5 y Apéndices desde el B hasta el L.	38
Validar al algoritmo propuesto mediante instancias de la literatura.	Capítulo 7	56
Elaborar un artículo de carácter publicable que contenga los resultados del proyecto de investigación.	Apéndice M	

1. Generalidades del proyecto

1.1 Objetivos

Objetivo General

Evaluar un algoritmo recocido simulado como alternativa de solución al problema Job Shop con eficiencia energética.

Objetivos Específicos

- Realizar una revisión de literatura sobre el problema Job shop, así como los diferentes enfoques propuesto por distintos autores relacionado con la eficiencia energética.
- Diseñar un algoritmo recocido simulado y programar en MATLAB, como solución al problema Job shop con eficiencia energética.
- Validar al algoritmo propuesto mediante instancias de la literatura.
- Elaborar un artículo de carácter publicable que contenga los resultados del proyecto de investigación.

1.2 Planteamiento del Problema

Las empresas manufactureras han centrado su atención en mejorar los indicadores del tiempo, costo y calidad, hasta hace poco tiempo el tema del medio ambiente no era una prioridad en las empresas, este enfoque ha venido cambiando en los últimos años ya que se han realizado llamados a nivel mundial sobre el cuidado y la preservación del medio ambiente, el efecto invernadero, el aumento de CO₂, y el calentamiento global. En el año 2010, el sector industrial fue responsable del 39,4 por ciento del consumo total de energía, el cual proviene en gran medida de las industrias manufactureras.

Con la urgencia de contribuir en la reducción del deterioro ambiental, la mayor parte de, los administradores y fabricantes de industrias manufactureras han planteado soluciones a nivel de máquina, diseñando e implementando maquinaria que contribuyan al ahorro de energía. Sin embargo, este hecho implica grandes inversiones por parte de la empresa, por esta razón, como se evidencia con el trabajo Mouzon, Yildirim y Twomey (2007), considerados como los pioneros en el campo de la programación de operaciones enfocado a la eficiencia energética, han planteado un modelo matemático multi-objetivo con el fin de disminuir el consumo de energía y Makespan, teniendo en cuenta varias reglas de despacho. A partir de la fecha, diferentes investigadores proponen la programación de la producción para reducir el consumo energético, esta estrategia no necesita de grandes inversiones para ser implementada por lo que ha llamado la atención en los últimos años (Ying Liu, 2013).

La presente investigación centra su atención en el problema de programación Job Shop con eficiencia Energética (EJSP), en el que las máquinas pueden trabajar con diferentes consumos de energía, este problema representa una extensión del problema clásico de programación del taller de trabajo Job Shop, con la variante de que cada operación puede ser ejecutada por una máquina y esta máquina puede funcionar a diferentes velocidades $v = \{v_1, v_2, v_3, \dots, v_d\}$ (Jiang, Zhang, Zhu, Gu, y Deng, 2018).

Formalmente, el EJSP 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 la programación de operaciones factible que minimice el tiempo de finalización de todas las tareas Makespan y la energía utilizada (Salido et al. 2013)

1.3 Justificación del proyecto

Según los datos estadísticos, la demanda mundial de energía se ha duplicado en los últimos 40 años y se duplicará una vez más para 2030. En el año 2010, el sector industrial fue responsable del 39,4 por ciento del consumo total de energía, el cual proviene en gran medida de las industrias manufactureras por tanto el ahorro de energía eléctrica equivale a disminuir el consumo de combustibles en la generación de electricidad evitando también la emisión de gases contaminantes hacia la atmósfera, teniendo así también una disminución en la parte de costos.

El problema EJSP propuesto en la presente investigación representa al clásico problema Job Shop, solo que en este caso además minimizar el Makespan también se tiene en cuenta las velocidades de las máquinas y el tiempo de procesamiento esto con el fin de minimizar el consumo de energía, por lo que el problema se plantea con un enfoque Bi-objetivo.

Durante la revisión de literatura se encontró que el problema EJSP empezó a ser un tema de estudio a partir del año 2003 y desde la fecha los diferentes autores han trabajado en resolverlo a través de métodos heurísticos y metaheurísticos, de las cuales se pueden destacar el algoritmo genético, optimización de colonias de hormigas, búsqueda tabú, optimización de enjambre por partículas y algoritmos híbridos. Cabe destacar que los artículos encontrados durante la revisión de literatura solo 2 estaban centrados en la minimización del Makespan y el consumo de energía cuándo el problema abordaba más de una máquina y diferentes velocidades.

Por ello el propósito de la presente investigación es buscar nuevos y diferentes aportes para contribuir a la solución de problema y así ser un nuevo punto de partida para futuras investigaciones.

2. Revision de la Literatura

2.1. Análisis Preliminar de la Literatura

En el análisis bibliométrico realizado (Ver apéndice 1), se obtuvieron un total de 21.781 artículos científicos (9595 Scopus, 2410 Web of Science, 9781 Science Direct), los cuales al ser filtraron con las palabras claves “Energy Efficiency in Job Shop Scheduling” dieron como resultado un total de 236 artículos (102 Scopus, 26 Web of Science, 108 Science Direct). Los países que más investigaciones de EJSP han realizado son China y Estados Unidos.

Usualmente, en cada 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. 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.

Durante el desarrollo de la presente investigación, se han encontrado diferentes enfoques que han utilizado los autores, entre los que se resaltan:

- Enfoque Mono-Objetivo
- Enfoque Multi-Objetivo

Adicionalmente 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) entre otras (Büyüksaatçi & Baray, 2014).

2.1.1 Enfoque Mono-Objetivo. Shrouf, Ordieres-Meré, García-Sánchez y Ortega-Mier, (2014) resolvieron el modelo matemático de optimización del costo de consumo de energía al determinar los tiempos de inicio para el procesamiento del trabajo, el tiempo de inactividad y el tiempo de “encendido” y “apagado” bajo la programación de una sola máquina, como resultado, lograron reducir significativamente el costo de consumo al evitar períodos de precios altos de energía. De igual manera, obtuvieron un efecto ambiental positivo al reducir el consumo de energía durante los períodos pico.

Así mismo Abedi , Chiong, Noman y Zhang (2017), usaron un modelo de programación lineal entera mixta, un Algoritmo Genético (Genetic Algorithm, AG), un Algoritmo de Optimización de Enjambre de Partículas (Particle Swarm Optimization, PSO) y un enfoque híbrido de PSO, para darle solución al problema de programación de máquinas individuales con eficiencia energética, deterioro acumulativo y mantenimiento múltiple, con el fin de minimizar el consumo de energía, para ese mismo año Wenbin, Yuxin, y Yi (2017) propusieron una serie de funciones analíticas para denotar la relación entre eficiencia energética y Makespan, de igual manera, desarrollaron un Algoritmo mejorado de Enjambre de Partículas (Particle Swarm Optimization, PSO), inspirado en el mecanismo hormonal, para resolver el problema de programación del EJSP.

Han, Zhigang, Yan, Hua y Yanhong (2018), también plantearon un método de optimización de ahorro de energía en dos etapas, pero este se enfocó en el problema Flexible Job Shop (FJSP) el cual es una generalización del Job Shop, con la diferencia de que este permite que cada tarea sea asignada a un grupo de máquinas en las cuales podrán ser procesadas. Durante el desarrollo la primera etapa aplicaron un Algoritmo Genético Modificado (Modified Genetic Algorithm, MGA) y en la segunda una secuencia de operación, adoptando un método híbrido que integra el Algoritmo Genético (Genetic Algorithm, GA) con la Optimización de Enjambre de Partículas (Particle Swarm Optimization, PSO).

Masmoudi, Delorme y Gianessi (2019) realizaron el más reciente trabajo en el que se considera el EJSP, cuyo objetivo es minimizar los costos de producción en términos de energía, respetando

al mismo tiempo una limitación de potencia máxima, junto con restricciones de producción más tradicionales. Los autores proponen dos modelos de programación lineal entera mixta para el problema abordado.

2.1.2 Enfoque Multi-Objetivo. Mouzon, Yildirim y Twomey (2007) propusieron un modelo de programación matemática multi-objetivo para determinar el ahorro de energía cuando las máquinas o equipos en cuello de botella se apagan al estar inactivos, asimismo, plantean algunas reglas de despacho, las cuales, según los autores, podrían influir en el consumo de energía de equipos subutilizados y el tiempo total de finalización. Al año siguiente, Mouzon y Yildirim (2008) resuelven el problema de minimización del consumo total de energía y tardanza total en una máquina, a través de un Metaheurístico de Búsqueda Adaptativa multi-objetivo Aleatorio Codicioso (Greedy Randomised Adaptative Search Procedure to minimize Total Energy Total Tardiness, GRASPTETT), descartando GRASP debido a que los autores indican que este no se ha utilizado ampliamente en la optimización de objetivos múltiples.

Cinco años más tarde Liu, Lohse, Petrovic y Gindy (2013) proponen un modelo para optimizar el costo, el consumo y la tardanza total de electricidad aplicando la política de apagón continuo (Rolling Blackout, RB), que consiste en cortes de energía eléctrica semanales, este fue solucionado empleando algoritmo genético de clasificación no denominado II (Non-dominated Sorting Genetic Algorithm, NSGA-II). Esta política suele ser usada en países desarrollados como China; Al siguiente año Liu, Dong, Lohse, Petrovic y Gindy (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 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, Tang, Giret, Salido y Li (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 2015 Tang y Dai, 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 que trabajan a diferentes velocidades de corte también con el objetivo de minimizar el consumo de energía, seguidamente, utilizan un Algoritmo Híbrido Genético con Recocido Simulado (Hybrid Genetic Simulated Annealing Algorithm, HGSAA) 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, Stahl, Taisch y Prabhu (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, desarrollaron diferentes políticas para controlar el comportamiento de los componentes de la máquina, decidir el estado energético bajo el que deberán operar (inactivo, abajo, configuración, mantenimiento o trabajando), dependiendo de la duración de cada período y la potencia de los diferentes estados, asimismo, 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.

La investigación de Tiwari y Liu (2015) se basa en una circunstancia del FJSP exponen un modelo para el problema de dos objetivos en el que se minimiza el consumo total de energía de procesamiento y el Makespan, del mismo modo, desarrollan un Algoritmo Genético modificado para resolver el problema de división del lote de materia prima, por último, presentaron un caso de estudio para mostrar la efectividad del modelo y la solución propuesta.

Coca, Castrillón, Ruiz, Mateo-Sanz y Jiménez (2016) 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, Shao, Yao, Zhou y Pham (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); con el fin de garantizar la variedad de la población, evitar la convergencia prematura y mejorar la

velocidad de búsqueda, se utilizan varias tecnologías clave, como la búsqueda de vecindad variable, la operación de mutación y cruce, búsqueda local de ruta crítica, conjunto de tabú de ruta crítica, entre otras, Finalmente, 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 Chiong (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, es decir, minimizar el consumo de energía y makespan en un modelo típico de programación de Job Shop basado en un marco de escala de velocidad de la máquina. Para darle solución al problema planteado llevaron a cabo experimentos computacionales exhaustivos para verificar la efectividad del enfoque de solución propuesto.

Posteriormente Tang, Dai, Salido y Giret (2016), plantearon un algoritmo basado en la optimización mejorada del enjambre de partículas, para resolver el problema Flexible Flow Shop Scheduling, el cual está relacionado con un grupo de máquinas en serie en el que cada trabajo debe ser procesado en cada una de las máquinas, y todos los trabajos tienen la misma ruta, el algoritmo tiene como objetivo la minimización del consumo de energía y el Makespan, como resultado obtuvieron que el algoritmo propuesto superó el comportamiento de los algoritmos de última generación; de igual forma Salido, Escamilla, Giret y Barber (2016) propusieron en ese mismo año un Algoritmo Genético (Genetic Algorithm, GA) para resolver el EJSP, en esta versión extendida del problema se considera que cada operación debe ser ejecutada por una máquina y esta a su vez puede trabajar con diferentes velocidades, durante la solución del problema los investigadores determinaron que el algoritmo propuesto pudo resolver todas las instancias con una buena calidad de solución, por otra parte, como complemento a esta investigación, en el mismo año Escamilla *et al.* (2016) presentaron un documento, el cual direccionaron hacia el análisis de tres objetivos importantes para el problema planteado, siendo estos, eficiencia energética, robustez y capacidad de fabricación, así como la relación entre ellos, determinando al final que: obtener soluciones eficientemente energéticas, supone obtener soluciones robustas y viceversa.

Che, Wu, Peng y Yan (2017) abordaron un problema de programación de una sola máquina con un mecanismo de apagado en el que los objetivos eran minimizar tanto el consumo total de energía como la tardanza máxima, del mismo modo, desarrollaron un modelo de programación lineal entera mixta (Mixed-Integer Linear Programming, MILP), empleando un método avanzado de

restricción ε con búsqueda local, preprocesamiento y desigualdades válidas. El método fue validado por instancias de referencia y generadas aleatoriamente.

Zhang, Tang, Wu y Wang (2017) formularon un nuevo modelo matemático entero no lineal mixto para el problema Flexible Job Shop con eficiencia energética (Energy-Efficient Flexible Job Shop Scheduling Problem EFJSP), con el fin de lograr una efectiva selección de máquina, la secuencia de trabajos y la toma de decisiones de encendido y apagado de la máquina, de igual manera, implementaron un algoritmo de Programación de Expresión Genética (efficient Gene Expression Programming, eGEP). Como resultado de los trabajos experimentales, obtuvieron que, con las decisiones de apagado de la máquina se logra una reducción significativa en el consumo total de energía. Yin, Li, Gao, Zhang y Lu (2017), plantearon un nuevo modelo de programación lineal entera mixta multi-objetivo teniendo en cuenta la productividad, la eficiencia energética y la reducción del ruido, para darle solución al problema Job Shop decidieron hacer uso de un algoritmo genético (Genetic Algorithm, GA) basado en un diseño reticular simple.

Wang, Huang y Li (2018) estudiaron un nuevo problema de programación de flujo híbrido de dos etapas, con máquinas de procesamiento simultáneo en la primera etapa y una sola máquina sin inactividad con una secuencia de trabajo predeterminada en la segunda, en este estudio los autores propusieron un modelo de programación entera mixta con el objetivo de minimizar el tiempo total de procesamiento para reducir el consumo de energía y mejorar la eficiencia de la producción.

Dai, Wang y Ji (2018) proponen un modelo de optimización integrada consciente de la energía para la planificación de procesos y la programación del taller considerando el transporte en un sistema de fabricación flexible. Para minimizar el consumo total de energía y el Makespan, adoptan un Algoritmo Genético de clasificación no dominado (Non-dominated Sorting Genetic Algorithm, NSGA-II) y finalmente, utilizan un estudio de caso que utiliza el modelo propuesto verificando que el consumo de energía del transporte tiene una influencia auténtica en el consumo total de energía y el esquema de programación.

Jiang, Zhang, Zhu, Gu y Deng (2018) prestaron atención al problema de programación FJSP, por ello, plantearon un modelo matemático del FJSP con bajas emisiones de carbono, con la intención de minimizar el costo de consumo de energía y el costo de tardanza, para la solución presentaron un tipo de algoritmo de Optimización de Enjambre de Gatos Discreto basado en Bi-poblaciones (Bi-population based Discrete Cat Swarm Optimization, BDCSO) para así obtener el

esquema de programación óptimo en el taller, durante el desarrollo descubrieron que con el algoritmo planteado se obtuvieron tiempos computacionales aceptables, que al ser comparados con otros, resultan ser mejores.

Jiang y Wang (2019) proponen un Algoritmo Evolutivo basado en la Descomposición (Modified Multiobjective Evolutionary Algorithm based on Decomposition MMOEA / D) para el problema de programación EEFJSP, con el objetivo de optimizar el consumo de energía total y temporal, los autores agregan al algoritmo una intensificación local basada en las propiedades de este problema para mejorar la explotación, finalmente, se llevan a cabo experimentos de comparación entre el MMOEA / D y el Algoritmo Revuelto Salto de Rana (Shuffled Frog-Leaping Algorithm, SFLA). Los resultados han demostrado que el MMOEA / D supera a SFLA en este problema.

Durante su investigación, Zhang, Wu, Peng y Jia (2018) propusieron un método de programación para darle solución al problema Flexible Job Shop con eficiencia energética (EFSS), considerando el consumo de energía de producción y el Makespan, de igual forma para darle solución a este, utilizaron un Algoritmo Genético de clasificación no denominado II (Non-dominated Sorting Genetic Algorithm, NSGA-II), el cual evita la interferencia de factores subjetivos, ayuda a la selección de una máquina adecuada y a obtener una secuencia operacional racional.

Zhang, Jiang, Zhu y Gu (2018) investigaron el EJSP, con el objetivo de minimizar el costo en el consumo de energía y el costo del tiempo de finalización, teniendo en cuenta la naturaleza del JSP tradicional deciden presentar un Algoritmo Mejorado de Optimización de Ballena (Improved Whale Optimization Algorithm, IWOA) considerando la utilización de tres estrategias de mejora como: reglas de despacho (Dispatching Rules, DR), un factor de convergencia (Nonlinear Convergence Factor NCF) y una operación de mutación (Mutation Operation, MO), siendo esta última empleada para reducir la posibilidad de caer en un óptimo local y evitar la convergencia prematura; como resultado obtuvieron que las estrategia propuestas mejoran el resultado computacional del algoritmo.

Zhou, Li, Du, Pang y Chen (2018) consideraron el problema de programación de una sola máquina de procesamiento por lotes (Batch Processing Machine, BPM) con llegadas de trabajo dinámicas y un esquema de precios, con el fin de minimizar simultáneamente el Makespan y el

consumo de electricidad (Total Electricity Cost, TEC) como índice de sostenibilidad ambiental, de igual forma, propusieron un modelo matemático de programación entera mixta y para darle solución a este, plantearon un algoritmo de Evolución Diferencial (Differential Evolution, DE), que al ser evaluado y comparado con en NSGA-II, obtuvieron como resultado que el algoritmo propuesto funciona mejor en términos de calidad y distribución de la solución.

Zineb Ibn, Barkany, Ikram, Jabri y Darcherif (2019) proponen una programación lineal entera mixta para integrar la planificación y la programación con la consideración del aspecto energético, a su vez, consideran una nueva restricción operativa en el nivel táctico debido al gran interés dado al consumo de energía y su fuerte vínculo existente con el sistema de producción, integraron el costo de energía en cada período de producción y consideraron implícitamente el consumo de energía de cada máquina en los períodos T con el objetivo de encontrar un compromiso entre el plan de producción y la programación para minimizar el costo total de producción.

En ese mismo año Tang, Dai, Giret y Salido (2019) modelaron el problema EJSP con limitaciones de transporte como un modelo matemático de programación lineal entera mixta resuelto a través de Algoritmo de Estimación de Distribución (Enhanced Estimation Of Distribution Algorithm, EEDA) y un diseño mejorado del algoritmo de Recocido Simulado (Simulated Annealing, SA).

Posteriormente González, Oddi y Rasconi (2019) plantearon 3 enfoques para darle solución al problema EJSP, estos fueron: un algoritmo evolutivo multi-objetivo basado en Pareto, que se hibrida con un método de búsqueda local y un paso de programación lineal, un algoritmo evolutivo multi-objetivo basado en descomposición hibridado y finalmente un enfoque de programación de restricciones.

Finalmente, Jiang , Zhang y Qi-Ming (2019) consideraron el problema EJSP, proponiendo para su solución un modelo matemático considerando máquinas a diferentes velocidades, asimismo para darle solución a este plantearon un algoritmo discreto de optimización de ballenas (Discrete Whale Optimization Algorithm, DWOA) integrando una estrategia de búsqueda de vecindario variable para mejor aún más la capacidad de la búsqueda.

3. Marco de Antecedentes

A continuación, se hace una descripción de 7 tesis de investigación desarrolladas en la Universidad Industrial de Santander (UIS) que se tomarán como guía durante el desarrollo del proyecto, puesto que estas contienen información relevante acerca del tema clave del JSP. Así mismo servirá para la definición y formulación del problema.

En el proyecto titulado “Método Particle Swarm Optimization (PSO - enjambre de partículas) aplicado al problema de múltiples objetivos del Job Shop Scheduling (JSP) o secuencia de máquinas”, se estudia el problema de la programación de tareas, del mismo modo, el autor realiza una revisión bibliográfica de los últimos métodos utilizados en la resolución del JSP, tema que contribuye al desarrollo y planteamiento de la revisión literaria logrando un avance en el objetivo número uno del presente proyecto (Sarmiento Ardila C. J., 2012).

Dentro de los objetivos de la investigación realizada en el proyecto “Un algoritmo memético para la minimización del makespan en el JSP”, se realiza una revisión bibliográfica sobre los métodos Metaheurísticos, dado que los autores proponen un Algoritmo Memético aprovechando las características del Algoritmo Genético buscando dar una solución al JSP, asimismo, en el proyecto, implementan métodos heurísticos y metaheurísticos que resultan ser muy eficientes arrojando soluciones aproximadas y en un tiempo computacional razonable, temas que se usan como referencia para el análisis de los diferentes métodos que se han usado para dar una solución al JSP (Aguilar Imitola & Pérez Diaz, 2013).

Pinzón, C. (2013) Utiliza una herramienta Software para el método PSO (Particle Swarm Optimization) en el que se toma como base el diagrama de MOPSO, dando respuesta a problemas del mundo real del tipo Job Shop sirviendo, así como soporte operativo en la planeación de las actividades y en la toma de decisiones propias de la industria manufacturera.

En la investigación “Estudio del impacto del tamaño de lote de transferencia en el desempeño de un sistema productivo tipo Job Shop considerando tiempos de transporte”, se aborda el problema de programación en un taller productivo tipo JSP para el cumplimiento del segundo objetivo del proyecto mencionado, debido a que se desarrolla un modelo de programación lineal entera mixta que consiste en un sistema de expresiones matemáticas relacionadas para determinar

la secuencia de operaciones que optimiza el desempeño de un sistema productivo tipo Job Shop. (Jaimes Sandoval & Rivera González, 2015).

Gomez, J., Orduz, E., (2015) diseñan un Algoritmo Genético y realizan una validación a través de las instancias encontradas acerca del FJSP con restricciones de transporte, esto, con el propósito de minimizar el makespan en el FJSP con restricciones de transporte utilizando Algoritmo Genético.

Aparicio, Y., González, C. (2016) Diseñan un algoritmo para dar solución al Job Shop Scheduling Problem tradicional bajo el objetivo del makespan teniendo como punto de partida la metaheurística GRASP, dentro de la revisión de literatura, incluyen el problema, el método de solución y la combinación de estos dos como base para la investigación, diseñan un algoritmo teniendo como resultado un híbrido compuesto por GRAPS y Algoritmo Genético.

Los autores del proyecto titulado “Solución al Flexible Job Shop Problem con tiempos de alistamiento dependientes de la secuencia mediante un algoritmo híbrido genético” donde parte de la población inicial se generó mediante la metaheurística Recocido Simulado, utilizando la selección por torneo, también, utilizaron dos tipos de operadores de cruce para modificar la secuencia de las operaciones (Precedence Preserving Orderbased, POX) y otro para la asignación de las máquinas (Precedence Preserving Shift, PPS), en el proyecto abordan el problema de programación de operaciones que involucra la secuenciación y asignación de máquinas, realizando un diseño de experimentos, y validando el algoritmo con instancias de literatura y metodologías propuestas por diversos autores (Lozano y Torres, 2018).

4. Marco Teórico

4.1 Optimización Combinatoria

La optimización es considerada una disciplina fundamental dentro del campo científico en áreas como la Informática, la Inteligencia Artificial o la Investigación Operativa. Por su parte la optimización combinatoria se clasifica dentro de esta disciplina en la categoría de soluciones que están codificadas por valores enteros y consiste en encontrar un objeto entre un conjunto finito (o al menos contable) de posibilidades. Este objeto suele ser un número natural (o conjunto de

naturales), una permutación o una estructura de grafo (o subgrafo). (Duarte Muñoz, Pantrigo Fernández y Gallego Carrillo, 2007)

4.2 Clases de complejidad computacional

Según el docente de la facultad de ingeniería de Sistemas e informática de la Universidad Nacional Mayor de San Marcos, Augusto Cortez, la teoría de la complejidad computacional es parte de la teoría de la computación, que estudia los recursos requeridos durante el cálculo para resolver un problema, es decir la complejidad del cálculo como la cantidad de recursos necesarios para efectuar este, entre los recursos más utilizados se encuentran el tiempo y el espacio (Cortez, 2004).

Para determinar el grado de complejidad que puede tener un problema, se hace uso del modelo computacional de la Máquina de Turing, la cual se puede tomar como punto de referencia para realizar la siguiente clasificación (Ver figura 1).

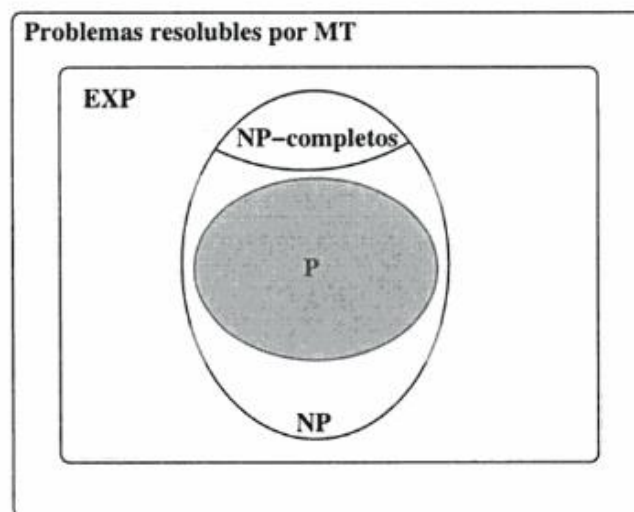


Figura 1. Identificación de problemas. Adaptado de Pérez, Jiménez y Fernando Sancho Caparrini. (2003). Maquinas Moleculares Basadas en ADN. pág. 32. Universidad de Sevilla. España.

4.2.1 Escala para medir la complejidad.

4.2.1.1 Complejidad P: un problema puede ser clasificado en esta clase de complejidad cuando, este puede ser resuelto por una máquina de Turing determinística en un tiempo polinómico.

4.2.1.2 Complejidad NP: si un problema puede ser resuelto en un tiempo polinómico por una máquina de Turing no determinística, el problema se puede clasificar como de complejidad NP.

4.2.1.3 Complejidad NP-Completo: es una subclase de la complejidad NP, donde los problemas que son los más difíciles de la clase, en el sentido de que cualquier otro problema de la clase, puede ser resuelto a través de él, con un coste en tiempo adicional de tipo polinomial. (*Pérez Jiménez & Sancho Caparrini, 2003*).

4.2.1.4 Complejidad NP-Hard: los problemas que clasifican en este tipo de complejidad son aquellos que resultan más difíciles que los NP. (*Ashrafi , Davoudpour y Abbassi, 2014*).

El problema Job Shop es una clase de problema optimización combinatoria del tipo NP-Hard, por lo que algunas técnicas de búsqueda no pueden encontrar una solución en un tiempo determinado. (*Abdolrazzagh-Nezhad y Abdullah, 2017*).

4.3 Espacios de Soluciones

El espacio de planificaciones factibles completo es grande y contiene soluciones que no son realmente interesantes, así que en la práctica se consideran espacios de búsqueda más reducidos y para esto se recurre a planificaciones semiactivas, Activas y densas (*Sierra Sánchez, 2010*).

4.3.1 Planificación Semiactiva. Esta planificación ocurre cuando para adelantar el tiempo de inicio de una tarea es necesario modificar el orden de al menos 2 (*Sierra Sánchez, 2010*).

4.3.2 Planificación Activa. Cuando se desea adelantar el tiempo de inicio de una tarea y para esto se requiere retrasar al menos una, se dice que el espacio de planificación es activo (Sierra Sánchez, 2010).

4.3.3 Planificación densa. Una planificación es densa si nunca se da la situación de que una máquina esta desocupada en el mismo instante en que una tarea debe ser procesada en dicha máquina (Sierra Sánchez, 2010).

4.4 Clases de Programación

4.4.1 Schedule Activo. Un Schedule factible es activo si al alterar la secuencia de procesamiento no se puede adelantar la finalización de una operación sin afectar el tiempo de inicio de la otra operación (Sarmiento Ardila C. J., 2012).

4.4.2 Schedule Semiactivo. En este tipo de programación ninguna operación se puede iniciar más temprano sin cambiar la secuencia operativa de cualquier máquina. La siguiente operación se asigna en el momento más temprano posible. Esta programación en general no es óptima ya que los Schedules óptimos se usualmente se encuentran en el espacio de la programación activa. Por esto, se recurre al procedimiento “permissible Left Shifting” (Cambios hacia la izquierda). Para modificar la programación semiactiva en una activa cambia la secuencia de dos (2) operaciones adyacentes, siempre y cuando no viole ninguna restricción de precedencia o cause una demora en alguna de las secuencias de la máquina. Cuando se hacen todos los cambios permitidos se obtiene una programación activa. (Sarmiento Ardila C. J., 2012)

4.4.3 Schedule sin retraso (Non-Delay). En esta programación, las máquinas no pueden mantenerse ociosas si hay alguna operación sin procesarse. (Sarmiento Ardila C. J., 2012)

4.5 Problemas de programación de operaciones

Dentro de las empresas manufactureras muchos de los procesos de fabricación pueden ser representados como problemas de programación o Scheduling Problems, el cual consiste en la asignación de recursos a tareas que se intentan ejecutar con el menor consumo de recurso posible

y reduciendo los tiempos de espera (Escamilla Fuster, Eficiencia Energética y Robustez en Problemas de Scheduling, 2016).

Hagar, Haddock y Heragu (1995) propusieron la siguiente clasificación del Scheduling problema, la cual ha sido una de las clasificaciones más populares, esta ubica a los problemas de programación según su naturaleza, la cantidad de máquinas involucradas, el método y el rendimiento (ver figura 2).

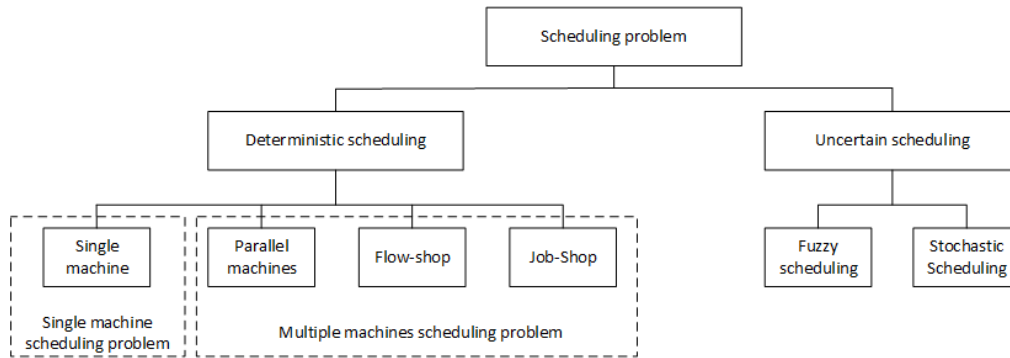


Figura 2. Clasificación del Scheduling Problem según Hagar, Haddock y Heragu, 1995.

4.5.1 Patrones de Flujo.

4.5.1.1 Open Shop Scheduling. Este problema consiste en un número de operaciones O_{ij} que deben ser procesadas en un conjunto de máquinas, en el cual no existen ninguna restricción de precedencia, ya que este problema solo se concentra en encontrar un orden de trabajo y un orden de operación a ser procesada en la misma máquina. (Delgado, 2012).

4.5.1.2 Flow-Shop Scheduling. Es una extensión del Job-Shop, pero con la singularidad de que los recursos deben de ser utilizados en el mismo orden para todos los trabajos. El número de tareas debe de ser el mismo para todos los trabajos. (Garey, D, & Sethi, 1976) en este tipo de problemas existen restricciones de precedencia, es decir cada trabajo debe ser procesado en la maquina m_i antes de pasar a la maquina m_{i+1} .

4.5.1.3 Job-Shop Scheduling. en un problema Job Shop los trabajos n están formados por secuencias totalmente ordenadas de operaciones, los cuales deben ser procesados una sola vez por m máquinas, con un tiempo determinado. (Delgado, 2012).

Para la solución de estos problemas de programación de operaciones se han utilizados los siguientes tipos de métodos

- **Grafo Disyuntivo.** Propuesto por Roy y Sussman (1994), es considerada una de las representaciones más claras y utilizadas en el JSP (Blazewicz, Pesch y Sterna, 2000), el cual se encuentra definido como:

$$G = (V, C, D)$$

Siendo V un conjunto de nodos, cada uno de estos representa una operación excepto 2, el nodo fuente y el nodo sumidero, también denominados nodos de inicio (V) y finales ficticios ($n \times m + 1$).

$$(V, V_{11}, V_{12}, \dots, n \times m + 1)$$

C y D , son consideradas aristas que unen los distintos nodos, donde C es una arista de precedencia (conjunciones) correspondientes a las restricciones de precedencia y D es una arista de recursos (Disyunciones) y representa las restricciones de recursos. Cada arista tiene asociada una longitud igual a la duración de la operación. (Barba Rodríguez, s.f).

La Figura 33, representa una planificación factible para el problema JSS con tres trabajos y tres recursos, en donde los arcos en negrita representan un camino crítico cuyo coste es 12, es decir, el Makespan de esta planificación será de 12.

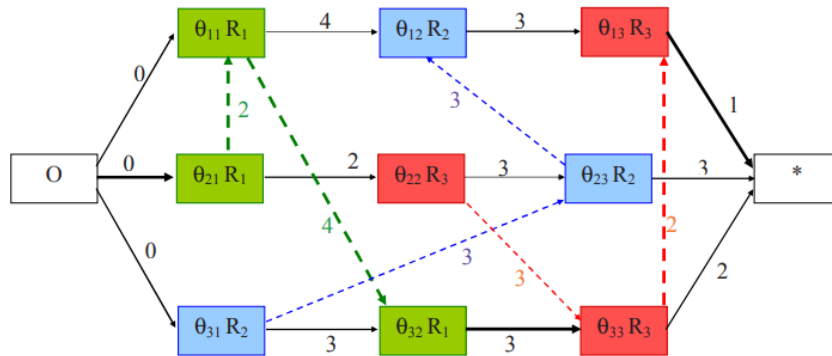


Figura 3. Representación en forma de grafo dirigido de un JSP. Adaptado de Mejora de algoritmos de búsqueda Heurística mediante poda por dominancia. Aplicación a problemas de Scheduling.

• **Diagrama de Gantt:** Es una forma de visualizar la representación de la solución del JSP. Donde se puede observar que en el eje vertical se encuentra la máquina y los trabajos a realizar en esta (dibujados como bloques) y en el eje Horizontal se muestra el tiempo de inicio y el procesamiento del trabajo en la máquina, como se muestra en la Figura 4.

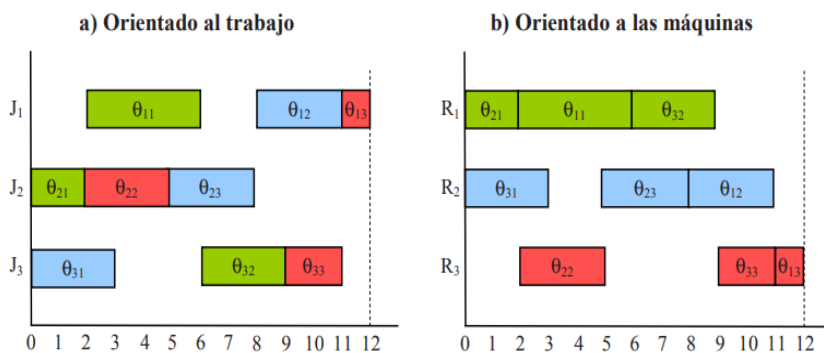


Figura 4. Representación del diagrama de Gantt en un problema JSP. Adaptado de Mejora de algoritmos de búsqueda Heurística mediante poda por dominancia. Aplicación a problemas de Scheduling.

Una extensión del problema Job Shop que se ha venido trabajando durante los últimos años debido a la importancia en la reducción de costos y la problemática ambiental, es el Job Shop con eficiencia energética, en el cual se estará trabajando durante el desarrollo de esta investigación.

4.5.2 Job Shop con Eficiencia energética. En el problema de programación Job Shop con eficiencia Energética (EJSP), las máquinas pueden trabajar a diferentes velocidades, con la variante de que cada operación puede ser ejecutada por una máquina y esta máquina puede funcionar a diferentes velocidades $v = \{v_1, v_2, v_3, \dots, v_d\}$ (Jiang, Zhang, Zhu, Gu, y Deng, 2018).

Formalmente, el EJSP 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 la programación de operaciones factible que minimice el tiempo de finalización de todas las tareas Makespan y la energía utilizada (Salido et al. 2013)

Para la solución de estos problemas de programación de operación se han empleado diferentes tipos de métodos, exactos y aproximados, a continuación, se encuentra la definición del método empleado en esta investigación.

4.6 Métodos Exactos. Los métodos exactos son aquellos con la gran capacidad de encontrar una solución matemática óptima o al menos una solución de precisión conocida. Este método es el ideal en muchos aspectos para problemas de ubicación. Sin embargo, puede requerir una memoria amplia, largos tiempos de ejecución y una definición del problema comprometida para cuando este sea aplicado a problemas prácticos.

4.7 Métodos Aproximados. Según el análisis preliminar de la literatura, se encontró que entre las metodologías utilizadas para resolver el Problema Job Shop con Eficiencia Energética (EJSP) se encuentran los métodos aproximados los cuales se clasifican como: métodos heurísticos y métodos metaheurísticos.

4.7.1 Métodos Heurísticos. “Son un conjunto definido de pasos para identificar rápidamente una solución de alta calidad para un problema dado, donde dicha solución es un conjunto de valores para el problema” (Barr, RS, Golden, BL, Kelly, JP et al. J Heuristics. (1995).

Según Blum y Roli (2003) las heurísticas se pueden clasificar en: métodos constructivos y de búsqueda local; los métodos constructivos son procedimientos iterativos capaces de construir una solución desde cero mediante la adición de componentes de una solución parcial inicialmente vacía hasta que la solución esté completa, un ejemplo de ellos es la regla de secuenciación. Y los métodos de búsqueda local: parten de alguna solución inicial e intentan sustituir iterativamente la solución actual por una solución mejor en un vecindario adecuadamente definido de la solución actual. Las reglas de secuenciación utilizan diferentes tipos de información (Lozano & Torres, 2018).

4.7.2 Métodos Metaheurísticos. El término metaheurístico, fue introducido por primera vez por Fred Glover en el año 1986, cuya expresión se deriva de la composición de dos palabras griegas, la heurística del verbo heuriskein (ἑύρισκω) que significa "encontrar", mientras que el sufijo meta significa "más allá, en un nivel superior". La definición del método metaheurístico según Osman y Laporte (1996) es que "Un metaheurístico se define formalmente como un proceso de generación iterativo que guía una heurística subordinada mediante la combinación de conceptos inteligentemente diferentes para explorar y explotar el espacio de búsqueda, las estrategias de aprendizaje se utilizan para estructurar la información con el fin de encontrar soluciones eficientes casi óptimas".

Partiendo del punto de que los métodos metaheurísticos son estrategias que "guían" el proceso de búsqueda, cuyo objetivo es explorar eficientemente el espacio de búsqueda para encontrar soluciones (casi) óptimas.

A continuación, se describen los métodos más comunes usados en la literatura para dar solución al EJSP.

- **Búsqueda Tabú:** (Tabu Search, TS), es un método metaheurístico cuyas ideas básicas se introdujeron por primera vez gracias a la definición de Fred Glover en los años de los 80, término que ha sido usado con frecuencia para la solución del JSP. Este método utiliza explícitamente el historial de la búsqueda, tanto para escapar de los mínimos locales como para implementar una estrategia exploratoria, de igual manera, la TS consiste en un proceso de búsqueda de soluciones que son modificadas en cada nueva posición y almacenadas en memorias a corto plazo para escapar de los mínimos locales y evitar ciclos con el fin de guiar inteligentemente el proceso de búsqueda, el termino tabú se refiere a que el algoritmo convierte a las regiones ya exploradas en prohibidas para evitar examinarlas nuevamente.

- **Algoritmos genéticos.** 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 (Goldberg, 1989). A lo largo de las generaciones, este algoritmo se ha aplicado en diferentes problemas de optimización comenzando un conjunto inicial de soluciones llamadas poblaciones, cada solución

se considera cromosoma como punto de búsqueda en el espacio el cual está compuesto por genes (Zandieh, Mahdavi, y Bagheri, 2008).

- **Optimización de colonias de hormigas:** La técnica computacional de optimización llamado Ant Colony Optimization (ACO) es un modelo inspirado en la capacidad que poseen las hormigas para establecer el camino más corto desde su nido o colonia hasta la fuente de alimento y viceversa, presentado en este trabajo. Al moverse, una hormiga deposita una sustancia denominada feromona, marcando el camino que fue recorrido. Las siguientes hormigas detectan la feromona ya depositada y deciden con alta probabilidad seguirla. El camino más corto recibirá mayor cantidad de feromonas por unidad de tiempo, y pronto todas las hormigas estarán recorriendo dicho trayecto.

Según Khalouli y Ghedjati, para moderar el comportamiento de las hormigas, se deben definir los siguientes componentes: representación adecuada de feromona, mecanismo de actualización de la cantidad de feromona y función que pueda brindar información sobre el problema específico.

- **Optimización por enjambre de Partículas (PSO).** Es un método para funciones no lineales en espacios continuos y discretos desarrollado por Kennedy y Eberhart (1995) basado en la analogía del comportamiento social de animales tales como el vuelo de aves, el movimiento de los peces y la teoría de enjambre es considerado como una de las técnicas de convergencia más rápidas y un algoritmo muy ajustable a funciones multimodales. PSO tiene algunas características atractivas, al tener memoria, el conocimiento de las buenas soluciones es retenido para todas las partículas adicionalmente como tiene cooperación constructiva entre partículas, y las partículas en el enjambre comparten información entre ellos. Debido a su simple concepto, fácil implementación y rápida convergencia, PSO ha sido de centro de atención y gran aplicación en diferentes campos, principalmente para problemas de optimización continua. (Lui, Wang , & Jin, 2007)

- **Algoritmos híbridos:** Los Algoritmos híbridos (AH) son creados a través de la combinación de dos o más metaheurísticas eso con el fin de obtener mejores resultados de forma más rápida y eficiente, pues, de esta forma se mezclan diferentes conceptos en los que se aprovecha los puntos fuertes de cada una de estos.

5 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 (1983) durante un trabajo de investigación sobre problemas de diseño de circuitos electrónicos y el cual se basa en la analogía que puede existir entre un proceso de optimización combinatoria y un proceso termodinámico conocido como recocido (ver figura 5). Este proceso se lleva a cabo a través de 2 etapas, la primera consiste en la elevación de temperatura de un sólido cristalino con defectos, temperatura que por lo general suele alta y la segunda, es la etapa de enfriamiento en la que, el material se enfría lentamente a través de un baño térmico. En esta última fase se espera que el sólido alcance a conformar una red cristalina más regular y así reducir sus defectos.

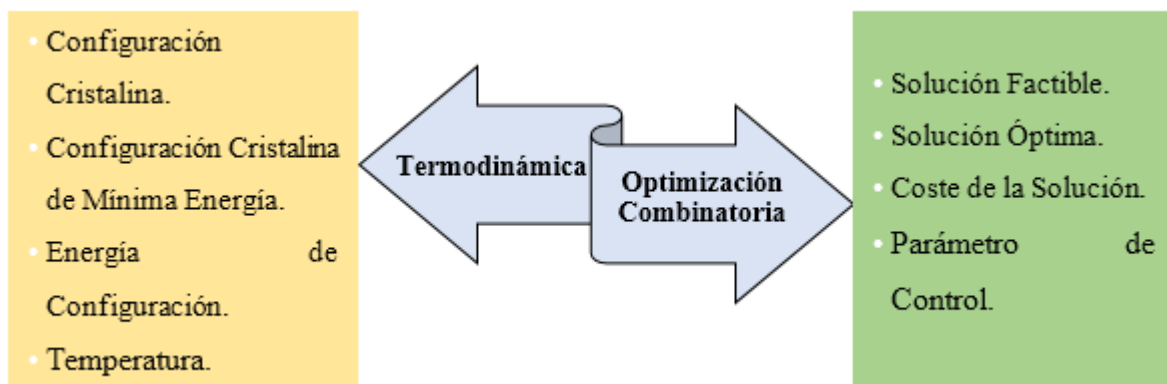


Figura 5. Analogía de los parámetros termodinámicos y de optimización. Adaptado de Duarte Muñoz, A., Pantrigo Fernández, J., y Gallego Carrillo, M. (2007). *Metaheurísticas*. Editorial DYKINSON, S.L. Madrid. Pág. 72.

Desde el punto de vista algorítmico, el SA es un procedimiento de búsqueda local que introduce una fase de aleatorización en la aceptación de movimientos, de tal forma que si el movimiento mejora se acepta. Esta aleatorización se encuentra regulada por el parámetro temperatura (T) que corresponde al proceso termodinámico. (Duarte Muñoz, Pantrigo Fernández y Gallego Carrillo, *Metaheurísticas*, 2007), en otras palabras el algoritmo empieza con una solución x que puede ser entregada inicialmente o generada aleatoriamente, posteriormente procede al cálculo de la función objetivo $f(x)$ y continua evolucionando a la vez que va realizando pequeños desplazamientos que son aceptados si el valor de la función objetivo es inferior a la solución anterior, si por el contrario, la solución es aceptada con una probabilidad que decrece exponencialmente con el cociente de la

diferencia entre los valores de la función objetivo (δ) y el parámetro T. Ecuación de probabilidad de aceptación $e^{-\frac{\delta}{T}}$.

El hecho de que la probabilidad de aceptar soluciones peores no sea nula permite que el algoritmo no queda atrapado en un mínimo local.

5.2 Elementos del SA

En 1994 Hira N. Ahuja en su libro titulado *Project Management Techniques in Planning and Controlling Construction Project*, presentó los siguientes elementos como los principales del SA.

- Solución de partida o solución inicial.
- La solución se deriva de las soluciones viables originales al buscar con el modo de movimiento de configuración y se ajusta a las restricciones del problema.
- La probabilidad de aceptación es la probabilidad de que la solución de vecindario pueda moverse.
- El conjunto de configuraciones, o estados, del sistema.
- Una regla de generación para nuevas configuraciones, que generalmente se obtiene definiendo la vecindad de cada configuración y eligiendo la siguiente configuración al azar de la vecindad de la actual.
- El objetivo significa función objetivo.
- El programa de enfriamiento del parámetro de control incluye valores iniciales y reglas para cuándo y cómo cambiarlo.
- La condición de terminación generalmente se basa en el tiempo y los valores de la función de costo y / o el parámetro de control.

5.3 Parámetros de entrada del Algoritmo

Antes de iniciar con la elaboración del algoritmo es necesario contar con los siguientes parámetros de entrada:

5.3.1 Temperatura inicial T_0 . Para calcular T_0 no existe un método definido, por lo que es común que, se adopten formulas o métodos de acuerdo con las características del problema, aunque también existe la posibilidad de que el valor de esta sea definido de forma experimental. En la literatura se recomienda que este parámetro sea lo suficientemente caliente o alto para permitir movimientos hacía cualquier lugar del vecindario, aunque también es necesario que tener en cuenta que si este valor es demasiado alto podría moverse hacia cualquier lugar del vecindario, convirtiéndose esta en una búsqueda aleatoria.

En la literatura algunos autores han planteado diferentes estrategias o métodos para determinar la T_0 , por ejemplo, Kurt y Serap (2007) en su trabajo titulado “A Simulated Annealing Approach to the Solution of Job Rotation Scheduling Problems” plantearon la siguiente ecuación e indicaron que el valor de esta temperatura debe permitir que el valor de la probabilidad de aceptación (PC) sea 0.99.

$$T_{in} = \frac{F_{min} - F_{max}}{\ln P_c}$$

Así mismo Rayward-Smith, Osman, Reeves y Smith (1996) propusieron comenzar con una alta temperatura y enfriarla rápidamente hasta que aproximadamente el 60% de los cambios aceptados han sido las soluciones más deficientes, posteriormente se continua con el enfriamiento de forma lenta convirtiéndose este en la temperatura de inicio. Por su parte Babak, Shahram y Jamal en su trabajo titulado Bi-objective Resource-Constrained Project Scheduling with Robustness and Makespan Criteria (2006) señalan que la T_0 debe ser lo suficientemente alto como para que la primera iteración del algoritmo permita que la probabilidad de aceptar una solución peor se por lo menos del 80%.

5.3.2 Temperatura final T_f . En la teoría T_f debería ser 0, pero esto resulta inviable, ya que los decrecimientos geométricos impedirían que se llegase a un $T_f = 0$, además en la práctica esto conduciría a tiempo computacionales mayores, así que se puede recurrir a opciones como: detener el algoritmo cuando cuando $T_f = 0.1T_o$.

5.3.3 Estructura de generación de Vecindario. Es una función que define una transición simple a partir de una solución x a otra solución mediante la inducción de un cambio que generalmente puede ser visto como una pequeña perturbación (Glover y Laguna , 1997).

5.3.4 Velocidad de enfriamiento L. se encuentra definida por la cantidad de iteraciones que se realizan con el mismo valor de T.

5.3.5 Tasa de disminución de Temperatura α . También conocida como enfriamiento debe estar entre 0-1, aunque en la literatura, lo convencional es usar valores entre 0.8-0.9.

5.3.6 Función de variación de Temperatura. En la literatura se encuentran con mayor frecuencia las siguientes leyes (Ver tabla 1), las cuales son aplicadas en la formulación del SA. (Lozano Suarez y Torres Cárdenas, 2018).

Tabla 2.

Funciones de Variación de Temperatura.

Modelo	Formula	Parámetro
Lineal	$T_{k+1} = T_k + C$	C
Exponencial	$T_k = T_o + \alpha^k$ $T_{k+1} = T_k * \alpha$	α
Logarítmico	$T_{k+1} = \frac{C}{\text{Log}(K + 1)}$	C

6 Problema Job Shop con Eficiencia Energética- EJSP

El Job Shop Scheduling con la extensión de eficiencia energética 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.

6.1 Descripción del problema

El modelo matemático seleccionado para la presente investigación es el propuesto por Tianhua Jiang, Chao Zhang y Qi-Ming Sun (2019). Este modelo fue nombrado por sus autores como “The green job shop scheduling problem with multi-speed machine” y se encuentra definido de la siguiente forma:

- 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 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.
- El problema considerado consiste en 2 subproblemas: la combinación de las operaciones 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 descompostura de la máquina 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 .

6.2 Formulación

El siguiente modelo es el propuesto por Jiang , Zhang y Qi-ming (2019), se define como 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.

6.2.1 Conjuntos del modelo.

i : Trabajos.

k : Máquinas.

o : Operaciones.

6.2.2 Parámetros del modelo

E_{kD} : Costo de consumo de energía por unidad de tiempo (aplica solo para el consumo de energía cuando la maquina cambia de velocidad).

SE_k : Costo de consumo de energía por unidad de tiempo (aplica solo para el consumo de energía cuando la maquina está en modo de espera (Stand-by)).

6.2.3 Variables del modelo

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

C_{ik} : Tiempo de finalización de O'_{ik}

S_{ik} : Hora de inicio de O'_{ik}

C_k : Tiempo de finalización de la máquina k .

W_k : Carga de trabajo total de la máquina k .

C_{max} : Makespan

η : Costo relacionado al tiempo de finalización por unidad de tiempo.

P_{ikd} : Tiempo de procesamiento del trabajo i en la máquina k con una velocidad d .

D_{ik} : Número de niveles ajustables de la máquina k .

Q : Constante positiva muy grande.

P_{ik} : Operación predecesora de O'_{ik} en el trabajo i .

CP_{ik} : Tiempo de finalización P_{ik} .

x_{ikd} : Si el trabajo es procesado en la máquina k con velocidad d , $x_{ikd} = 1$; de lo contrario $x_{ikd} = 0$.

z_{ilk} : Si el trabajo es procesado en la máquina k antes del trabajo l , $z_{ilk} = 1$; de lo contrario $z_{ilk} = 0$.

6.2.4 Función objetivo.

$$\min F = \sum_{i=1}^n \sum_{k=1}^m \sum_{d=1}^{D_k} E_{kd} P_{ikd} x_{ikd} + \sum_{k=1}^m SE_k (C_k - W_k) + \eta C_{max} \quad (1)$$

6.2.5 Restricciones.

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

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

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

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

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

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

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

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

La expresión (1) representa la función objetivo y las siguientes corresponden a las restricciones bajo las cuales se deberá regir el modelo. (2) esta restricción asegura que la velocidad de una maquina no pueda ser cambiada durante el procesamiento de una operación. (3) restricción de precedencia entre operaciones de un trabajo. (4) restricción que garantiza que cada máquina pueda procesar una operación a la vez. (5) Restricción de carga de trabajo de cada máquina. (6) esta restricción proporciona el tiempo de finalización de cada máquina. (7) restricción de no negatividad. (8) y (9) restricciones lógicas o binarias.

Para el desarrollo del presente proyecto se tomó la decisión de ajustar la función objetivo, intercambiando las variables de costo por variables de consumo de energía y, también se optó por eliminar los tiempos de ocio, esto se debe a que nuestra investigación no se encuentra en función de costos, quedando de la siguiente forma:

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

Donde,

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

6.2.6 Ejemplo. Para explicar este problema, se presenta a continuación un ejemplo con una de las instancias descritas por Escamilla J., Salido, Giret y Barber (2016), esta cuenta con 3 trabajos, 3 máquinas y 5 operaciones por cada uno de estos, la tabla 3 representa la cantidad de operaciones que componen un trabajo.

Tabla 3
Número de operaciones.

Trabajo	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
Operaciones	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

La tabla 4 corresponde a la matriz de máquinas disponibles por operación, es decir cada número representa la máquina que trabajará en cada operación (teniendo en cuenta la restricción que garantiza que cada máquina pueda procesar una operación a la vez).

Tabla 4
Matriz de máquinas disponibles.

Trabajo	Secuencia de máquina				
1	3	2	1	3	1
2	1	2	3	2	3
3	1	3	2	1	3

Seguidamente se encuentra la matriz de velocidades (ver tabla 5) donde se muestra la forma en la que están ordenadas las 3 velocidades, teniendo en cuenta que cualquiera de estas puede ser utilizadas en las operaciones.

Tabla 5
Matriz de velocidades

Velocidades														
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

La matriz de tiempo de procesamiento (P_{ikd} /tiempo_de_procesamiento- expresan el tiempo de procesamiento en el modelo matemático y en Matlab.) (ver tabla 6) corresponde a los tiempos de procesamiento posibles teniendo en cuenta la máquina y velocidad con que se desarrolla dicha operación.

Tabla 6
Matriz de procesamiento

	Trabajo 1					Trabajo 2					Trabajo 3				
	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇	O ₈	O ₉	O ₁₀	O ₁₁	O ₁₂	O ₁₃	O ₁₄	O ₁₅
Velocidad 1	14	6	15	5	4	14	9	4	3	3	9	10	4	10	6
Velocidad 2	10	5	11	4	3	12	7	3	2	2	8	7	3	9	4
Velocidad 3	7	4	10	3	2	8	6	2	1	1	6	5	2	7	3

La matriz de energía usada (A_{kd}/EO) está relacionada con el gasto de energía teniendo en cuenta la velocidad con que se desarrolló la operación.

Tabla 7
Matriz de energía usada

	Trabajo 1					Trabajo 2					Trabajo 3				
	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇	O ₈	O ₉	O ₁₀	O ₁₁	O ₁₂	O ₁₃	O ₁₄	O ₁₅
Velocidad 1	14	4	13	3	1	16	5	2	1	1	8	4	1	3	3
Velocidad 2	10	5	14	4	2	17	10	3	1	1	12	7	1	5	5
Velocidad 3	19	6	15	5	3	18	12	4	2	2	17	13	2	9	6

6.3 Desarrollo del algoritmo

6.3.4 Solución inicial. El Recocido Simulado parte de una solución inicial del EJSP la cual funciona en el algoritmo como punto de comparación para las nuevas soluciones vecinas halladas.

La solución inicial del presente problema fue obtenida de manera aleatoria, donde un vector es generado con sus respectivas operaciones y velocidades (ver Apéndice B). En la tabla 6 se muestra una solución aleatoria obtenida para el ejemplo en mención, en ella se indican las posiciones de cada operación junto con las velocidades en las que se trabajarán.

Tabla 8
Solución inicial para el ejemplo 3,3,5

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

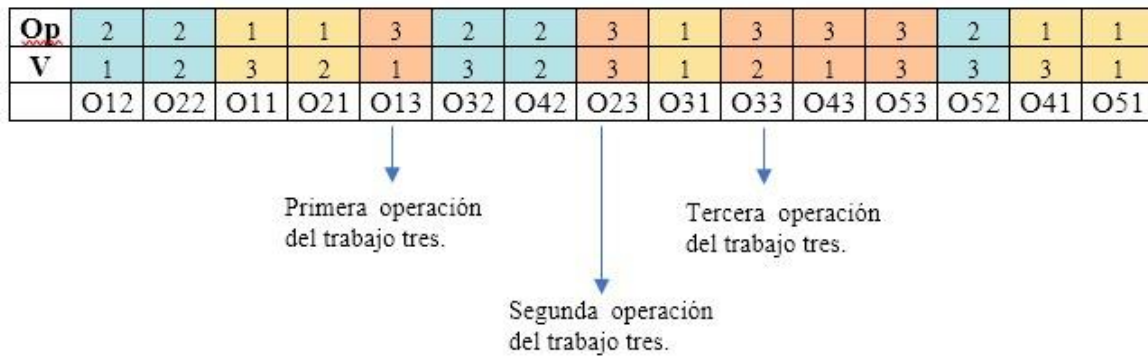


Figura 6. Explicación de la estructura del vector solución inicial.

6.3.5 Makespan. En el proceso del cálculo del Makespan, inicialmente se creó un código el cuál funcionaba para instancias con 3 máquinas; continuar con la ampliación de la programación requería de muchas más horas de trabajo, por lo que se optó por adaptar el algoritmo propuesto por Karim y Yuleini (2013), en su trabajo titulado “*un Algoritmo Memético para la minimización del Makespan en el problema del Job Shop Scheduling*”, para el cálculo del makespan ya que se ajustaba a las necesidades requeridas en la presente investigación (ver Apéndice C).

El algoritmo debía tener en cuenta que ningún trabajo era prioritario, pero a su vez debía cumplir con una secuencia, es decir, el trabajo 1 no es más importante que el trabajo 2, pero si es necesario que al momento de acomodar las operaciones de cada trabajo estas sean consecutivas; no podía realizar la operación 3 del trabajo 1 (O₃₁) sin haber realizado la operación 2 del trabajo 1 (O₂₁); seguido a esto era necesario que adelantará operaciones en los tiempos de espera siempre y cuando esto no interrumpiera las operaciones que iban antes o después, por ejemplo, si la operación 5 del trabajo 2 (O₅₂) está programada después de la operación 5 del trabajo 3 (O₅₃), pero entre la operación 2 del trabajo 3 (O₂₃) y la operación 5 del trabajo 3 (O₅₃) hay un tiempo de espera mayor o igual que el tiempo de procesamiento de la O₅₂, esta ocuparía ese espacio y ayudaría a disminuir el Makespan pasando de 56 a 47, como se aprecia en las Figuras 7 y 8.

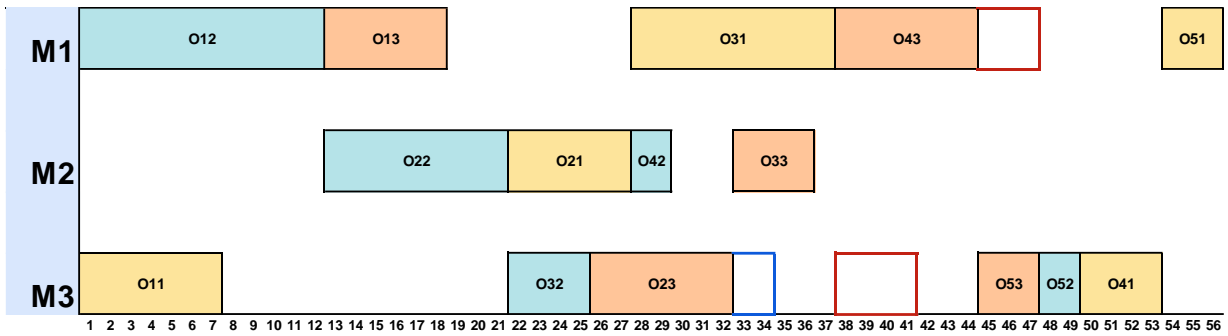


Figura 7. Diagrama de Gantt sin adelantar operaciones.

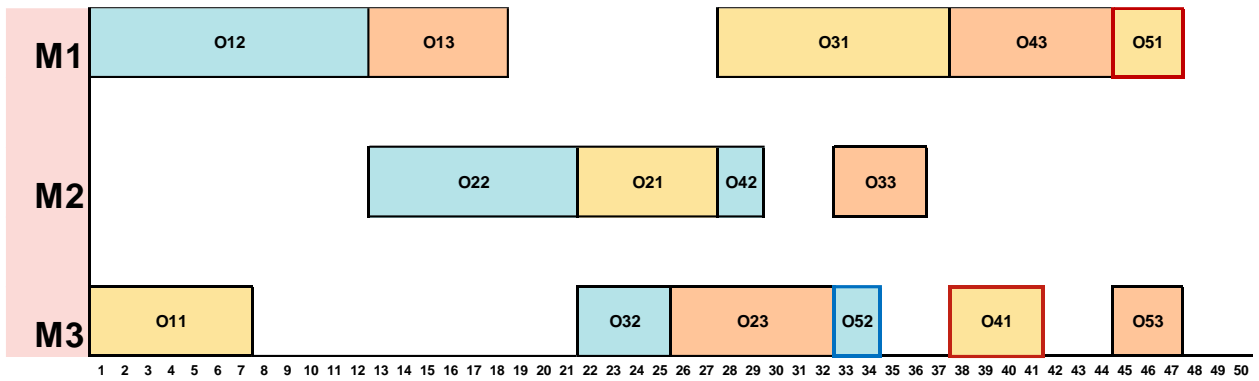


Figura 8. Diagrama de Gantt con adelanto operaciones.

Los tiempos de espera entre operaciones que no puedan ser adelantados son los llamados tiempos ociosos y estos también se tuvieron en cuenta para realizar el cálculo del Makespan.

6.3.6 Consumo de Energía. El cálculo del consumo de energía se determinó teniendo en cuenta el vector solución inicial creado inicialmente que está compuesto por dos filas, una de operaciones y otra de velocidades, seguido a esto se creó un vector que corresponde a las asignaciones de las energías de cada operación con su respectiva velocidad de procesamiento y, por último, se realizó la suma de este la cual correspondiente a la energía total consumida durante el proceso (ver Apéndice D y E).

6.3.7 Parámetros de entrada del Recocido Simulado.

6.3.7.1 *Estructura del Vecindario*. El vecindario fue desarrollado a partir de la generación de 2 números aleatorios, estos representaban las posiciones de 2 operaciones las cuales serían intercambiadas de lugar junto con las velocidades (recuadros rojos) en la que deben ser procesadas, como se muestra en la Figura 9 (ver apendicce F) (Osorio, Lasso, y Ruiz, 2012).

Op	2	2	1	1	3	2	2	3	1	3	3	3	2	1	1
V	1	2	3	2	1	3	2	3	1	2	1	3	3	3	1
	O12	O22	O23	O24	O25	O26	O27	O28	O29	O30	O31	O32	O33	O34	O35

Op	2	1	1	1	3	2	2	3	1	3	3	3	2	2	1
V	1	3	3	2	1	3	2	3	1	2	1	3	3	2	1
	O12	O22	O23	O24	O25	O26	O27	O28	O29	O30	O31	O32	O33	O34	O35

Figura 9 Ejemplo de la estructura de vecindario

6.3.7.2 *Temperatura Inicial*. Durante la revisión de literatura se encontró información en la que se indicaba que para obtener buenos resultados con el Recocido Simulado era necesario empezar con temperaturas altas, investigadores sugieren que este valor debe permitir que la probabilidad de aceptar una solución sea por lo menos del 80%, por su parte otros concluyen que este valor debe establecerse para que haya una probabilidad de 95% de aceptar una solución peor en un 30% de la solución inicial.

En la presente investigación por temas de capacidad en los equipos usados fue necesario hacer algunas iteraciones de prueba para determinar que temperatura inicial a usar, esto debido a que, con la información recolectada de la literatura se obtenían tiempos de procesamientos largos y se sobrecargaban los equipos de trabajo.

Continuando con el ejemplo 3,3,5 en el siguiente capítulo se puede observar el diseño de experimento realizado del cual se extrajeron el valor de la temperatura inicial.

6.3.7.3 *Temperatura Final*. Este criterio de parada del algoritmo fue definido en 1 para las instancias pequeñas, para las grandes se tuvo en cuenta el tiempo de procesamiento el cual no debía superar más de una hora.

6.3.7.4 *Decremento de Temperatura α* . En la literatura varios autores sugieren que este parámetro debería estar entre 0.7 y 0.95, aunque aclaran que en el límite mayor es donde se encuentran los mejores resultados, pero entre más grande sea el decremento mayor será el tiempo de procesamiento, ya que la temperatura disminuirá más lentamente antes de alcanzar el criterio de parada. Para el problema trabajado se utilizó un modelo exponencial con un valor de α 0.8, esto con el objetivo de disminuir los tiempos de procesamiento y evitar sobrecargar los equipos con los que se estaba trabajando.

$$T_{K+1} = T_K * \alpha$$

El valor de α fue seleccionado después de realizar un diseño de experimento 2^k .

6.3.7.5 *Número de vecinos*. Para seleccionar el número de vecinos que iban a ser evaluados durante cada disminución de temperatura se decidió realizar un diseño experimentos, los límites explorados para el ejemplo 3,3,5 mencionado anteriormente fueron de 10 y 30, nivel bajo y alto respectivamente.

6.4 Función Fitness

Cuando se trabaja con problemas de optimización simples o Mono-objetivos las soluciones óptimas encontradas se limitan a una o varias equivalentes. En la presente investigación, el problema Job Shop con Eficiencia Energética maneja 2 objetivos, por lo que se convierte en un problema de optimización multiobjetivo en el que se busca minimizar tanto el Makespan como el Consumo de Energía, para este tipo de problemas no existe una solución óptima sino un conjunto de soluciones.

Varias técnicas han sido desarrolladas para resolver los problemas de optimización multiobjetivo, pero el método más conocido debido a su fácil uso y capacidad para identificar soluciones eficientes es la función de Normalización denominada como “Normalized Weighted

Additive Utility” (NWAUF)”); en esta función los múltiples objetivos se normalizan y se unen para crear una función de utilidad, obteniendo así una ecuación homogénea adimensional que permite resolver el problema multiobjetivo como uno simple (Escamilla Fuster, Eficiencia Energética y Robustez en Problemas de Scheduling, 2016).

Siendo f_i el i -ésimo valor de la función objetivo, la función Fitness se encuentra definida como:

$$U_j = w_1 f'_1 + w_2 f'_2 + \dots + w_k f'_k$$

Donde w_1, w_2, \dots, w_k son los pesos de cada objetivo y f'_1, f'_2, \dots, f'_k son valores normalizados f_1, f_2, \dots, f_k . Los pesos muestran la importancia de cada objetivo, los pesos muestran la importancia de cada objetivo donde:

$$\sum_{i=1}^k w_i = 1$$

$$0 \leq w_i \leq 1 \text{ para } i = 1, \dots, k$$

El objetivo del EJSP es lograr minimizar el Makespan y el Consumo de Energía, para esto, la función Fitness quedaría de la siguiente manera (ver Apéndice G):

$$F_i = \lambda * NormMakespan_i + (1 - \lambda) * NormEnergia_i$$

$$NormMakespan_i = \frac{Makespan_i}{MaxMakespan}$$

$$NormEnergia_i = \frac{EnergiaTotal_i}{MaxEnergia}$$

Los valores máximos del Makespan y Consumo de Energía se encuentran definidos dentro de las instancias usadas. Los pesos asignados a las variables están dados por λ . El máximo makespan corresponde al tiempo máximo que pueden demorarse las operaciones en terminar de ejecutarse, este valor ya se encontraba calculado en las instancias elegidas a comparar de la literatura, por otro lado, MaxEnergía corresponde a la suma de los consumos de energías de cada operación a una velocidad máxima, para nuestra investigación es 3.

6.5 Diagrama de flujo del Recocido Simulado (ver Apéndice H)

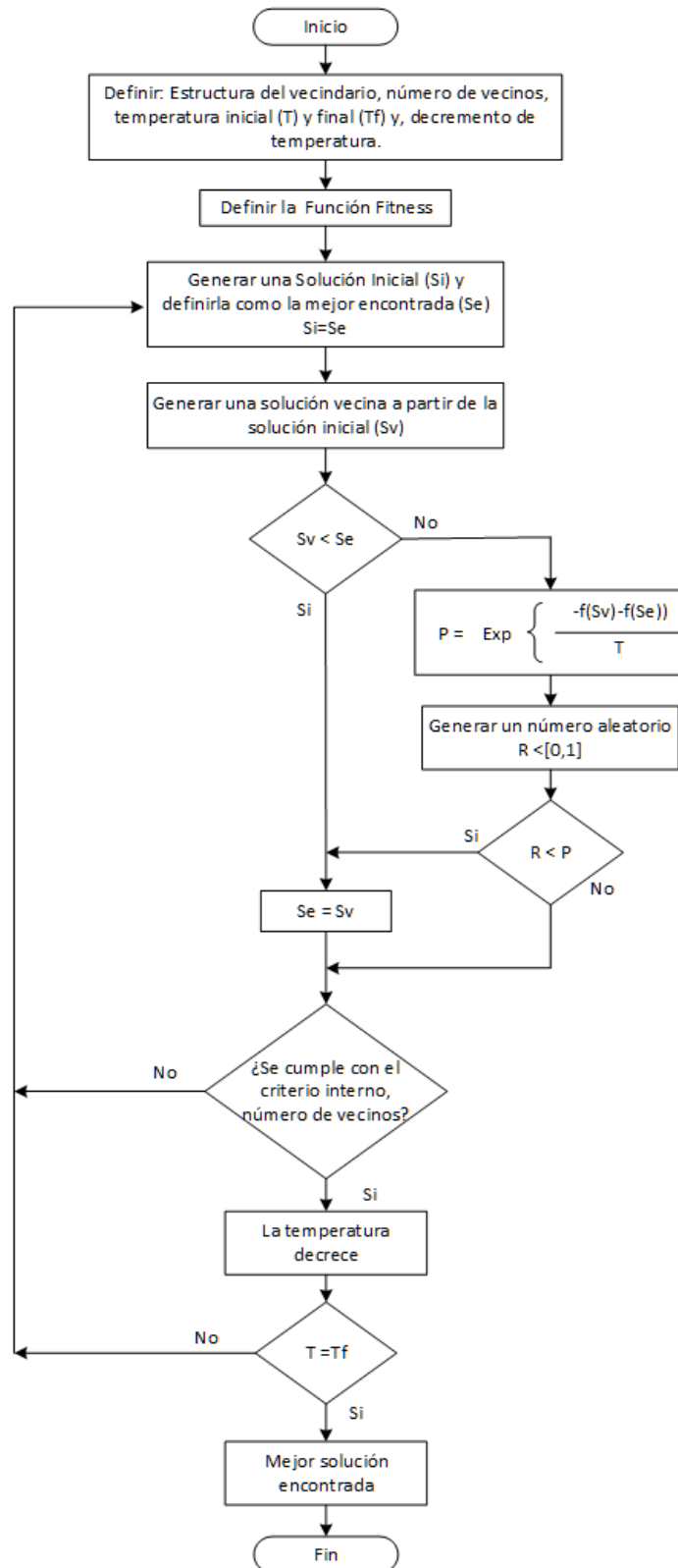


Figura 10. Diagrama de flujo del Recocido Simulado para las pequeñas y medianas instancias.

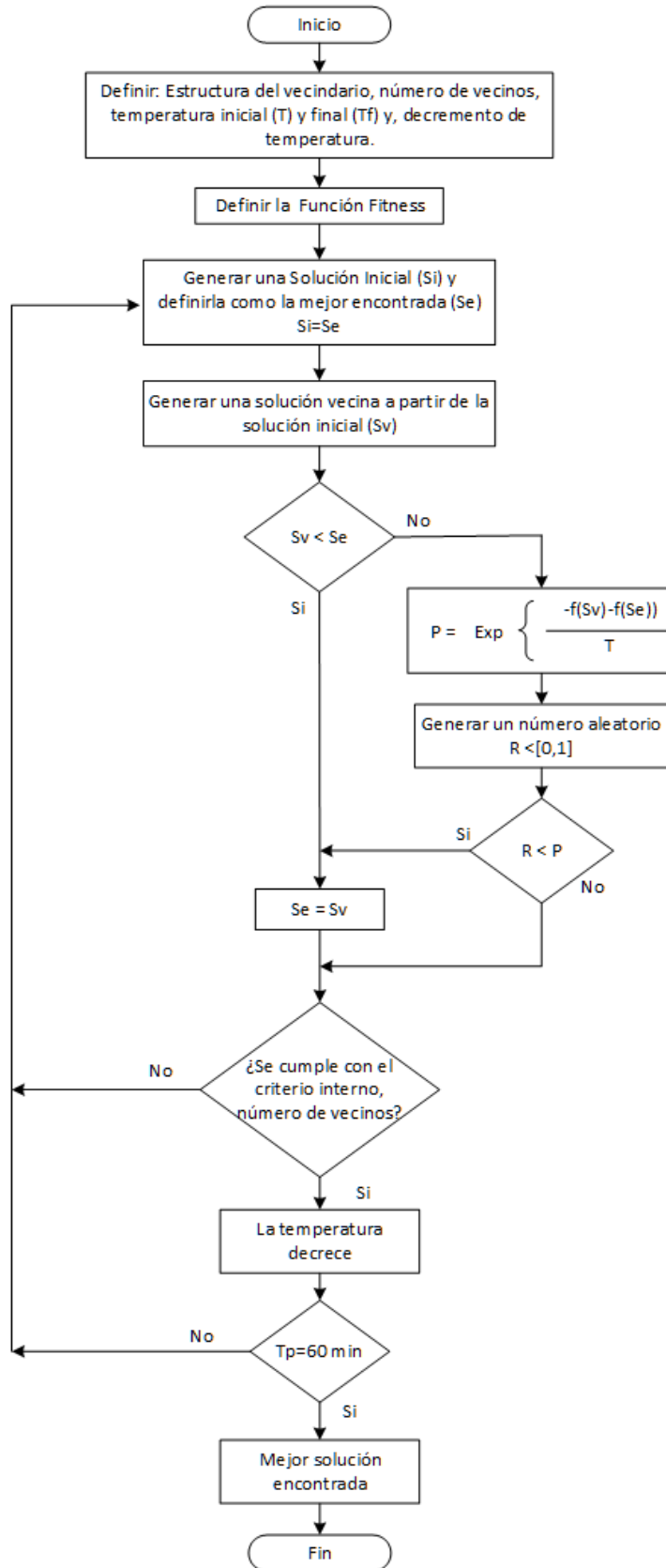


Figura 11. Diagrama de flujo del Recocido Simulado para las grandes instancias.

7. Validación del algoritmo Propuesto

Para analizar el comportamiento de Recocido Simulado en el EJSP se realizó un diseño de experimentos 2^k (ver Apéndices I, J y K), este diseño factorial se ejecutó en el software Minitab. Las instancias usadas fueron clasificadas en tres grupos pequeñas, medianas y grandes (ver tabla 8).

Tabla 9
Instancias usadas

Tamaño	txmxo	Nombre
Pequeña	3 x 3 x 5	Ejemplo 3,3,5
	3 x 7 x 10	Ejemplo 3,7,10
Mediana	50 x 20 x 20	Ejemplo 50,20,20
Grande	100 x 20 x 20	Ejemplo 100 x 20 x 20

Nota: t: trabajo; m: máquina y o: operación.

Las instancias pequeñas fueron evaluadas con un diseño de experimento 2^3 con 3 réplicas, los factores y niveles se muestran a continuación,

Tabla 10
Factores y niveles del diseño experimental para las pequeñas instancias

Factores	Niveles	
	Bajo	Alto
Temperatura	80	200
Vecindario	10	30
Decremento T	0,6	0,8

Los niveles fueron seleccionados a partir de diferentes iteraciones que se realizaron con el algoritmo diseñado en Matlab, ya que al momento de utilizar los encontrados en la literatura junto los equipos personales con los que contaba el grupo de trabajo se obtuvieron tiempos de procesamientos largos (de más de 4 horas).

7.1 Análisis de Varianza

La Tabla 11 corresponde al análisis de varianza arrojado por los resultados del diseño de experimento correspondientes a la instancia Ejemplo 3,3,5.

Tabla 11
Análisis de Varianza para la instancia Ejemplo 3,3,5

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	7	0,010443	0,001492	1,35	0,290
Lineal	3	0,006451	0,002150	1,95	0,162
Vecinos	1	0,000079	0,000079	0,07	0,792
Temperatura	1	0,000183	0,000183	0,17	0,690
Decremento T	1	0,006189	0,006189	5,61	0,031
Interacciones de 2 términos	3	0,003199	0,001066	0,97	0,433
Vecinos*Temperatura	1	0,000129	0,000129	0,12	0,737
Vecinos*Decremento T	1	0,000064	0,000064	0,06	0,813
Temperatura*Decremento T	1	0,003006	0,003006	2,72	0,118
Interacciones de 3 términos	1	0,000794	0,000794	0,72	0,409
Vecinos*Temperatura*Decremento T	1	0,000794	0,000794	0,72	0,409
Error	16	0,017652	0,001103		
Total	23	0,028095			

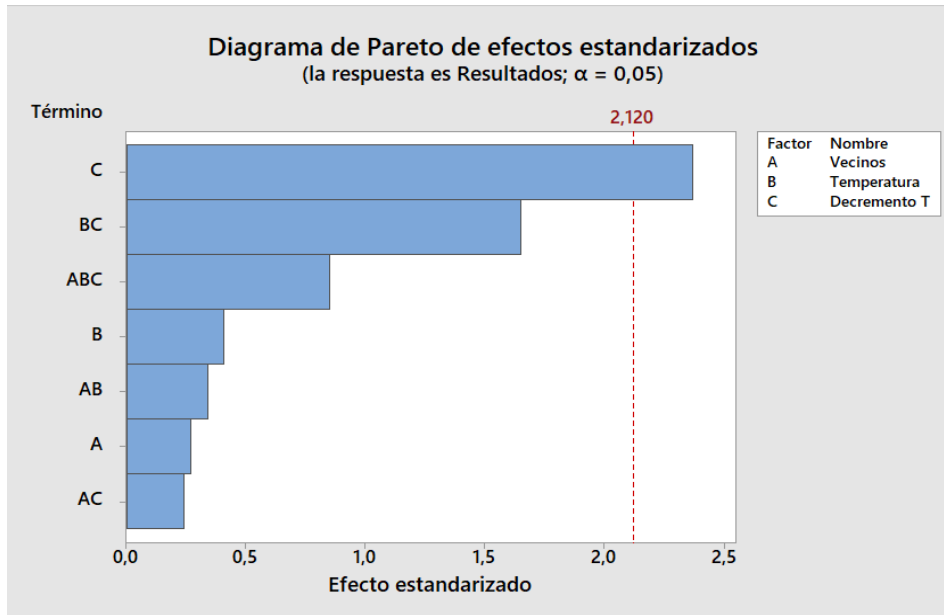


Figura 12. Diagrama de Pareto de efecto estandarizados para la instancia Ejemplo 3,3,5

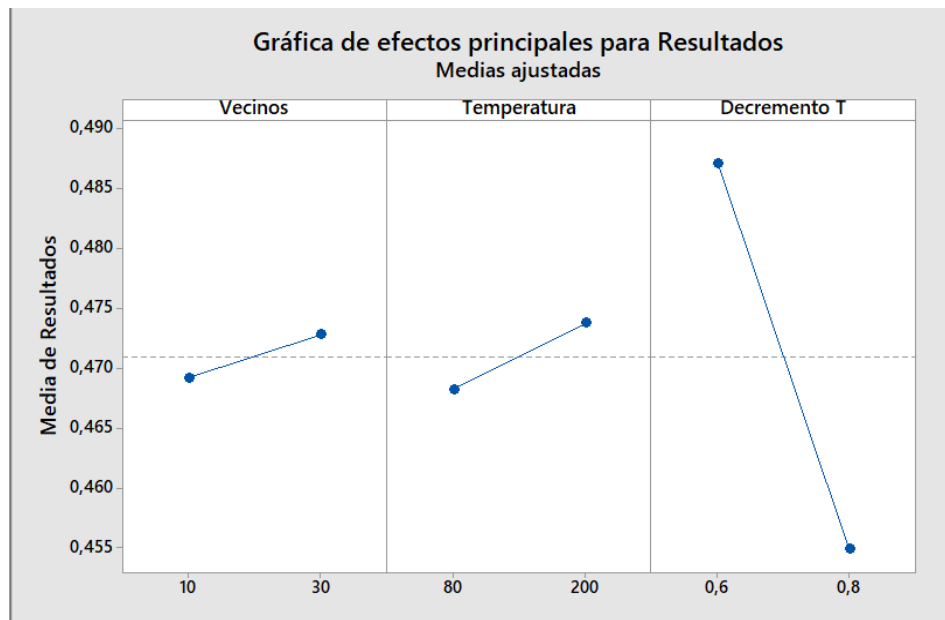


Figura 13. Gráfica de efectos principales para la instancia Ejemplo 3,3,5

En el diagrama de Pareto (ver figura 12) se observa que el efecto C, Decremento de Temperatura influye de manera significativa en la función Fitness, la cual está representada en un 90% por el Makespan; así mismo en la Gráfica 13 de efectos principales se puede evidenciar que los niveles de los factores que minimizan la función Fitness son: Vecinos 10, Temperatura 80 y Decremento T 0,8.

7.2 Análisis de Varianza para la instancia ejemplo 3,7,10 de Escamilla J., Salido, Giret y Barber (2016)

La Tabla 12 corresponde al análisis de varianza arrojado por los resultados del diseño de experimento correspondientes a la instancia Ejemplo 3,7,10.

Tabla 12
Análisis de Varianza para la instancia ejemplo 3,7,10

Fuente	GL	SC Ajust.	MC Ajust.	Valor F
Modelo	7	0,001107	0,000158	0,37
Lineal	3	0,000443	0,000148	0,34
Temperatura inicial	1	0,000255	0,000255	0,60
Vecinos	1	0,000152	0,000152	0,35
Decremento de temperatura	1	0,000036	0,000036	0,08
Interacciones de 2 términos	3	0,000532	0,000177	0,41
Temperatura inicial*Vecinos	1	0,000006	0,000006	0,01
Temperatura inicial*Decremento T	1	0,000439	0,000439	1,02
Vecinos*Decremento T	1	0,000087	0,000087	0,20
Interacciones de 3 términos	1	0,000132	0,000132	0,31
Temperatura inicial*Vecinos*Decremento T	1	0,000132	0,000132	0,31
Error	16	0,006866	0,000429	
Total	23	0,007973		

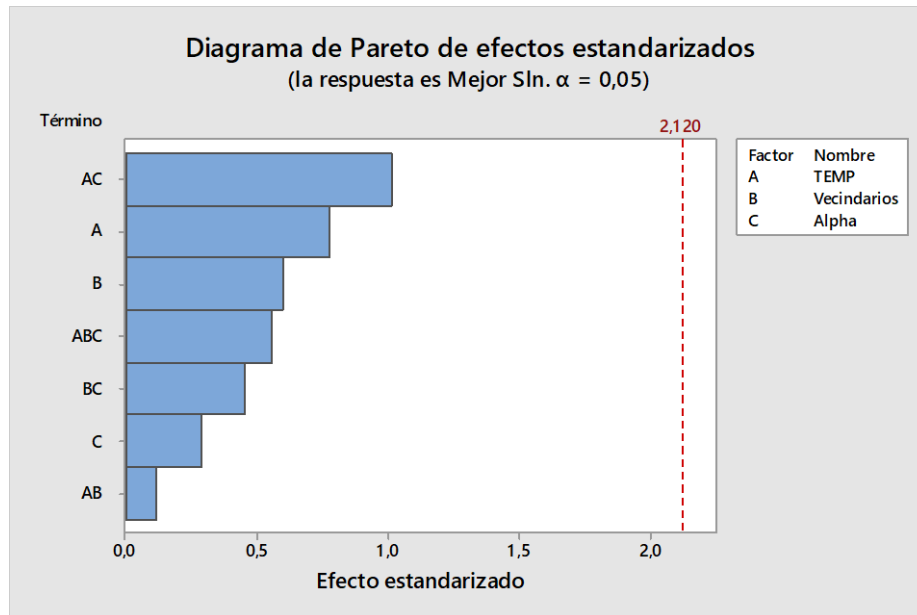


Figura 14. Diagrama de Pareto de efecto estandarizados para la instancia Ejemplo 3,7,10

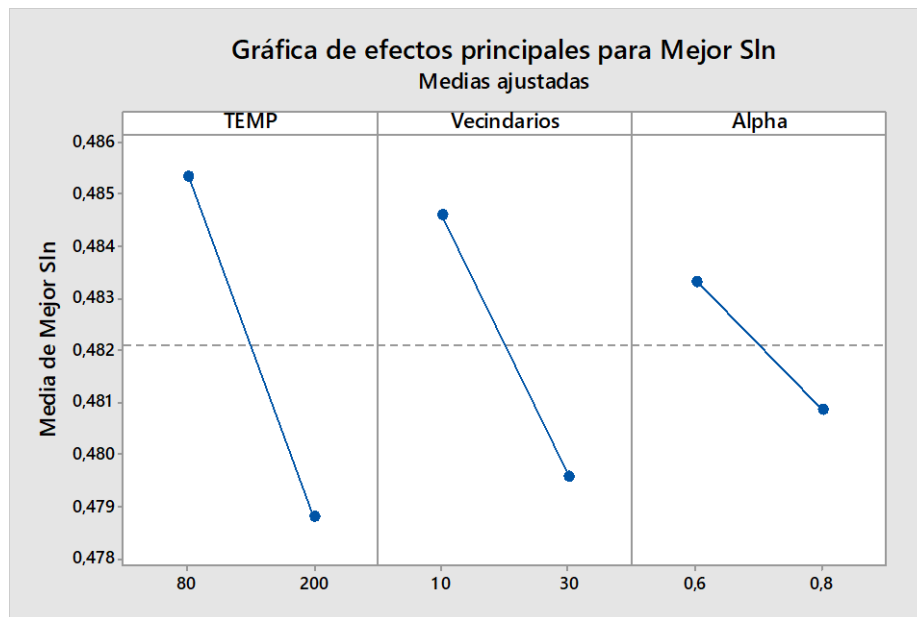


Figura 15. Gráfica de efectos principales para la instancia Ejemplo 3,7,10

En el diagrama de Pareto (ver figura 14) se observa que ninguno de los efectos influye de manera significativa en la función Fitness, la cual está representada en un 90% por el Makespan; así mismo en la Gráfica 15 de efectos principales se puede evidenciar que los niveles de los factores que minimizan la función Fitness son los más altos: Vecinos 30, Temperatura 200 y Decremento T 0,8.

7.3 Análisis de Varianza para la instancia Ejemplo 50,20,20 de Watson

Las instancias medianas fueron evaluadas con un diseño de experimento 2^3 con 5 réplicas, los factores y niveles se muestran a continuación,

Tabla 13

Factores y niveles del diseño experimental para las medianas instancias

Factores	Niveles	
	Bajo	Alto
Temperatura	50	100
Vecindario	2	4
Decremento T	0,6	0,8

Los niveles fueron seleccionados a partir de diferentes iteraciones que se realizaron con el algoritmo diseñado en Matlab, ya que al momento de utilizar los encontrados en la literatura junto los equipos personales con los que contaba el grupo de trabajo se obtuvieron tiempos de procesamientos largos (de hasta 14 horas).

Tabla 14

Análisis de Varianza para la instancia ejemplo 50,20,20 de Watson

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	7	0,004345	0,000621	0,64	0,723
Lineal	3	0,002053	0,000684	0,70	0,558
Temperatura	1	0,001068	0,001068	1,09	0,303
Vecinos	1	0,000847	0,000847	0,87	0,358
Decremento T	1	0,000137	0,000137	0,14	0,710
Interacciones de 2 términos	3	0,002291	0,000764	0,78	0,512
Temperatura*Vecinos	1	0,000857	0,000857	0,88	0,356
Temperatura*Decremento T	1	0,001410	0,001410	1,45	0,238
Vecinos*Decremento T	1	0,000024	0,000024	0,02	0,877
Interacciones de 3 términos	1	0,000001	0,000001	0,00	0,969
Temperatura*Vecinos*Decremento T	1	0,000001	0,000001	0,00	0,969
Error	32	0,031221	0,000976		
Total	39	0,035566			

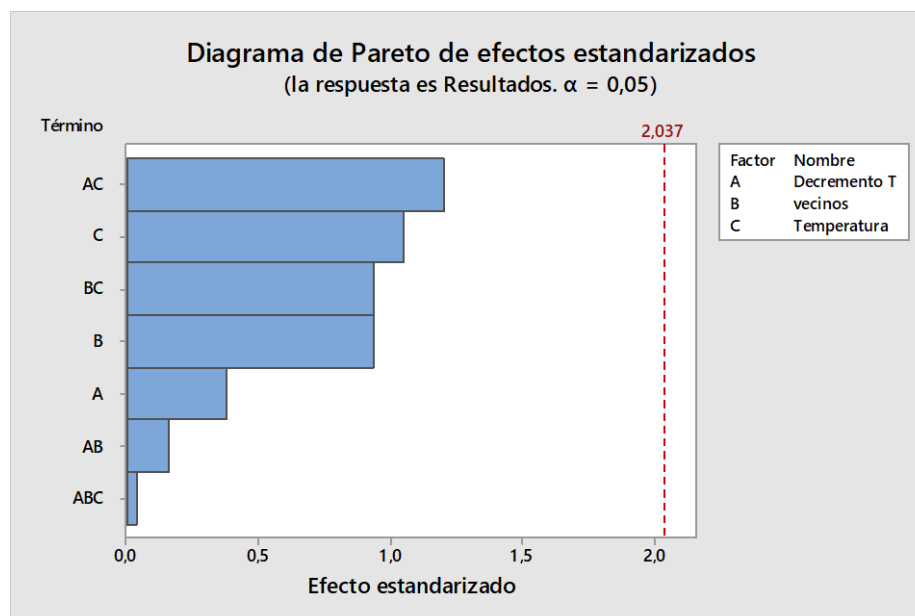


Figura 16. *Diagrama de Pareto de efecto estandarizados para la instancia Ejemplo 50,20,20*

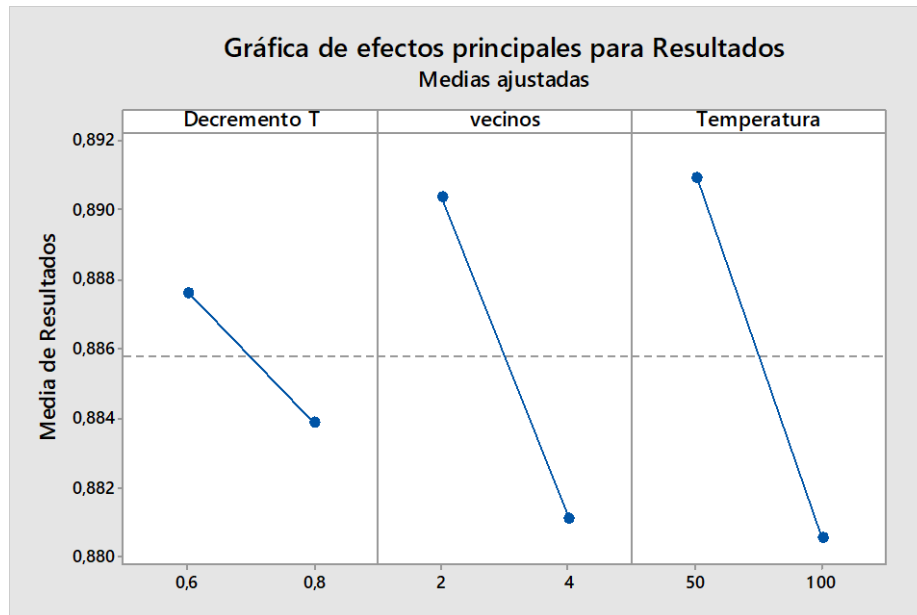


Figura 17. Gráfica de efectos principales para la instancia Ejemplo 50,20,20.

En el diagrama de Pareto (ver figura 16) se observa que ninguno de los efectos influye de manera significativa en la función Fitness, la cual está representada en un 90% por el Makespan; así mismo en la gráfica de efectos principales se puede evidenciar que los niveles de los factores que minimizan la función Fitness son los más altos: Vecinos 4, Temperatura 100 y Decremento T 0,8.

En la instancia grande no fue posible desarrollar el diseño de experimento, esto debido a que el algoritmo desarrollado en Matlab al ser corrido desde los equipos personales del grupo de trabajo tardaba como mínimo 14 horas en arrojar un solo resultado de los necesarios en el diseño factorial. Para las corridas de prueba que se lograron realizar se usaron los mismos niveles que en la instancia mediana (ver tabla 12).

8. Resultados

Para la obtención de resultados se ha ejecutado el algoritmo en las computadoras Intel (R) Core (TM) i3-9100T CPU @ 3.10GHz 3.10 GHz, 4.00 GB de memoria RAM con sistema operativo de 64 bits, procesador basado x64 y la computadora Intel(R) core (TM) i5-4210u CPU @ 1.70GHz 2.40 GHz, 6.00 GB de memoria RAM con sistema operativo 64 bits, procesador basado en x64.

En las instancias pequeñas, el ejemplo 3,3,5 (ver Apéndice M) se ejecutaba durante 45,584 minutos en promedio, caso contrario para las instancias Ejemplo 3,7,10 (ver Apéndice M), Watson 50 (ver Apéndice M), y Watson 100 (ver Apéndice M), donde su tiempo de ejecución mínimo era de 4 horas por tanto se estableció un criterio de parada de una hora (60 minutos).

En las siguientes tablas (15-22) se presentan los resultados promedios obtenidos después de las corridas, son valores promedios debido a que, la instancia contaba con 10 ejemplos y a cada ejemplo se realizó una variación de λ de 0 hasta 1, aumentando de 0,1 en 0,1, para un total de 110 corridas (ver Apéndice L). Se observa en forma de cuadro comparativo los resultados obtenidos en la presente investigación a través del Recocido Simulado 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.

En la tabla 15 los valores correspondientes a Función fitness muestran un comportamiento similar entre CP Optimizer y AG, caso contrario con los obtenidos en el recocido simulado ya que estos son mayores. En el caso del makespan el recocido simulado presenta mejores resultados en un intervalo de $\lambda \in [0,0.4]$, finalmente en la energía se observa un comportamiento similar en un intervalo de $\lambda \in [0.6,0.1]$.

En la tabla 16 se observa el GAP correspondiente al promedio de los resultados comparando el recocido simulado con CP Optimizer y AG. Cuando los valores porcentuales son negativos o positivos su conclusión varía, en $\lambda=0$ los valores correspondientes a FF son positivos, es decir, que CP Optimizer y AG arrojaron mejores resultados orientados a la eficiencia energética, por tanto, su valor de makespan fue superior. Caso contrario al recocido simulado que tuvo inclinación hacia la minimización del makespan con valores más bajos teniendo así energías más altas.

Tabla 15

Comparación de resultados Recocido Simulado Vs CP Optimizer y AG, para instancia Ejemplo 3,3,5.

Recocido Simulado				CP Optimizer			AG		
λ	Mk	E	F.F	Mk	E	F.F	Mk	E	F.F
0	65,6	106,5	0,70039	71,4	84,4	0,53762	65,8	84,4	0,553762
0,1	64,7	108,4	0,69968	65,2	84,5	0,556581	65,2	84,5	0,556581
0,2	57,7	113,1	0,69951	64,4	84,7	0,558703	64,4	84,7	0,558703
0,3	60,9	113,7	0,68503	63,2	85,2	0,559422	63,2	85,2	0,559422
0,4	54,7	113,3	0,64376	59,7	88,1	0,557816	59,7	88,1	0,557816
0,5	54	113,2	0,61197	53,9	94,3	0,547187	54,2	93,9	0,547270
0,6	55,8	115,6	0,60068	48,4	104,2	0,529935	48,9	103,2	0,529687
0,7	54,1	115,7	0,56383	45,3	111,9	0,500419	45	112,7	0,500130
0,8	54,8	115,2	0,53793	42,2	123,4	0,461368	42,2	123,5	0,461509
0,9	51,8	118,6	0,48961	41	133,2	0,414361	41	133,7	0,414712
1	52,8	114,4	0,48558	41	143,1	0,363050	41	145,3	0,363050

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

Tabla 16

GAP entre resultados de Recocido Simulado Vs CP Optimizer y AG para la instancia Ejemplo 3,3,5

	Recocido Simulado Vs CP Optimizer	Recocido Simulado Vs AG	Recocido Simulado vs CP Optimizer	Recocido Simulado vs AG	Recocido Simulado vs CP Optimizer	Recocido Simulado vs AG
λ	F.F	F.F	MK	MK	E	E
0	30,28%	26,48%	-8,12%	-0,30%	26,18%	26,18%
0,1	25,71%	25,71%	-0,77%	-0,77%	28,28%	28,28%
0,2	25,20%	25,20%	-10,40%	-10,40%	33,53%	33,53%
0,3	22,45%	22,45%	-3,64%	-3,64%	33,45%	33,45%
0,4	15,41%	15,41%	-8,38%	-8,38%	28,60%	28,60%
0,5	11,84%	11,82%	0,19%	-0,37%	20,04%	20,55%
0,6	13,35%	13,40%	15,29%	14,11%	10,94%	12,02%
0,7	12,67%	12,74%	19,43%	20,22%	3,40%	2,66%

Continuación Tabla 16

	Recocido Simulado Vs CP Optimizer	Recocido Simulado Vs AG	Recocido Simulado vs CP Optimizer	Recocido Simulado vs AG	Recocido Simulado vs CP Optimizer	Recocido Simulado vs AG
λ	F.F	F.F	MK	MK	E	E
0,8	16,59%	16,56%	29,86%	29,86%	-6,65%	-6,72%
0,9	18,16%	18,06%	26,34%	26,34%	-10,96%	-11,29%
1	33,75%	33,75%	28,78%	28,78%	-20,06%	-21,27%

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

En la Tabla 17 del problema 3,7,10 se evidencian mejores resultados para CP Optimizer y AG en FF. Cuando $\lambda \in [0,0.5]$, en el Recocido Simulado los resultados del makespan son mejores en comparación a los arrojados por CP Optimizer y AG esto demuestra que la solución mostrada por ellos está orientada a la minimización de la energía.

Tabla 17

Comparación de resultados Recocido Simulado Vs CP Optimizer y AG, para instancia Ejemplo 3,7,10

	Recocido Simulado			CP Optimizer			AG		
λ	Mk	E	F.F	Mk	E	F.F	Mk	E	F.F
0	936,8	2067,1	0,70874	1088,4	1571,4	0,533616	1006,3	1571,4	0,533616
0,1	923,1	2075,6	0,70327	999,3	1572,6	0,540932	999,3	1572,6	0,540931
0,2	891,4	2063,6	0,68358	987,2	1576,5	0,547508	987,2	1576,5	0,547509
0,3	868,7	2056	0,65621	922,2	1613,3	0,550868	926,9	1610	0,551018
0,4	815,3	2141,6	0,64646	885,9	1649,2	0,550650	891	1642,9	0,550638
0,5	831,7	2058,1	0,61003	838,8	1716	0,545249	847,5	1704,8	0,515938
0,6	833,5	2127,7	0,59138	779,1	1859,3	0,535095	782,4	1845,7	0,535361
0,7	807,6	2180,7	0,57173	708,5	2068,4	0,511331	703,9	2099,5	0,512783
0,8	797,9	2191,9	0,54259	651,8	2346	0,475184	642,4	2418,9	0,475810
0,9	819,1	2181,7	0,52534	626	2560,7	0,428228	626	2573,3	0,428603
1	818,6	2147,3	0,49812	625,9	2664,1	0,378956	626,9	2773,4	0,378956

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

En la tabla 18 se observa un comportamiento similar por parte del Recocido Simulado, CP Optimizer y AG cuando los valores de $\lambda \in [0,0.5]$, caso contrario cuando $\lambda \in [0.6,1]$ pues su comportamiento es similar en el consumo de energía.

Tabla 18

GAP entre resultados de Recocido Simulado Vs CP Optimizer y AG para la instancia Ejemplo 3,7,10

	Recocido Simulado Vs CP Optimizer	Recocido Simulado Vs AG	Recocido Simulado vs CP Optimizer	Recocido Simulado vs AG	Recocido Simulado vs CP Optimizer	Recocido Simulado vs AG
λ	F.F	F.F	MK	MK	E	E
0	32,82%	32,82%	-13,93%	-13,93%	31,55%	31,55%
0,1	30,01%	30,01%	-7,63%	-7,63%	31,99%	31,99%
0,2	24,85%	24,85%	-9,70%	-9,70%	30,90%	30,90%
0,3	19,12%	19,09%	-5,80%	-6,28%	27,70%	27,70%
0,4	17,40%	17,40%	-7,97%	-8,50%	30,35%	30,35%
0,5	11,88%	18,24%	-0,85%	-1,86%	19,94%	20,72%
0,6	10,52%	10,46%	6,98%	6,53%	14,44%	15,28%
0,7	11,81%	11,50%	13,99%	14,73%	5,43%	3,87%
0,8	14,19%	14,04%	22,41%	24,21%	-6,57%	-9,38%
0,9	22,68%	22,57%	30,85%	30,85%	-14,80%	-15,22%
1	31,45%	31,45%	30,79%	30,79%	-19,40%	-22,58%

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

En la tabla 19 se evidencian resultados favorables en el recocido simulado para $\lambda \in [0.7,1]$ en el consumo de energía de la comparación realizada con AG. Se puede observar que los valores más factibles para la FF son de CP optimizer y AG.

Tabla 19

Comparación de resultados Recocido Simulado Vs CP Optimizer y AG, para instancia de Watson Ejemplo 50,20,20

λ	Recocido Simulado			CP Optimizer			AG		
	Mk	E	F.F	Mk	E	F.F	Mk	E	F.F
0	7690,1	75287,3	0,75067	7648.9	53631.5	0.534084	8359	61290.4	0.610346
0,1	7669,2	75855	0,75883	7322.7	53669.5	0.558776	6985.1	63122.2	0.639945
0,2	7676,4	75562,7	0,76062	7258.6	53875	0.583375	6496	64934.4	0.655306

Continuación Tabla 19

Recocido Simulado				CP Optimizer			AG		
λ	Mk	E	F.F	Mk	E	F.F	Mk	E	F.F
0,3	7615,3	75797,5	0,76494	7192.5	54094.1	0.606205	6121.4	67834.6	0.667899
0,4	7610,2	75650	0,76723	7172	54257.6	0.628810	5840.1	71034	0.672533
0,5	7396,7	76058,1	0,77089	7121	54467.3	0.649284	5657.8	73653.8	0.667159
0,6	7368,7	75959,6	0,77202	6552.1	62253.7	0.665704	5559.8	75243.6	0.653968
0,7	7322,8	76151,3	0,77147	6039.1	71014.7	0.661105	5456.9	76643.2	0.634583
0,8	7396	75280,9	0,77478	5947.3	71489.3	0.647691	5466.9	76972.8	0.617737
0,9	7393,8	75860,9	0,78183	5876.3	70699.2	0.632014	5420.4	77869.3	0.595589
1	7375,7	75757,3	0,78803	5008.3	81527.7	0.531812	5396.8	78976.6	0.573094

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

Tabla 20

GAP entre resultados de Recocido Simulado Vs CP Optimizer y AG para la instancia Watson Ejemplo 50,20,20

λ	Recocido Simulado Vs CP Optimizer F.F	Recocido Simulado Vs AG F.F	Recocido Simulado vs CP Optimizer MK	Recocido Simulado vs AG MK	Recocido Simulado vs CP Optimizer E	Recocido Simulado vs AG E
0	40,55%	22,99%	0,54%	-8,00%	40,38%	22,84%
0,1	35,80%	18,58%	4,73%	9,79%	41,34%	20,17%
0,2	30,38%	16,07%	5,76%	18,17%	40,26%	16,37%
0,3	26,19%	14,53%	5,88%	24,40%	40,12%	11,74%
0,4	22,01%	14,08%	6,11%	30,31%	39,43%	6,50%
0,5	18,73%	15,55%	3,87%	30,73%	39,64%	3,26%
0,6	15,97%	18,05%	12,46%	32,54%	22,02%	0,95%
0,7	16,69%	21,57%	21,26%	34,19%	7,23%	-0,64%
0,8	19,62%	25,42%	24,36%	35,29%	5,30%	-2,20%
0,9	23,70%	31,27%	25,82%	36,41%	7,30%	-2,58%
1	48,18%	37,50%	47,27%	36,67%	-7,08%	-4,08%

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

En la Tabla 21 los valores de $\lambda \in [0,0.5]$ evidencian comportamientos similares en el makespan para Recocido Simulado y CP Optimizer aunque los mejores resultados son los correspondientes a AG. Para $\lambda \in [0.7,1]$ se observa valores más favorables de la energía por parte del Recocido Simulado.

Tabla 21

Comparación de resultados Recocido Simulado Vs CP Optimizer y AG, para instancia de Watson Ejemplo 100,20,20

Recocido Simulado				CP Optimizer			AG		
λ	Mk	E	F.F	Mk	E	F.F	Mk	E	F.F
0	12608,8	149608,9	0,75262	12671.2	105478.1	0.530618	13763.5	126772.1	0.637760
0,1	12578,8	148585,9	0,75816	12378.1	105513.8	0.561048	11700.8	128923.7	0.662490
0,2	12391,4	149060,9	0,76714	12297.4	105710.8	0.591010	11048	132690.7	0.682760
0,3	12189,2	135595,5	0,77166	12408	105654	0.622653	10347.5	139304	0.699511
0,4	12467,4	149376	0,7853	12376.4	105634.3	0.652105	9909.1	144731.7	0.703700
0,5	12477,4	149166,2	0,79475	12097.2	110089.6	0.684038	9769.5	147040.8	0.698713
0,6	12453,5	149684	0,80353	10296.9	140905.5	0.699602	9638.4	149638.5	0.690415
0,7	12429,8	148770,3	0,81241	9835.1	150495.9	0.690528	9558.8	151011.4	0.678329
0,8	12503,2	149546,9	0,82067	9785.8	150468.8	0.678396	9524.3	151835.1	0.665705
0,9	12376,1	149672,9	0,82625	9813.2	149937.6	0.669925	9484.8	153036.8	0.651611
1	12397	150238,8	0,83669	9434.5	152047.9	0.635030	9511	153336.4	0.640230

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

En la Tabla 22 se evidencian comportamientos similares por parte de Recocido Simulado y CP Optimizer en el MK cuando $\lambda \in [0,0.5]$, por otra parte, cuando $\lambda \in [0.6,1]$ se evidencia comportamientos similares en la E entre Recocido simulado y AG.

Tabla 22

GAP entre resultados de Recocido Simulado Vs CP Optimizer y AG para la instancia Watson Ejemplo 100,20,20

	Recocido Simulado Vs CP Optimizer	Recocido Simulado Vs AG	Recocido Simulado vs CP Optimizer	Recocido Simulado vs AG	Recocido Simulado vs CP Optimizer	Recocido Simulado vs AG
λ	F.F	F.F	MK	MK	E	E
0	41,84%	18,01%	-0,49%	-8,39%	41,84%	18,01%
0,1	35,13%	14,44%	1,62%	7,50%	40,82%	15,25%
0,2	29,80%	12,36%	0,76%	12,16%	41,01%	12,34%
0,3	23,93%	10,31%	-1,76%	17,80%	28,34%	-2,66%
0,4	20,43%	11,60%	0,74%	25,82%	41,41%	-15,81%
0,5	16,19%	13,74%	3,14%	27,72%	35,50%	1,45%
0,6	14,86%	16,38%	20,94%	29,21%	6,23%	0,03%
0,7	17,65%	19,77%	26,38%	30,04%	-1,15%	-1,48%
0,8	20,97%	23,28%	27,77%	31,28%	-0,61%	-1,51%
0,9	23,33%	26,80%	26,12%	30,48%	-0,18%	-2,20%
1	31,76%	30,69%	31,40%	30,34%	-1,19%	-2,02%

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

Para instancias pequeñas (Ver figura 18), se puede observar que los tres algoritmos tienen el mismo comportamiento, es decir, cuando λ aumenta la función fitness va disminuyendo su valor.

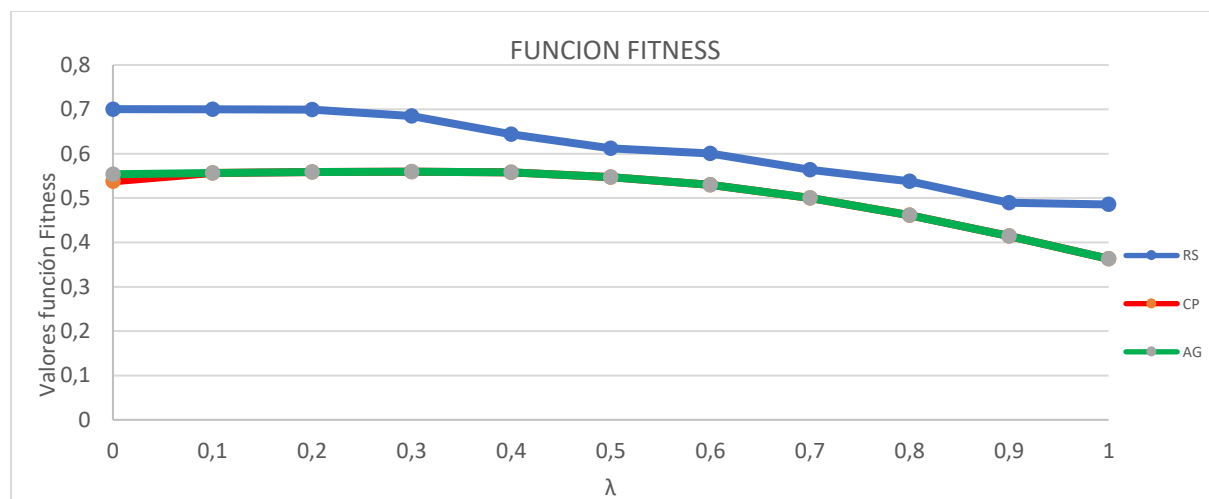


Figura 18. Comportamiento de los valores promedio de la Función Fitness para las instancias pequeñas, Nota: RS: Recocido Simulado, CP: CP Optimizer, AG: Algoritmo Genético

Para instancias grandes (Ver figura 19), se puede observar que los algoritmos de la literatura tienen el mismo comportamiento, es decir, cuando λ aumenta la función fitness va disminuyendo su valor, caso contrario en el recocido simulado dado que este valor aumenta.

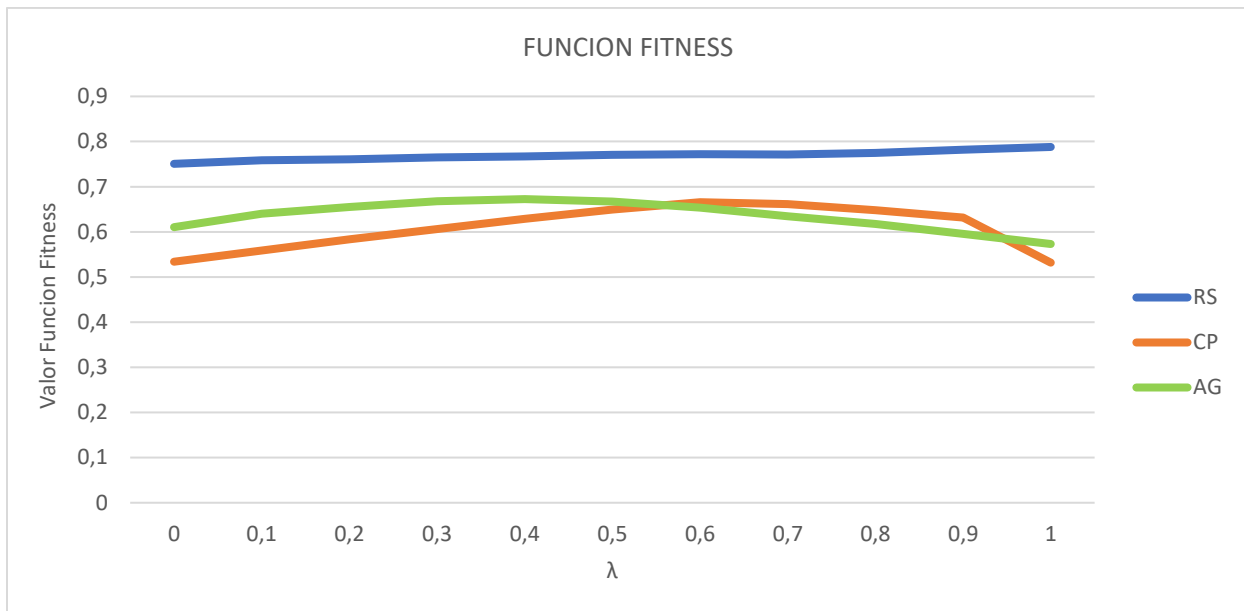


Figura 19. Comportamiento de los valores promedio de la Función Fitness para las instancias grandes
Nota: RS: Recocido Simulado, CP: CP Optimizer, AG: Algoritmo Genético

9. Conclusiones

Durante la revisión de literatura se observó que el Problema Job Shop con eficiencia energética ha sido poco estudiado, a pesar de esto, el interés por su estudio ha venido aumentando en los últimos años pues su enfoque no solo es reducir los costos si no también la huella de carbono.

En el proceso de programación del algoritmo se observó que la función del makespan era la que consumía más tiempo computacional, esto limitó la evaluación y exploración de instancias más grandes.

En las instancias pequeñas se observó que cuando $\lambda \in [0,0.5]$ los valores del makespan son mejores hasta en un 13% a los encontrados en la literatura, sin embargo, en la energía no sucede lo mismo por tanto la función fitness no alcanzar a tener mejores resultados. Para las instancias grandes el algoritmo presentó dificultades al momento de encontrar mejores soluciones en términos de makespan y energía.

Teniendo en cuenta los resultados obtenidos de la función fitness en todas las instancias concluimos que, si bien nuestros resultados no fueron mejores a los encontrados en la literatura, siguen siendo factibles dado que resuelven el problema “Job Shop con eficiencia energética” mediante la metaheurística del recocido simulado.

10. Recomendaciones

Para futuras investigación se sugiere que se tomen en consideración el consumo de energía de las máquinas durante los tiempos ociosos, para que de esta forma el ejercicio se pueda aproximar un poco más a entornos reales. Así mismo se aconseja documentar los tiempos computacionales, esto debido a que, permitiría realizar un análisis más completo.

También se recomienda realizar las corridas del algoritmo en equipos de cómputo más eficientes que permitan efectuar un análisis más profundo como el de una exploración más amplia con vecindarios y temperaturas más grandes para así obtener mejores soluciones en tiempos computacionales más cortos.

Se recomienda modificar la estructura del algoritmo, partiendo de una mejor solución, que permita reducir los tiempos de cómputo.

Motivar en los estudiantes de la Escuela de Estudios Industriales y Empresariales de la Universidad Industrial de Santander la adquisición de conocimientos y habilidades en lenguajes de programación, ya que esto permitiría que se lograsen construir algoritmos más eficientes y, también llevar a cabo un proceso de aprendizaje más satisfactorio.

Referencias Bibliográficas

- Abdolrazzagh-Nezhad, M., & Abdullah, S. (2017). Job Shop Scheduling: Classification, Constraints and. *International Journal of Computer and Information Engineering*, 6.
- Abedi , M., Chiong, R., Noman, N., & Zhang, R. (2017). A hybrid particle swarm optimisation approach for energy-efficient single machine scheduling with cumulative deterioration and multiple maintenance. *IEEE Symposium Series on Computational Intelligence (SSCI)*, 1-8.
- Agencia Internacional de la Energía. (2019). *Grupo Banco Mundial*. Obtenido de Grupo Banco Mundial:
datos.bancomundial.org/indicador/EN.CO2.MANF.ZS?end=2014&name_desc=false&start=2002
- Aguilar Imitola, K. J., & Pérez Diaz, Y. T. (2013). *Un algoritmo memético para la minimización del makespan en el problema del Job Shop Scheduling*. Bucaramanga.
- Aparicio Torres, Y. A., & González Patiño, C. (2016). *Metaheurística Greedy Randomized adaptive search procedure (GRAPS- procedimiento de búsqueda codiciosa aleatorizada y adaptativa) aplicada al JSSP*. Bucaramanga.
- Ashrafi , M., Davoudpour, H., & Abbassi, M. (2014). Investigating the Efficiency of GRASP for the SDST HFS with Controllable Processing Times and Assignable Due Dates. *Handbook of Research on Novel Soft Computing Intelligent Algorithms*, 30.
- Babak, A., Shahram, S., & Jamal, A. (2006). Bi-objective resource-constrained project scheduling with robustness and makespan criteria. *Applied Mathematics and Computation* , 7.
- Barba Rodríguez, I. (s.f). *Algoritmos de planificación basados en restricciones para la sustitución de componentes defectuosos*. Sevilla: Universidad de Sevilla.
- Beasley, D., Bull, D., & Martin, R. (1993). An overview of genetic algorithms: Part 1. *University computing* 15, 58-58.

- Błażewicz, J. (1986). *Scheduling under resource constraints : deterministic models*. Basel : Baltzer.
- Blazewicz, J., Pesch, E., & Sterna, M. (2000). The disjunctive graph machine representation of the job shop scheduling problem. *European Journal of Operational Research*, 317-331.
- Büyüksaatçi, S., & Baray, A. (2014). A comparison of three nature-inspired metaheuristics for the facility layout problem (opti 2014). *An International Conference on Engineering and Applied Sciences Optimization*, 1032-1043.
- Che, A., Wu, X., Peng, J., & Yan, P. (2017). Energy-efficient bi-objective single-machine scheduling with power-down mechanism. *Computers and Operations Research*, 172-183.
- 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*.
- Cheung, W., & Zhou, H. (2001). Using Genetic Algorithms and Heuristics for Job Shop Scheduling with Sequence-Dependent Setup Times. *Annals of Operations Research*, 65-81.
- 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*, 146-168.
- Cortez, A. (2004). Teoría de la complejidad computacional y teoría de la computabilidad. *Revista de investigación de Sistemas e Informática*, 69.
- Dai, M., Tang, D., Giret, A., Salido, M. A., & Li, W. (2013). Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, 418-429.
- Dai, M., Wang, Y., & Ji, Z. (2018). Energy-aware integrated optimization of process planning and scheduling considering transportation. *Modern Physics Letters B*, 32.
- Delgado, J. E. (2012). *Optimización de la programación en Talleres de Mecanizado*. Madrid: Universidad Politécnica de Madrid.

- Duarte Muñoz, A., Pantrigo Fernández, J. J., & Gallego Carrillo, M. (2007). *Metahuerísticas*. Madrid: DYKINSON.
- Duarte Muñoz, A., Pantrigo Fernández, J. J., & Gallego Carrillo, M. (2007). *Metahuerísticas*. Madrid: Dykinson.
- Escamilla , J., Salido, M., Barber, F., Giret, A., Tang, D., & Dai, M. (2016). Energy efficiency, robustness, and makespan optimality in job-shop scheduling problems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 300-3012.
- Escamilla Fuster, J. (2016). *Eficiencia Energética y Robustez en Problemas de Scheduling*. Valencia.
- Escamilla Fuster, J. (2016). Eficiencia Energética y Robustez en Problemas de Scheduling. *Universidad Politecnica de Valencia*, 1-64.
- Escamilla, J., Salido, M. A., Giret, A., & Barber, F. (2016). A Metaheuristic Technique for Energy-Efficiency in Job-Shop Scheduling. *The Knowledge Engineering Review*, 50.
- Frutos, M. (2012). *Tópicos avanzados en diseño y desarrollo de técnicas meta-heurísticas para uso en la secuenciación de operaciones*. Buenos Aires: EdiUNS.
- Garey, M., D, J., & Sethi, R. (1976). The complexity of flowshop and job shop. *Mathematical Operation Research 1*, 117-129.
- Glover, F., & Laguna , M. (1997). *Tabu search*. Boston: Kluwer academic.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine learning*. Addison-Weasley Longman, 372.
- Gómez Moreno, J. D., & Orduz González, E. A. (2015). *Minimización del Makespan en el problema de Job Shop Flexible con restricciones de transporte utilizando algoritmo genético*. Bucaramanga.
- González, M., Oddi, A., & Rasconi, R. (2019). Efficient Approaches for Solving a Multiobjective Energy-aware Job Shop Scheduling Problem. *Fundamenta Informaticae*, 93-132.
- Graham, R. L. (1966). Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 1563-1581.

- Graham, R., Lawler, E., Lenstra, J., & RinnooyKan, A. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. *Annals of Discrete Mathematics*, 287-326.
- Gutowski, T., & Dahmus, J. (2004). An environmental analysis of machining. *ASME International Mechanical Engineering Congress and RD&D Exposition, Anaheim,*.
- Han, W., Zhigang, J., Yan, W., Hua, Z., & Yanhong, W. (2018). A two-stage optimization method for energy-saving flexible job-shop scheduling based on energy dynamic characterization. *Journal of Cleaner Production*, 575-588.
- Huang, K. L., & Liao, C. J. (2008). Ant colony optimization combined with taboo search for. *Computers & Operations Research* 35.4, 1030-1046.
- Jaimes Sandoval, P. J., & Rivera González, J. V. (2015). *Estudio del impacto del tamaño de lote de transferencia en el desempeño de un sistema productivo tipo Job Shop considerando tiempos de transporte*. Bucaramanga.
- Jiang , E., & Wang, L. (2019). A Modified MOEA/D for Energy-efficient Flexible Job Shop Scheduling Problem. *The Institute of Electrical and Electronic Engineers (IEEE)*.
- Jiang , T., Zhang , C., & Qi-Ming, S. (2019). Green Job Shop Scheduling Problem with Discrete Whale Optimization Algorithm. *Institute of Electrical and Electronic Engineers (IEEE)*, 43153 - 43166.
- Jiang, T., Zhang, C., Zhu, H., Gu, J., & Deng, G. (2018). Energy-Efficient Scheduling for a Job Shop Using an Improved Whale Optimization Algorithm. *Mathematics*.
- Jiang, T., Zhang, C., Zhu, H., Gu, J., & Deng, G. (2018). Energy-Efficient Scheduling for a Job Shop Using an Improved Whale Optimization Algorithm. *mathematics*.
- Kirkpatrick, S., Vecchi, M. P., & Gelatt, C. D. (1983). Optimization by Simulated Annealing. *Science*, 671-680.
- Li, W., Zein, A., Kara, S., & Herrmann, C. (2011). An investigation into fixed energy consumption of machine tools. *Glocalized Solutions for Sustainability in Manufacturing*, 268-273.

- 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*, 87-96.
- Liu, Y., Lohse, N., Petrovic, S., & Gindy, N. (2013). An Investigation into Minimising Total Energy Consumption, Total Energy Cost and Total Tardiness Based on a Rolling Blackout Policy in a Job Shop. *IFIP International Conference on Advances in Production Management Systems*, 103-11.
- Lozano, L. M., & Torres, F. A. (2018). *Solución al "Flexible Job Shop Problem" con tiempos de alistamiento dependientes de la secuencia mediante un algoritmo Híbrido Genético*. Bucaramanga.
- Lui, B., Wang, L., & Jin, Y. (2007). An effective PSO- based memetic algorithm for flow shop scheduling. *IEEE Transactions*, 18-27.
- Masmoudi, O., Delorme, X., & Gianessi, P. (2019). Job-shop scheduling problem with energy consideration. *International Journal of Production Economics*, 12-22.
- May, G., Stahl, B., Taisch, M., & Prabhu, V. (2015). Multi-objective genetic algorithm for energy-efficient job shop scheduling. *International Journal of Production Research*, 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*, 105-116.
- Mouzon, G., Yildirim, M., & Twomey, J. (2007). Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of Production Research*, 4247-4271.
- Neugebauer, R., Wabner, M., Rentzsch, H., & Ihlenfeldt, S. (2011). Structure principles of energy efficient machine tools. *CIRP Journal of Manufacturing Science and Technology* 4.2, 136-147.
- Osman, I., & Laporte, G. (1996). *Annals of Operations Research*. Science Publishers, 511–623.

- Osorio, J., Lasso, D., & Ruiz, G. (2012). JOB SHOP SCHEDULING BI OBJETIVO MEDIANTE ENFRIAMIENTO SIMULADO Y ENFOQUE DE PARETO. *Ingenierías de Medellín*, 114-126.
- Pérez Jiménez, M., & Sancho Caparrini, F. (2003). Máquinas Moleculares Basadas en ADN. En M. Pérez Jiménez, & F. Sancho Caparrini, *Máquinas Moleculares Basadas en ADN* (pág. 195). Sevilla: Universidad de Sevilla.
- Quan, G., & Hu, X. (2001). Energy efficient fixed-priority scheduling for real-time systems on variable voltage processors. *Design Automation Conference*, 828-833.
- Rayward-Smith, V. J., Osman, I. H., Reeves, C. R., & Smith, G. D. (1996). *Modern Heuristic Search Methods*. New York: John Wiley y Sons.
- Rodrigue, J. P. (2013). *The geography of transport systems*. Routledge.
- Salazar-Hornig, E., & C. Medina, S. (2013). Makespan Minimization for The Identical Machine Parallel Shop with Sequence Dependent Setup Times Using a Genetic Algorithm. *Ingeniería, Investigación y Tecnología*, 138.
- Salido, M. A., Escamilla, J., Barber, F., Giret, A., Tang, D., & Dai, M. (2013). Energy-aware Parameters in Job-shop Scheduling Problems. *International Joint Conference on Artificial Intelligence*. Beijing, China.
- Salido, M. A., Escamilla, J., Giret, A., & Barber, F. (2016). A genetic algorithm for energy-efficiency in job-shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 1303–1314.
- Sarmiento Ardila, C. (2010). *Método Particle Swarm Optimization (PSO-Enjambre de Partículas) aplicado al problema de múltiples objetivos del Job Shop Scheduling (JSP) o Secuenciamiento de Máquinas*. Bucaramanga.
- Sarmiento Ardila, C. J. (2012). *Método Particle Swarm Optimization (PSO - enjambre de partículas) aplicado al problema de múltiples objetivos del Job Shop Scheduling (JSP) o secuenciamiento de máquinas*. Bucaramanga.

- Sarria Cerro, V. M. (2010). *Metaheurísticas aplicadas al problema QAP. Estudio y experiencia computacional*. Barcelona: Escuela Técnica Superior de Ingeniería Industrial y Aeronáutica de Terrassa.
- Seo, E., Jeong, J., Seonyeong, P., & Lee, J. (2008). Energy efficient scheduling of real-time tasks on multicore processors. *Parallel and Distributed Systems, IEEE Transactions on 19.11*, 1540-1552.
- Seow, Y., & Rahimifard, S. (2011). «A framework for modelling energy consumption within. *CIRP Journal of Manufacturing Science and Technology 4.3*, págs. 258-264.
- Shrouf, F., Ordieres-Meré, J., García-Sánchez, Á., & Ortega-Mier, M. (2014). Optimizing the production scheduling of a single machine to minimize total energy consumption costs. *Journal of Cleaner Production*, 197-207.
- Sierra Sánchez, M. R. (2010). *Mejora de Algoritmos de búsqueda Heurística Mediante Poda por Dominancia. Aplicación a Problemas Scheduling*. Oviedo: Universidad de Oviedo.
- Tang, D., & Dai, M. (2015). Energy-efficient approach to minimizing the energy consumption in an extended job-shop scheduling problem. *Chinese Journal of Mechanical Engineering*, 1048–1055.
- Tang, D., Dai, M., Giret, A., & Salido, M. (2019). Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. *Robotics and Computer-Integrated Manufacturing*, 143-157.
- Tang, D., Dai, M., Salido, M., & Giret, A. (2016). Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. *Computers in Industry*, 82-95.
- Tiwari, A., & Liu, Y. (2015). An Investigation into Minimising Total Energy Consumption and Total Completion Time in a Flexible Job Shop for Recycling Carbon Fiber Reinforced Polymer. *Procedia CIRP*, 722-727.
- Ulusam Sec, kiner , S., & Kurt, M. (2007). A simulated Annealing Approach to the Solution of job Rotation Scheduling Problems. *Applied Mathematics and Computation*, 15.

- Vallejos Cifuentes, P. (2017). *Uso de Algoritmos Genéticos para el logro de la eficiencia energética en plantas de manufactura bajo un enfoque de Programación de Operaciones*. Medellín, Antioquia, Colombia: Universidad Nacional de Colombia.
- Vallejos Cifuentes, P. E. (2017). *Uso de algoritmos genéticos para el logro de la eficiencia energética en plantas de manufactura bajo un enfoque de programación de operaciones*. Medellín.
- Vicentini , F., & Puddu, S. (2003). Algoritmos heurísticos y el problema. *Facultad de Ciencias Exactas y Naturales Universidad de Buenos Aires*, 7.
- Vieira , G., J.W, H., & Lin , A. (2003). Rescheduling manufacturing systems:a framework of strategies, policies, and method. *Journal of Scheduling*.
- Wang, B., Huang, K., & Li, T. (2018). Two-stage hybrid flowshop scheduling with simultaneous processing machines. *Journal of Scheduling*, 387–411.
- Weinert, N., Chiotellis, S., & Seliger, G. (2011). Methodology for planning and operating. *CIRP Annals-Manufacturing Technology*, 41-44.
- Wenbin, G., Yuxin, L., & Yi, W. (2017). Energy-efficient job shop scheduling problem using an improved particle swarm algorithm. *IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, 830-834.
- 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*, 86-101.
- Yin, L., Li, X., Gao, L., Zhang, Z., & Lu, C. (2017). Energy-efficient job shop scheduling problem with variable spindle speed using a novel multi-objective algorithm. *Advances in Mechanical Engineering*, 1-21.
- Zandieh, M., Mahdavi, I., & Bagheri, A. (2008). Solving the Flexible job-shop Scheduling Problem by a Genetic Algorithm. *Journal of Applied Sciences*, 1812-5654.

- Zhang , Z., Wu , L., Peng , T., & Jia, S. (2018). An improved scheduling approach for minimizing total energy consumption and makespan in a flexible job shop environment. *Sustainability*, 179.
- Zhang, C., Jiang, T., Zhu, H., & Gu, J. (2018). Energy-Efficient Scheduling for a Job Shop Using an Improved Whale Optimization Algorithm. *sustainability*.
- Zhang, L., Tang, Q., Wu, Z., & Wang, F. (2017). Mathematical modeling and evolutionary generation of rule sets for energy-efficient flexible job shops. *Energy*, 210-227.
- 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*, 3361-3375.
- Zhou, S., Li, X., Du, N., Pang, Y., & Chen, H. (2018). A multi-objective differential evolution algorithm for parallel batch processing machine scheduling considering electricity consumption cost. *Computers & Operations Research*, 55-68.
- Zineb Ibn , M. H., Barkany, A. E., Ikram , E. A., Jabri, A., & Darcherif, A. M. (2019). New model of planning and scheduling for job-shop production system with energy consideration. *Management and Production Engineering Review*, 89-97.