

PROPUESTA DE UN ALGORITMO EVOLUTIVO PARA EL PROBLEMA DE
SECUENCIACIÓN DE LA PRODUCCIÓN EN AMBIENTE DE MANUFACTURA
FLOWSHOP CON RESTRICCIÓN DE TIEMPOS DE ALISTAMIENTO

Zaine Johanna Madrid Plata



Universidad Pontificia Bolivariana Seccional Bucaramanga
Escuela de Ingeniería
Facultad de Ingeniería Industrial
Piedecuesta
2018

PROPUESTA DE UN ALGORITMO EVOLUTIVO PARA EL PROBLEMA DE
SECUENCIACIÓN DE LA PRODUCCION EN AMBIENTE DE MANUFACTURA
FLOWSHOP CON RESTRICCIÓN DE TIEMPOS DE ALISTAMIENTO

Proyecto de grado presentado como requisito para optar el título de:
INGENIERA INDUSTRIAL

Zaine Johanna Madrid Plata

Director del Proyecto:
M.Sc. Orlando Federico González Casallas



Universidad Pontificia Bolivariana Seccional Bucaramanga
Escuela de Ingeniería
Facultad de Ingeniería Industrial
Piedecuesta
2018

DEDICATORIA

El presente proyecto está dedicado a mis padres, por el amor y apoyo incondicional que me brindan, por ser el pilar fundamental de todo lo que soy. Todo este trabajo es gracias a ellos.

A mis hermanos por ser mis compañeros de vida y a quienes amo sinceramente por siempre.

AGRADECIMIENTOS

A mi Director de Proyecto Gonzales Casallas, Orlando Federico MSc, quien por vocación y orientación contribuyo de manera significativa en la culminación del presente proyecto de pregrado; por creer en mis capacidades y depositarme su confianza.

A mi hermano Madrid Plata, Jorge Luis y Paredes, Alberto Jorge, mi hermano y mi amigo que por su gran experiencia, conocimiento y amabilidad me han guiado en el proyecto.

LISTA DE CONTENIDO

	Pág.
LISTA DE TABLAS.....	9
LISTA DE FIGURAS.....	11
LISTA DE ILUSTRACIONES.....	14
LISTA DE ANEXOS.....	15
RESUMEN	16
INTRODUCCION	18
1. PLANTEAMIENTO DEL PROBLEMA.....	20
1.1. Descripción del problema	20
1.2. Formulación del problema	21
1.3. Justificación	22
2. OBJETIVOS.....	23
2.1. Objetivo general.....	23
2.2. Objetivos específicos	23
3. MARCO REFERENCIAL.....	24
3.1. Antecedentes	24
3.1.1. Sistemas de manufactura.....	24
3.1.2. Descripción del sistema HAS-200.....	24
3.1.3. Descripción del proceso.....	26
3.1.4. Estaciones de trabajo de la HAS-200.....	27
3.1.4.1. Estación de producción.....	27
3.1.4.2. Estación de control y almacenamiento.....	28
3.2. Marco teórico	28
3.2.1. Programación de operaciones.....	28
3.2.2. Taller de producción continua.....	29
3.3. Problemas de configuración de talleres de flujo continuo.....	30
3.3.1. Flowshop.....	30
3.3.2. Flowshop con doble máquina.....	31

3.3.2.1. Modelo con retraso de tiempo.....	32
3.3.3. Tiempo de procesamiento.....	33
3.3.4. Tiempos de alistamiento.....	33
3.3.5. Minimización del Makespan.....	33
3.3.6. Soluciones algorítmicas.....	34
3.3.7. Métodos exactos.....	35
3.3.8. Métodos Heurísticos.....	35
3.3.9. Algoritmo de Clarke and Wright.....	37
3.3.9.1. Descripción del modelo de Clarke y Wright.....	37
3.3.9.2. Metodología del algoritmo.....	38
3.3.10. Formulación matemática.....	39
3.3.11. Pseudocódigo del Algoritmo de Clarke and Wright.....	41
3.3.12. Soluciones Metaheurística.....	41
3.3.12.1. Métodos metaheurísticos.....	42
3.3.12.2. Métodos evolutivos.....	43
3.3.13. Algoritmo genético.....	43
3.3.14. Teoría de Holland.....	44
3.3.14.1. Terminología los Algoritmos genéticos.....	45
3.3.15. Operadores de los Algoritmos genéticos.....	46
3.3.15.1. Selección.....	46
3.3.15.2. Cruzamiento.....	47
3.3.15.3. Mutación.....	48
3.3.16. Pseudocódigo del Algoritmo Genético.....	49
3.4. Diseño de Experimentos.....	50
3.5. Estructuras del diseño experimental.....	51
3.5.1. Estructura de Tratamiento.....	51
3.6. Diseño factorial fraccionado 2^k	52
3.6.1. La fracción un medio del diseño 2^k	53
3.6.1.1. Diseños de resolución IV.....	53
4. DISEÑO METODOLÓGICO.....	54
4.1. Diseño de la investigación.....	54

4.2. Método de investigación	55
4.3. Población.....	55
4.3.1. Tamaño de la muestra.	56
4.4. Definición de las variables de investigación.....	56
4.5. Hipótesis	56
5. RESULTADOS	58
5.1. Modelamiento del Algoritmo Clarke & Wright.....	58
5.2. Propuesta del Algoritmo Genético	60
5.2.1. Generación de la población inicial.....	61
5.2.2. Cruzamiento.....	61
5.2.3. Mutación.	62
5.2.4. Nueva población.	63
5.3. Implementación del AG a la celda de manufactura HAS-200.....	63
5.4. Pseudocódigo del programa propuesto (C&W – AG)	64
6. ANÁLISIS DE LOS RESULTADOS	67
6.1. Diseño de experimentos (DOE) para validar el AG.....	67
6.1.1. Resultados experimentales con tiempos de alistamiento en cero.	69
6.1.1.1. Análisis de varianza para instancias Tai_SDST.	69
6.1.1.2. Análisis de varianza para instancias 20X5_1.....	69
6.1.1.3. Análisis de varianza para instancias 20X5_3.....	71
6.1.1.4. Análisis de varianza para instancias 20X5_9.....	73
6.1.1.5. Análisis de varianza para instancia 20X20_1.	75
6.1.1.6. Análisis de varianza instancias 20X20_3.	77
6.1.1.7. Análisis de varianza para instancia 20X20_10	79
6.1.1.8. Análisis de varianza para instancia 50X5_2.....	80
6.1.1.9. Análisis de varianza de instancia 50X5_4	82
6.1.1.10. Análisis de varianza de instancia 50X5_6	84
6.1.2. Resultados experimentales con tiempos de alistamiento.	86
6.1.2.1. Análisis de varianza para instancias SDST10.....	86
6.1.2.2. Análisis de varianza para instancias Ta001_M1.....	86
6.1.2.3. Análisis de varianza para instancias Ta003_M4.....	87

6.1.2.4. Análisis de varianza para instancia Ta009_M4.	89
6.1.2.5. Análisis de varianza instancias Ta021_M0.....	90
6.1.2.6. Análisis de varianza para instancia Ta023_M12.	92
6.1.2.7. Análisis de varianza para instancia Ta030_M9	93
6.1.2.8. Análisis de varianza de instancia Ta032_M2.	94
6.1.2.9. Análisis de varianza de instancia Ta034_M0.	96
6.1.2.10. Análisis de varianza de instancia Ta036_M6.	97
7. CONCLUSIONES.....	100
8. RECOMENDACIONES.....	102
REFERENCIAS.....	103
ANEXOS	109

LISTA DE TABLAS

	Pág.
<i>Tabla 1.</i> Software del sistema HAS-200.....	25
<i>Tabla 2.</i> Tipos de problemas Flowshop	31
<i>Tabla 3.</i> Demanda requerida por clientes (Ejemplo)	40
<i>Tabla 4.</i> Tabla de distancias de localidades	40
<i>Tabla 5.</i> Pseudocódigo Clarke and Wright	41
<i>Tabla 6.</i> Pseudocódigo Algoritmo Genético.....	49
<i>Tabla 7.</i> Relaciones de alias para diseños factoriales fraccionados 2^{k-p} con $k \leq 15$	53
<i>Tabla 8.</i> Vector Tiempos de procesamiento de los trabajos. (Ejemplo)	59
<i>Tabla 9.</i> Matriz de tiempos de alistamiento. (Ejemplo).....	59
<i>Tabla 10.</i> Parámetros del Algoritmo Genético	60
<i>Tabla 11.</i> Generación de la población inicial.....	61
<i>Tabla 12.</i> Cruzamiento del AG.....	62
<i>Tabla 13.</i> Parámetros para generar la mejor solución (Ejemplo).....	62
<i>Tabla 14.</i> Mejor secuencia (Ejemplo).....	62
<i>Tabla 15.</i> Tiempos de procesamiento de la HAS-200.	63
<i>Tabla 16.</i> Tiempos de Alistamiento de la HAS-200.....	64
<i>Tabla 17.</i> Variables para el arreglo experimental factorial 2^K fraccionado.	67
<i>Tabla 18.</i> Instancias con tiempos de alistamiento en cero.	68
<i>Tabla 19.</i> Instancias con tiempos de alistamiento.....	69
<i>Tabla 20.</i> Análisis de varianza de la instancia 20X5_1.	70
<i>Tabla 21.</i> Análisis de varianza para la instancia 20X5_3.....	72

<i>Tabla 22.</i> Análisis de varianza para instancias 20X5_9.....	73
<i>Tabla 23.</i> Análisis de varianza para instancia 20X20_1.....	75
<i>Tabla 24.</i> Análisis de varianza para las instancias 20X20_3.....	77
<i>Tabla 25.</i> Análisis de varianza para instancia 20X20_10.....	79
<i>Tabla 26.</i> Análisis de varianza para instancias 50X5_2.....	80
<i>Tabla 27.</i> Análisis de varianza de la instancia 50X5_4.....	82
<i>Tabla 28.</i> Análisis de varianza para la instancia 50X5_6.....	84
<i>Tabla 29.</i> Análisis de varianza para la instancia Ta001_M1.....	86
<i>Tabla 30.</i> Análisis de varianza para la instancia Ta003_M4.....	88
<i>Tabla 31.</i> Análisis de varianza para instancias Ta009_M4.....	89
<i>Tabla 32.</i> Análisis de varianza para la instancia Ta021_M0.....	91
<i>Tabla 33.</i> Análisis de varianza para instancia Ta023_M12.....	92
<i>Tabla 34.</i> Análisis de varianza para instancia Ta030_M9.....	93
<i>Tabla 35.</i> Análisis de varianza para la instancia Ta032_M2.....	95
<i>Tabla 36.</i> Análisis de varianza para la instancia Ta034_M0.....	96
<i>Tabla 37.</i> Análisis de varianza para instancia Ta036_M6.....	97

LISTA DE FIGURAS

	Pág.
<i>Figura 1.</i> Referencias de la HAS200.	26
<i>Figura 2.</i> Estación de producción.	33
<i>Figura 3.</i> Interacciones del c_{\max} para las instancias 20X5_1.	70
<i>Figura 4.</i> Efectores principales para el C_{\max} de la instancia 20X5_1	71
<i>Figura 5.</i> Diagrama de Pareto para el tiempo computacional del AG instancias 20X5_1.	71
<i>Figura 6.</i> Interacciones del c_{\max} para las instancias 20X5_3.	72
<i>Figura 7.</i> Efectos principales para el C_{\max} de la instancia 20X5_3.	73
<i>Figura 8.</i> Diagrama de Pareto efectos del tiempo computacional de la instancia 20X5_3.	73
<i>Figura 9.</i> Interacciones del c_{\max} para las instancias 20X5_9	74
<i>Figura 10.</i> Efectos principales para el C_{\max} de las instancias 20X5_9.	74
<i>Figura 11.</i> Diagrama de Pareto efectos del tiempo computacional de la instancia 20X5_9 ...	75
<i>Figura 12.</i> Interacciones del c_{\max} para las instancias 20X20_1	76
<i>Figura 13.</i> Efectos principales para instancias 20X20_1.	76
<i>Figura 14.</i> Diagrama de Pareto efectos del tiempo computacional de instancia 20X20_1	77
<i>Figura 15.</i> Interacciones del c_{\max} para las instancias 20X20_3	78
<i>Figura 16.</i> Efectos principales para las instancias 20X20_3.	78
<i>Figura 17.</i> Diagrama de Pareto efectos del tiempo computacional de instancia 20X20_3	78
<i>Figura 18.</i> Interacciones del c_{\max} para las instancias 20X20_10	79
<i>Figura 19.</i> Grafica de efectos principales para el C_{\max} de instancias 20X20_10.	80
<i>Figura 20.</i> Diagrama de Pareto para tiempo computacional	80

<i>Figura 21.</i> Interacciones del c_{\max} para las instancias 50X5_2.	81
<i>Figura 22.</i> Grafica de efectos principales para el C_{\max} de instancias 50X5_2.....	81
<i>Figura 23.</i> Diagrama de Pareto para tiempo computacional de la instancia 50X5_2.....	82
<i>Figura 24.</i> Interacción para el C_{\max} de la instancia 50X5_4.	83
<i>Figura 25.</i> Efectos principales de la instancia 50X5_4.	83
<i>Figura 26.</i> Diagrama de Pareto para el Tiempo computacional de las instancias 50X5_4	84
<i>Figura 27.</i> Interacción para el C_{\max} de la instancia 50X5_6.....	85
<i>Figura 28.</i> Grafica de efectos principales para las instancias de 50X5_6.	85
<i>Figura 29.</i> Diagrama de Pareto para la instancia 50X5_6.....	85
<i>Figura 30.</i> Interacciones para el C_{\max} de la instancia Ta001_M1	87
<i>Figura 31.</i> Efectos principales de la instancia Ta001_M1	87
<i>Figura 32.</i> Interacciones para el C_{\max} de la instancia Ta003_M4.....	88
<i>Figura 33.</i> Efectos principales de la instancia Ta003_M4.	89
<i>Figura 34.</i> Interacciones para el C_{\max} de la instancia Ta009_M4.....	90
<i>Figura 35.</i> Efectos principales para el C_{\max} de la instancia Ta009_M4.	90
<i>Figura 36.</i> Interacciones para el C_{\max} de la instancia Ta021_M0.....	91
<i>Figura 37.</i> Efectos principales para el C_{\max} de la instancia Ta021_M0.	92
<i>Figura 38.</i> Interacciones de C_{\max} para instancias Ta023_M12.....	93
<i>Figura 39.</i> Efectos principales para el C_{\max} de la instancia Ta023_M12.	93
<i>Figura 40.</i> Interacciones para el C_{\max} de la instancia Ta030_M9.	94
<i>Figura 41.</i> Efectos principales para la instancia Ta030_M9.	94
<i>Figura 42.</i> Interacciones para el C_{\max} de la instancia Ta032_M2.	95
<i>Figura 43.</i> Efectos principales para la instancia Ta032_M2.	96

<i>Figura 44.</i> Interacción del C_{\max} para la instancias Ta034_M0.	97
<i>Figura 45.</i> Efectos principales para la instancia Ta034_M0	97
<i>Figura 46.</i> Interacción del C_{\max} para la instancias Ta036_M6.	98
<i>Figura 47.</i> Efectos principales para la instancia Ta036_M6.	98
<i>Figura 48.</i> Diagrama de Gantt inicial de la HAS-200.	111
<i>Figura 49.</i> Diagrama de Gantt situación actual de la HAS-200.	114

LISTA DE ILUSTRACIONES

	Pág.
<i>Ilustración 1.</i> Diagrama de operaciones de Referencia 2.	27
<i>Ilustración 2.</i> Resumen de un proceso continuo general.	29
<i>Ilustración 3.</i> Flujo continuo en un Flowshop general. Fuente: Baker y Trietsch (p.226)	30
<i>Ilustración 4.</i> Problema general según tipos de configuración.	31
<i>Ilustración 5.</i> Métodos de optimización. Fuente: (Talbi, 1965, p. 18).	35
<i>Ilustración 6.</i> Clasificaciones de técnicas heurística.....	36
<i>Ilustración 7.</i> Representación del Modelo del método de ahorros.....	39
<i>Ilustración 8.</i> Métodos metaheurísticos	43
<i>Ilustración 9.</i> Métodos de solución búsqueda basada en poblaciones evolutivas.....	43
<i>Ilustración 10.</i> Terminología y componentes de un Algoritmos genético.	45
<i>Ilustración 11.</i> Representación de selección natural en la teoría de la evolución.	46
<i>Ilustración 12.</i> Representación del cruce natural en la teoría de la evolución.	47
<i>Ilustración 13.</i> Representación de la Mutación en la teoría de la evolución.	49
<i>Ilustración 14.</i> Esquema general del funcionamiento de un Algoritmo genético.	50
<i>Ilustración 15.</i> Diagrama de Gantt de situación actual del sistema HAS200	58
<i>Ilustración 16.</i> Carta Gantt de la solución inicial del algoritmo de Clarke & Wright.	60
<i>Ilustración 19.</i> Diagrama de Gantt del mejor individuo. (Ejemplo).	63
<i>Ilustración 20.</i> Diagrama de flujo general para generación de la solución inicial.	65
<i>Ilustración 21.</i> Diagrama de flujo general del Algoritmo genético propuesto.	66
<i>Ilustración 22.</i> Makespan promedio por cada instancia AG_UPB Vs. SDST-Ta	99

LISTA DE ANEXOS

	Pág.
<i>Anexo A.</i> Parametros de la HAS-200.	109
<i>Anexo B.</i> Secuenciación de la HAS-200.	109

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO:	Propuesta de un algoritmo evolutivo para el problema de secuenciación de la producción en ambiente de manufactura Flowshop con restricción de tiempos de alistamiento.
AUTOR(ES):	Zaine Johanna Madrid Plata
PROGRAMA:	Facultad de Ingeniería Industrial
DIRECTOR(A):	Orlando Federico González Casallas

RESUMEN

En la presente investigación se desarrolla un algoritmo genético para el problema de programación de la secuenciación de la producción tipo Flowshop con restricciones de tiempos de alistamiento, en la celda de manufactura HAS-200 del laboratorio de automatización de la Universidad Pontificia Bolivariana Seccional Bucaramanga, el proyecto abarca tres fases claves, la primera fase de tipo exploratorio se justifica en la revisión teórica y elementos referentes a la secuenciación tipo Flowshop y métodos de resolución, la segunda fase tiene un alcance transeccional, por la caracterización del sistema de producción que integra la celda, para analizar los tiempos estándar de producción del sistema. Finalmente, se hace una validación de la estructura algorítmica por medio de instancias propuestas en la literatura, para determinar la calidad de respuesta y el rendimiento del programa para encontrar la solución al problema de secuenciación; además se evalúa las fuentes de variación de los parámetros del AG con un diseño experimental factorial y se comprueba que el algoritmo propuesto arroja buenos resultados.

PALABRAS CLAVE:

Algoritmo Genético, Tiempos de Alistamiento, Flowshop, Manufactura, Scheduling.

Vº Bº DIRECTOR DE TRABAJO DE GRADO

GENERAL SUMMARY OF WORK OF GRADE

TITLE: Proposal of an evolutionary algorithm for the problem of sequencing production in the Flowshop manufacturing environment with restriction of setup times.

AUTHOR(S): Zaine Johanna Madrid Plata

FACULTY: Facultad de Ingeniería Industrial

DIRECTOR: Orlando Federico González Casallas

ABSTRACT

In the present investigation a genetic algorithm is developed for the programming problem of the sequencing of the production Flowshop type with restrictions of times of enlistment in the cell of manufacture HAS-200 of the automation laboratory of the Universidad Pontificia Bolivariana Seccional Bucaramanga, the project It covers three key phases, the first phase of exploratory type is justified in the theoretical review and elements related to sequencing type Flowshop and resolution methods, the second phase has a transectional scope, by the characterization of the production system that integrates the cell, to analyze the standard production times of the system. Finally, a validation of the algorithmic structure is done by means of instances proposed in the literature, to determine the quality of response and the performance of the program to find the solution to the problem of sequencing; In addition, the sources of variation of AG parameters are evaluated with a factorial experimental design and it is verified that the proposed algorithm yields good results.

KEYWORDS:

Genetic Algorithm, Setup Times, Flowshop, Manufacturing, Scheduling.

Vº Bº DIRECTOR OF GRADUATE WORK

INTRODUCCIÓN

La investigación de operaciones es una disciplina que se ocupa en la aplicación de métodos analíticos, es una herramienta básica para la resolución de problemas que usualmente involucran la asignación de recursos escasos, dicha resolución se da gracias a las técnicas y modelos matemáticos, basados en abstracciones de la realidad, que tiene por objeto realizar análisis que permitan llegar a conclusiones validas que sustenten cuantitativamente las decisiones que se tomaron frente a la situación problema. Según Hamdy (2004), la IO cada día toma más fuerza entre las organizaciones que ven en este concepto una posibilidad de aventajar sus competidores, por sus múltiples aplicaciones. El avance tecnológico y el desarrollo de la investigación de operaciones motivaron a los especialistas de esta disciplina para reunirse y provocar su rápido crecimiento, mezclado a la computadora con los métodos, dando origen a los algoritmos que brindan soluciones para problemas de programación de la producción.

La programación de la producción requiere de planificación, control y supervisión, con el propósito de mejorar las entregas en las fechas estipuladas, calcular las necesidades en cuanto a recursos y minimizar los costos en la fabricación de un producto; una de las técnicas más efectivas es el control de los tiempos y asignación de operaciones: como el orden de productos que se deben producir o la asignación de tareas por cada máquina que se requiera, Taha (2012) los estudios de complejidad han involucrado modelos heurísticos y metaheurísticos para mejorar los tiempos de producción.

El presente trabajo se enfocará en la aplicación de una técnica metaheurística para dar solución a problemas de programación tipo Flowshop con restricciones de tiempo de alistamiento; donde se estudia y se diseña un algoritmo genético para mejorar los tiempos de

procesamiento de la celda de manufactura HAS-200 que se encuentra en el laboratorio de procesos de la Universidad Pontificia Bolivariana seccional Bucaramanga.

1. PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción del problema

En los sistemas productivos, la definición de tiempos generales de procesamiento es un factor crítico para la programación de máquinas, herramientas y personal; por tanto, se desarrollan enfoques de solución de carácter cuantitativo para lograr la determinación de los tiempos. Un primer acercamiento de los modelos matemáticos hacia la producción es la estructura de optimización lineal para abordar problemas de planeación agregada y secuenciación de la producción. La programación lineal es una técnica de solución conformada por una estructura algebraica definida como una función objetivo, evaluada en una región de factibilidad o restricciones del problema. Según Taha (2012), los estudios de complejidad de los modelos de optimización lineal y no lineal han involucrado modelos metaheurísticos para solucionar problemas de ingeniería; proporcionando soluciones a problemas complejos en tiempo real. Para optimizar una línea de secuenciación existen diferentes estructuras algorítmicas que manifiestan buenos resultados a la hora de su implementación.

En efecto, una secuenciación eficiente es donde los tiempos de preparación y ejecución de una máquina se minimizan logrando el cumplimiento de los requerimientos de la demanda, bajo las condiciones propias del sistema de producción. Por otro lado, los algoritmos genéticos han sido una solución para problemas de alistamiento gracias a su adaptabilidad para el tiempo.

Un sistema de producción depende del ambiente de la planta productiva y un sistema apropiado depende de la configuración o la programación; gracias a la inversión de la maquina HAS-200 de la Universidad Pontificia Bolivariana seccional Bucaramanga se puede hacer una simulación de un ambiente de manufactura la cual permite desarrollar, analizar y tomar

decisiones para el cumplimiento de metas y la búsqueda de la mejora continua que requiere un proceso bajo este enfoque a través de la automatización de procesos industriales.

Dentro de los procesos de automatización industrial, se puede identificar que existe un nivel de incertidumbre en los tiempos de procesamiento debido a los elementos de instrumentación y control, así como los tipos de referencias de productos que se pueden obtener en el sistema. Por tanto, se desea estudiar los tiempos de procesamiento dependientes de la secuencia en el sistema de producción automatizado HAS200 que hace parte del laboratorio de automatización de procesos de la Universidad Pontificia Bolivariana, Seccional Bucaramanga.

1.2. Formulación del problema

Con base a lo anterior, ¿Qué enfoque metaheurístico se deben aplicar al problema de secuenciación de producción bajo ambiente de Manufactura Flowshop considerando restricciones de tiempo de alistamiento en secuencia en el sistema de producción HAS-200 de la Universidad Pontificia Bolivariana?

1.3. Justificación

En la actualidad, ambientes de manufactura no cuenta con un sistema de producción adecuado, ya que, durante su ciclo productivo, las maquinas puedan requerir de cierto tiempo de preparación para iniciar su producción. Los tiempos de estas máquinas dependen de la secuenciación que, en consecuencia, afectan la meta de fechas límites de pedidos. Estas limitaciones en un sistema productivo son consideradas como uno de los más críticos en la gestión y planificación de la producción, pues un proceso con restricciones de tiempos también afecta los costos marginales y la efectividad del sistema.

Un sistema secuencial con tiempos de alistamiento mejorados ayuda a disminuir, tiempos ociosos en la producción, logrando que los procesos sean más efectivos y confiables, ya que en ello también influye el logro de objetivos gerenciales como la capacidad de producción y el control adecuado de inventarios. Con base en el contexto de la problemática de los tiempos de alistamiento en la producción con elementos de automatización, este proyecto está orientado a ofrecer una alternativa de solución basada en una estructura algorítmica que permita un mejoramiento en el sistema de producción HAS-200, de tal forma que permita la programación de producción adecuada, y que sea base para el mejoramiento de las prácticas de laboratorio referentes a la Planeación y Control de la Producción de la Universidad Pontificia Bolivariana, Seccional Bucaramanga.

2. OBJETIVOS

2.1. Objetivo general

Proponer un algoritmo genético que permita el mejoramiento de la programación de producción en un ambiente de producción tipo Flowshop con restricción de tiempos de alistamiento en el sistema de producción HAS-200.

2.2. Objetivos específicos

- Caracterizar la dinámica del sistema de producción HAS-200, de tal forma que se identifiquen las estaciones de trabajo que hacen parte del problema de secuenciación de manufactura.
- Determinar los tiempos de alistamiento y tiempos de procesamiento de las referencias de manufactura en el sistema de producción HAS-200, como punto de partida de la construcción de información para el algoritmo genético.
- Diseñar una estructura metaheurística de algoritmo genético, según el tipo de referencias que se producen, para la definición del programa de producción del sistema HAS-200.
- Validar los parámetros del algoritmo genético propuesto mediante la aplicación de diseño de experimentos según las instancias definidas para el sistema de producción HAS-200.

3. MARCO REFERENCIAL

3.1. Antecedentes

3.1.1. Sistemas de manufactura.

Los sistemas de manufactura se conocen como procesos productivos que integran recursos con el fin de obtener un producto final de calidad basado en el mejoramiento de los procesos y los insumos usados; (Grover, 2007) los procesos que se llevan a cabo en un sistema manufacturero involucran una combinación de maquinaria, herramientas, energía y recurso humano. (p, 12). Los sistemas de producción dependen del tipo de empresa ya que al realizar sus operaciones requiere adoptar una estructura física de operación adecuada para garantizar la eficiencia y eficacia del proceso, es por ello, que existen diferentes configuraciones productivas que varían dependiendo a sus características:

- *Flujo continuo o taller de flujo*, la producción sigue una línea de operación y los productos no sufren una modificación durante un tiempo, donde el volumen de productos es alto.
- *Configuración por lote o batch*, está organizada para que diferentes líneas de producción produzcan diferentes productos a gran escala.
- *Configuración Job shop*, se encarga de sacar producciones por referidos y el flujo de producción es relativamente bajo.
- *Línea de ensamble*, las estaciones de trabajo se mueven de acuerdo con la producción, normalmente su operación es más compleja y requiere de mucha tecnificación.

3.1.2. Descripción del sistema HAS-200.

La Universidad Pontificia Bolivariana Seccional Bucaramanga realizó la inversión económica de la máquina HAS200 en el año 2017, para afianzar conceptos relacionados a la automatización, usada para prácticas de laboratorio de ingeniería industrial, ingeniería electrónica y la especialización de control e instrumentación industrial, en la tabla 1 se presenta el software que suministra el sistema HAS-200. (Ver Tabla 1)

Tabla 1. Software del sistema HAS-200

ELEMENTO	DESCRIPCIÓN
3DSupra ver2.0	Software diseñado específicamente para controlar y supervisor el funcionamiento de los módulos HAS-200 de forma remota.
EdMES Ver4.0	Software para cubrir el 4° nivel de la pirámide de automatización, un (MES) Manufacturing Execution System adaptado para realizar prácticas.
KepServerEX Ver5.17	OPC Server genérico, se trata de una versión demo que funciona durante dos horas. Se puede descargar de forma gratuita de la siguiente dirección: http://www.kepware.com
MySQL Server Ver5.6.20.0	Se requiere SQL Server para mantener la base de datos de la información necesaria para llevar a cabo la gestión integrada de la producción.
ACT Controller Ver1.20 o superior	Software SMC para el mantenimiento de los Servodrive- LEC.
Crew Editor Ver1.05 o superior	Software ESAware para el mantenimiento local y remoto de HMI

Fuente: Manual de usuario y practicas HAS-200. (p.20)

La máquina tiene dos paneles de control, situada en la parte frontal donde el operario da la orden de producción. Al dar la orden de producción, la maquina se ejecuta por medio de cilindros neumáticos, contiene una báscula para efectuar control del peso de cada referencia donde la salida de datos es por medio de un PCL para la visualización del usuario; durante el llenado del dosificador la báscula tiene su propio pulsador de encendido/apagado, es decir si la estación esta activada la báscula también debe encenderse, por otro lado, el lector de código de

barras o BCR permite identificar el recipiente que está transportando por la cinta, donde la información es transmitida al PLC por la línea de serie RS-232.

Ademas posee dos semaforos, donde se indica la situacion de produccion en cada maquina, y en el area de control, posee una cinta transportadora integrado por un motor corriente, que es donde se encuentran los sensores para la identificacion del producto.

En el area de la plataforma la cual se divide en dos carriles, se almacenan los botes de diferentes referencias, la selección del producto se logra a traves de la electrovalvula de un cilindro neumatico, los botes se depositan al comienzo del carril y gracias al mecanismo formado por un cilindro neumatico y un levantamiento los productos se deslizan.

3.1.3. Descripción del proceso.

La HAS-200 consta de dos estaciones preparadas para trabajar de manera independiente (producción, y control), estas estaciones ofrecen al usuario una visión general de un sistema de manufactura. El sistema puede simular la fabricación de tres productos diferentes, compuesto por un recipiente y unas esferas como su materia prima, cada recipiente tiene una etiqueta con un código de barras para identificar al producto durante el proceso; las referencias de los productos dependen del peso de cada una: 15 gramos para la referencia 1, 30 gramos para la referencia 2 y 45 gramos para la referencia 3. (Ver Figura 1)



Figura 1. Referencias de la HAS200. De izquierda a derecha (15-30-45 gr)

En el diagrama de operaciones se puede apreciar las actividades que realiza el sistema; se toma como ejemplo la referencia No 2 para eslabonar el tiempo de procesamiento de cada tarea con un tiempo total de 107,8572 segundos. (Ver Ilustración 1).

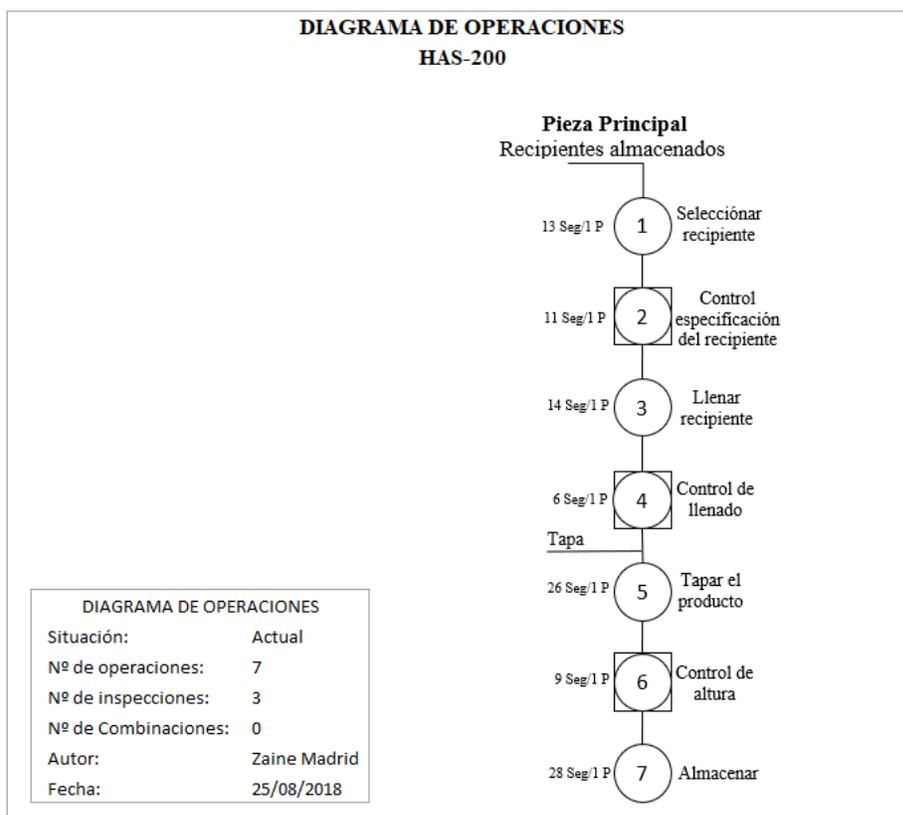


Ilustración 1. Diagrama de operaciones de Referencia 2.

3.1.4. Estaciones de trabajo de la HAS-200.

3.1.4.1. Estación de producción.

Se encarga de proporcionar al sistema recipientes vacíos almacenados en cuatro compartimentos contiguos con una capacidad de 36 botes, para extraer el recipiente se da la orden al sistema.

Al momento de ser transportado, se lleva a la zona de control para identificar la etiqueta del recipiente, tal cual como un sistema de reparación y averías. Después el recipiente es transportado a la báscula para depositar las esferas por medio de un dosificador, en este instante se determina el peso de acuerdo con el pedido requerido por el operario.

3.1.4.2. Estación de control y almacenamiento.

Con el recipiente llenado, se transporta a la zona de control para identificar el tipo de producto, gracias a este sistema se pueden hacer estudio de cuellos de botella; si el recipiente cumple con el peso correspondiente, pasa a la siguiente tarea, donde el operario se encarga de tapar el producto. En la última etapa del proceso, si el recipiente cumple con los requerimientos de altura es transportado por un brazo robótico a un buffer que es donde se almacenaran los productos.

3.2. Marco teórico

A continuación, se exponen los conceptos teóricos relacionados con el tema objeto de estudio del presente trabajo siguiendo un enfoque que va de lo general a lo particular. Inicialmente se hace una corta revisión de la literatura con la que se pretende contextualizar al lector acerca del problema de la programación en un Flowshop.

3.2.1. Programación de operaciones.

A mediados del siglo XX, la programación de operaciones ha sido un factor clave en la competitividad de los mercados, ya que, en muchas organizaciones, tener un amplio portafolio de productos, depende de la capacidad y de las exigencias de la producción para cumplir con las necesidades de los clientes.

La complejidad de optimizar, mejorar o encontrar soluciones para tener un buen funcionamiento en un sistema de producción ha sido objeto de estudio, en muchas investigaciones; la programación de operaciones se entiende como, la planificación y el proceso de establecer actividades, Morton y Pentico (1993) afirman:

“Programar es el proceso de organizar, elegir y dar tiempos al uso de recursos para llevar a cabo todas las actividades necesarias, para producir las salidas deseadas en los tiempos deseados, satisfaciendo a la vez el número de restricciones de tiempo y relaciones entre actividades y los recursos”.

3.2.2. Taller de producción continua.

Los trabajos son las actividades que se desarrollan en un sistema de producción, estos trabajos pueden ser realizados por operarios y/o máquinas. En un taller de producción continua, cada trabajo debe procesarse en una serie de máquinas exactamente una vez, por la secuencia de los trabajos que realiza. Los objetivos más comunes en estos problemas es minimizar el tiempo de flujo de flujo total, la tardanza máxima o el número de trabajos retrasados.

La mayoría de los problemas de programación suponen certidumbre en los datos; los tiempos de preparación son independientes del orden de procesado. (Sippper, Bulfin, 1998, p.403). (Ver Ilustración 2).

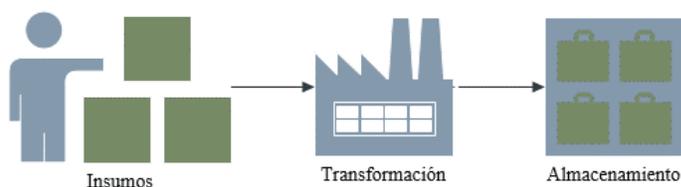


Ilustración 2. Resumen de un proceso continuo general.

3.3. Problemas de configuración de talleres de flujo continuo.

3.3.1. Flowshop.

El problema de producción Flowshop (PFSP, por sus siglas en inglés) desempeña un papel importante en la teoría de la programación. La versión determinista fue introducida en la literatura por Johnson (1954), se identifica como el primer estudio formal de un problema en la teoría de la programación. El PFSP es un modelo basado en el diseño de las máquinas organizadas en serie. En el 2006 Baker y Trietsch define el PFSP como:

“Cada trabajo i consiste en m operaciones O_{ij} con tiempos de procesamiento p_{ij} ($j = 1, \dots, m$) donde O_{ij} debe procesarse en la máquina m_j y hay restricciones de precedencia de la forma $O_{ij} \rightarrow O_{i,j+1}$ ($i = 1, \dots, m-1$) para cada $i = 1, \dots, n$, es decir. Cada trabajo se procesa primero en la máquina 1, luego en la máquina 2, luego en la máquina 3, etc”. (p.225).

Para tener un proceso productivo eficiente y controlado en las operaciones, es necesario encontrar la programación adecuada que permita el mejor flujo de los productos, dependiendo de los tiempos de procesamiento de las máquinas. En la siguiente ilustración se puede describir un taller de flujo continuo con un Flowshop general. (Ver Ilustración 3)

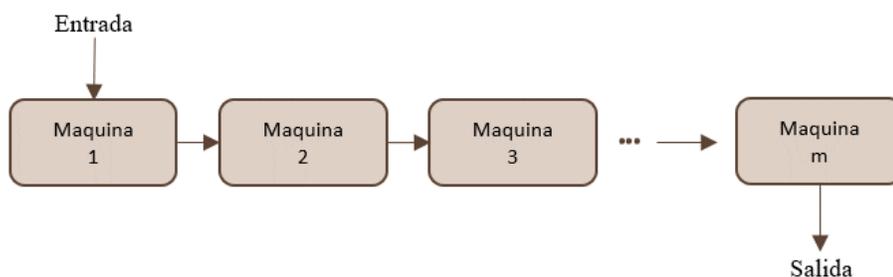


Ilustración 3. Flujo continuo en un Flowshop general. Fuente: Baker y Trietsch (p.226)

Dentro del proceso de producción con una secuenciación lineal se puede encontrar diferentes problemas de programación ya que, muchos trabajos pueden requerir menos m operaciones, y en otros casos no requerir siempre máquinas adyacentes y las operaciones iniciales y finales no siempre ocurren en las máquinas 1 y m . (Ver Ilustración 4).

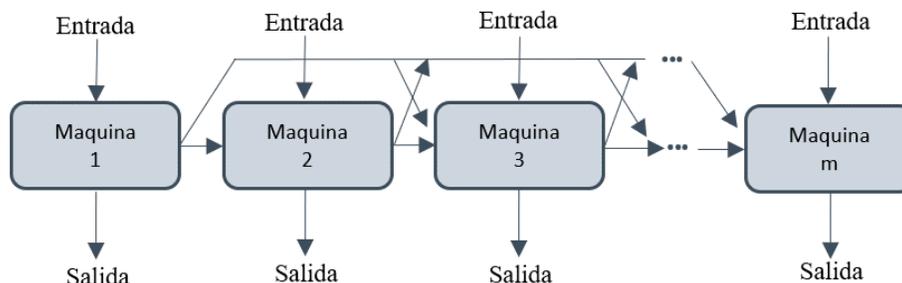


Ilustración 4. Problema general según tipos de configuración. Fuente: Baker y Trietsch (p.226)

En la siguiente tabla se presentan los tipos de configuración que presenta un taller de flujo lineal. (Ver Tabla 2).

Tabla 2. Tipos de problemas Flowshop

PROBLEMA	DESCRIPCIÓN
Horarios de permutación	Los tiempos de programación en cada máquina son diferentes
El problema de las dos máquinas	Los tiempos de configuración para las operaciones son dependientes de la secuencia e incluidos en los tiempos de procesamiento.
El problema de las tres máquinas	El problema general de 3 máquinas se conoce como NP-hard, ya que también tiene horarios de permutación según sea el tipo de producción.
Variaciones del modelo m-Machine	Depende de la cantidad de máquinas en un flujo continuo y las restricciones anteriormente mencionadas.

3.3.2. Flowshop con doble máquina.

Los modelos lineales con dos máquinas presentan diferentes restricciones de acuerdo con los tiempos programados para la producción. Baker y Trietsch, (2006):

- *Modelo con retraso de tiempo*, existe un procesamiento en la segunda máquina, pero la primera máquina tiene una demora; una aplicación típica es cuando cada estación consta de un lote de varias unidades discretas e idénticas. Una vez la primera unidad se completa en la maquina 1, puede comenzar a procesar en la siguiente Maquina 2.
- *El modelo con configuraciones*, para este modelo los tiempos de configuración son independientes de la secuencia y de los tiempos de procesamiento, es decir este modelo básico debe completarse en la primera operación de un trabajo antes de que pueda comenzar la configuración de la maquina 2 e incluso esta característica es conocida como *configuración adjunta*.

La HAS-200 adopta el modelo de retraso de tiempo ya que posee dos estaciones que se configuran de manera independiente (estación de producción y estación de control-Almacenaje), donde el tiempo de alistamiento es dependiente en cada una.

3.3.2.1. Modelo con retraso de tiempo.

Con la condición del modelo básico de dos máquinas, donde *los trabajos deben procesarse en la segunda maquina dentro de un periodo de tiempo, después de haberse completado en la primera máquina*, estos tiempos son dependientes de la secuencia en la segunda máquina, para dar inicio a un nuevo lanzamiento. Los tiempos de la secuencia requieren minimizar el tiempo de procesamiento de todas las tareas, también conocido como makespan $\{C_{\max}\}$. En la figura 2 se ilustra la estación de entrada de los productos de la HAS- 200. (Ver figura 2).

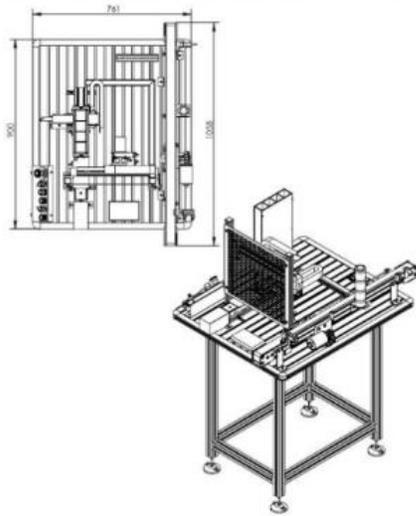


Figura 2. Estación de producción. Fuente: Manual de usuario HAS-200. (p. 7)

3.3.3. Tiempo de procesamiento.

Es el tiempo total de proceso, en que tarda un proceso productivo en finalizar su trabajo.

3.3.4. Tiempos de alistamiento.

Los sistemas industriales tienden adoptar configuraciones lineales con tiempos de alistamiento o de preparación, que se definen como lapsos de tiempo en que la máquina toma en iniciar la operación o en otros casos es el tiempo de cambiar de un producto a otro. El tiempo de alistamiento en dos estaciones puede afectar el tiempo de procesamiento en un ambiente de manufactura, supongamos que hay dos automóviles que requieren ser lavados, ingresa un auto color gris lleno de barro y el otro es blanco que solo se necesita una lavada parcial, para facilitar el lavado es necesario atender el primer al auto.

3.3.5. Minimización del Makespan.

El makespan se puede describir como el lapso del tiempo en el que se procesa la totalidad de los trabajos. “El objetivo de minimizar el makespan, C_{\max} , también denominado M , está relacionado con el rendimiento del programa, porque se define como la cantidad de trabajo completado por unidad de tiempo”. (Baker y Trietsch, 2006, p.169). “Matemáticamente, el problema consiste en encontrar una permutación de las tareas que resulte en un tiempo mínimo de procesamiento de todas ellas”. (Mercado y Bard, 2001, p.2)

3.3.6. Soluciones algorítmicas

En 1966, Ronald L. Graham introdujo formalmente los algoritmos, con el propósito de generar soluciones próximas a problemas de optimización que no eran eficientes por técnicas computacionales existentes, con el paso del tiempo la introducción de nuevas técnicas para resolver problemas de programación lineal fortaleció la metodología y la capacidad de respuesta frente a problemas Np. (Teofilo, 2007, p.6). Sippper y Bulfin (año) afirman:

El conjunto Np es el conjunto de todos los problemas que se pueden resolver por enumeración total. El conjunto P es un subconjunto de NP que consiste en todos los problemas para los que se conocen algoritmos eficientes. El conjunto de NP-Hard también es un subconjunto de NP, pero se ha comprobado que estos son los problemas más complejos de resolver. (p.400).

La complejidad algorítmica, depende del problema al que se trata de dar respuesta; existen soluciones con capacidad de dar soluciones próximas y eficientes para problemas NP-Hard, estas soluciones permiten modificar los NP en tiempo polinomial. Por tanto, Sippper (1998) define a un algoritmo eficiente como “Aquel, en el que el esfuerzo dedicado a un problema está acotado por un polinomio cuyo grado es el tamaño del problema, como el número de trabajos”. Existen

diferentes modelos de optimización, los modelos se ajustan a la robustez, incertidumbre y periodicidad, para resolver estos problemas. (Ver Ilustración 5)

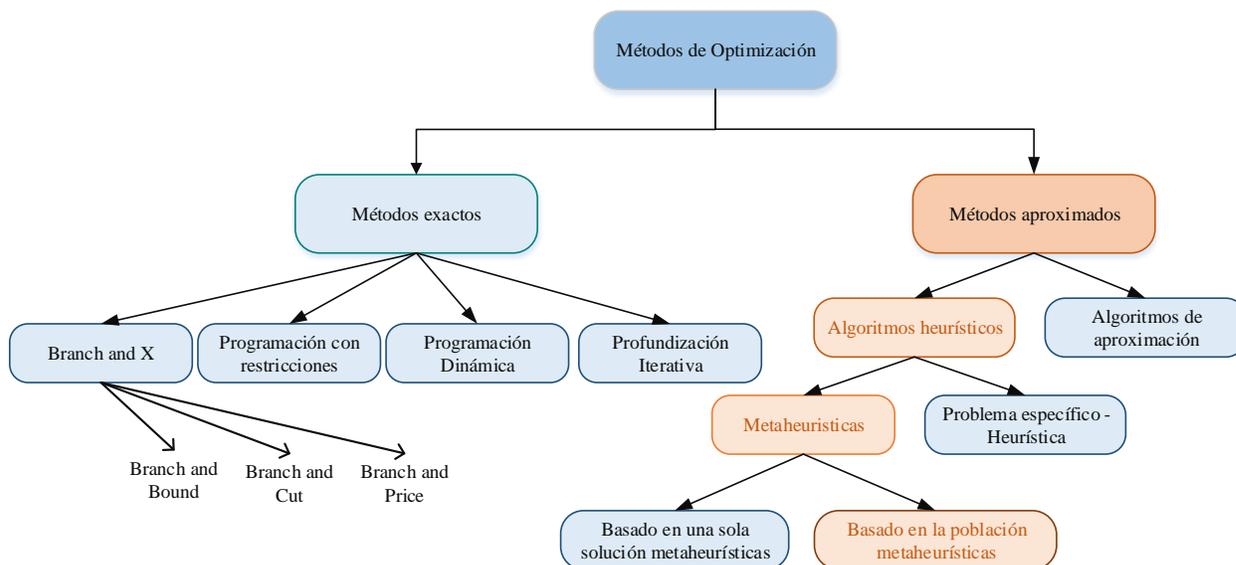


Ilustración 5. Métodos de optimización. Fuente: (Talbi, 1965, p. 18).

3.3.7. Métodos exactos.

Proporcionan soluciones analíticas o matemáticas, que asegura la convergencia de soluciones óptimas, la desventaja de los métodos exactos se debe a falta de adaptabilidad de acuerdo con el tipo de problema que se presenta. “Esta problemática suele ser ocasionada por las características inherentes de un problema, ya que éstas pueden impedir el uso de ciertos métodos exactos y crear la necesidad de elaborar otros más apropiados”. (Morillo, Moreno y Díaz, 2013, p, 254).

3.3.8. Métodos Heurísticos

Brindan soluciones a grandes problemas con esfuerzo computacional, donde se espera que su capacidad de respuesta sea óptima o cercanas a la óptima. “El inconveniente de los enfoques

heurísticos, por supuesto, es que no garantizan la optimalidad; y en algunos casos, incluso puede ser difícil juzgar su efectividad”. (Baker y Trietsch, 2009, p.189).

Dentro de los algoritmos heurísticos, se encuentran los métodos metaheurísticos, estos modelos imitan fenómenos simples de la naturaleza que se han adaptado a la inteligencia artificial, brindando soluciones a problemas complejos de la optimización. En la presente ilustración se hace un resumen de técnicas tradicionales en los métodos heurísticos. (Ver Ilustración 6)

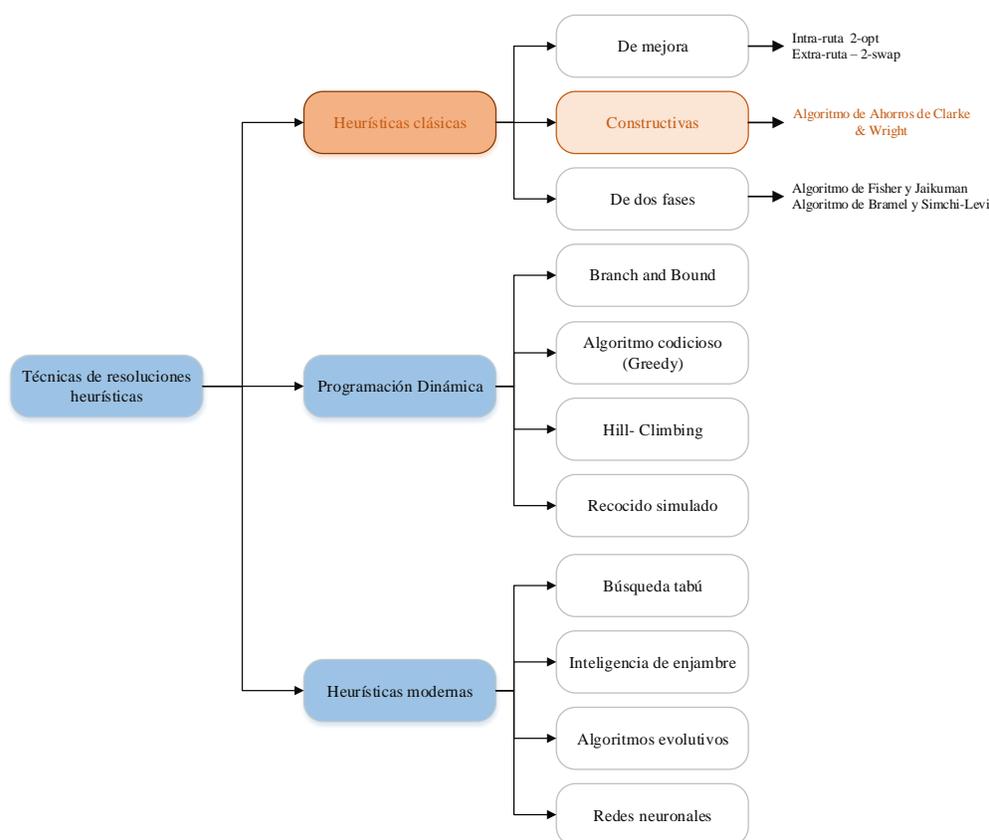


Ilustración 6. Clasificaciones de técnicas heurística.

Existen incontables técnicas y combinaciones para la resolución de grandes problemas en la optimización, lo cual se hace muy complejo definir todos los métodos existentes, además se ha comprobado métodos exactos para resolver un problema de optimización, pero su

implementación computacional se vuelve muy costosa. Además, una de las ventajas de los modelos heurísticos es que permite condicionarse a la problemática que se requiere resolver, obteniendo nuevos modelos o condiciones para tomar una decisión.

- a) **Heurísticas constructivas:** “Las heurísticas constructivas construyen la solución paso a paso, de manera que se añaden elementos a la solución a medida que tienen en cuenta los diferentes criterios”.
- b) **Procesos iterativos:** “Los métodos que se basan en procedimientos iterativos parten de una solución, modificándola y obteniendo soluciones parecidas que pueden ser mejores”.

3.3.9. Algoritmo de Clarke and Wright

En 1964 fue introducido por Clarke y Wright para resolver problemas de VRP (Problemas de rutas de vehículos, por sus siglas en inglés), es un algoritmo que también es conocido como el método de ahorros, consiste en realizar una exploración limitada del espacio de búsqueda y dar una solución de calidad aceptable en un tiempo moderado. (Olivera, 2004). Pérez (2016) afirma:

“Se cuenta con un depósito central y un número de vehículos no limitado para atender la demanda conocida de un número de clientes, siendo el objetivo el de encontrar las rutas que deben realizar dichos vehículos de manera que los costes sean mínimos y se satisfaga la demanda”.

3.3.9.1. Descripción del modelo de Clarke y Wright.

Estos métodos dependen de la naturaleza de la red; el modelo de Clark y Wright son un ejemplo de un modelo constructivo. Patro, Suero y Guzmán (2015) afirma:

Inicialmente cada cliente es visitado por un vehículo, por lo tanto, existirán tantos vehículos como clientes. Luego, de acuerdo a la capacidad del vehículo, se determina si este puede visitar un segundo nodo cercano al primero, si esto es posible, entonces se constituye un ahorro de un viaje. (p.14).

Estas técnicas por lo general permiten minimizar la distancia, donde se obtienen resultados satisfactorios y por lo general suelen implementarse con algoritmos de búsqueda para mejorar su capacidad de respuesta. Sin embargo, no es suficiente implementando el método de Clark y Wright, ya que este, solo es el procedimiento analítico de secuenciación inicial de producción de la HAS-200. Como bien sabemos una de las ventajas más significativas de los algoritmos es que permiten acomodarse, de acuerdo con al objeto de estudio.

3.3.9.2. Metodología del algoritmo.

El pivote o nodo central se denota como O y la cantidad de clientes están de 1 hasta n, donde las localidades están representadas por i,j; el costo del trayecto del nodo central a cada cliente (d_{oj}) y los costes de desplazamiento entre cada pareja de clientes (S_{ij}). Luan (2008) afirma que:

Si el ahorro es favorable, se mantiene la ruta unida generado a su vez una sub-ruta más larga, en el caso contrario, termina la asignación de dicha ruta. Sobre este mismo mecanismo se itera a continuación, realizando uniones en todos los puntos posibles. (p.36).

El ahorro en distancias i y j se denota en la ecuación (1), el costo de viajar entre los nodos i y j se demuestran en las ecuaciones (2) y (3); en la siguiente ilustración se plantea un modelo general, de acuerdo con las posibles uniones. (Ver Ilustración 7)

$$S_{ij} = d_{io} + d_{oj} - d_{ij} \quad (1)$$

$$\text{----- } d_{oj} + d_{io} + d_{jo} + d_{oj} \quad (2)$$

$$\text{--- } d_{ij} + d_{oj} + d_{io} \quad (3)$$

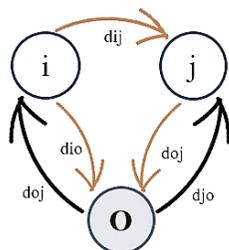


Ilustración 7. Representación del Modelo del método de ahorros

Según la metodología de este modelo se considera que la ecuación (3) es el mejor escenario para la representación del ahorro.

3.3.10. Formulación matemática

De acuerdo con Clarke y Wright (1964), el problema puede formularse matemáticamente mediante la programación lineal entera, el objetivo principal del modelo es la minimización del coste del recorrido de los vehículos:

$$\text{Minimizar: } \sum_{(i,j) \in E} C_{ij} X_{ij} \quad (4)$$

Sujeto a:

$$\sum_{j \in \Delta^+(i)} X_{ij} = 1 \quad \sum_{j \in \Delta^-(i)} X_{ij} = 1 \quad \forall i \in V \setminus \{0\} \quad (5)$$

$$\sum_{(i,j) \in E} X_{ij} \leq |U| - 1 \quad \forall U \in V \quad 2 \leq |U| \leq n - 2 \quad (6)$$

$$X_{ij} \in \{0,1\} \quad (7)$$

(4) → Es la función objetivo del costo a minimizar

(5) → Restricciones que limita la llegada y partida de cada uno, es decir cada vendedor debe entrar y salir una sola vez en cada localidad.

(6) → Condición para evitar la aparición de una sub-ruta

(7) → Variable de decisión.

La base metodológica de la heurística se define de la siguiente manera:

- a) Se determina la distancia más corta, entre cada localidad.
- b) Se crea una asignación inicial para cada destino.
- c) Se calcula el ahorro para cada unión de clientes
- d) Se identifica la celda que contenga el ahorro máximo en la matriz de ahorro neto.

Para comprender el funcionamiento del algoritmo se procede con el siguiente ejemplo; supongamos que en una organización espera construir una ruta de transporte teniendo en cuenta la capacidad del camión. El vehículo tiene una capacidad de 35.000 unidades, se sabe que la demanda requerida por cada cliente se representa en la siguiente tabla. (Ver Tabla 3)

Tabla 3. Demanda requerida por clientes (Ejemplo)

Cliente	1	2	3
Cantidad/Unidades	10.000	7.500	12.000

Para hallar la ruta que favorezca a la empresa, es conveniente conocer las distancias de todos los destinos. (Ver Tabla 4)

Tabla 4. Tabla de distancias de localidades

	0	1	2	3
0	0	32.01	18.02	15.81
1	--	0	14.14	42.72
2	--	--	0	30.41
3	--	--	--	0

El cálculo del ahorro, se denota la ecuación (1), donde $S_{21}=d_{o2} + d_{o1} - d_{21} \rightarrow 18.02 + 32.61 - 14.14 = 35.89$, donde el mayor ahorro proviene de la ruta **2-1-3** teniendo en cuenta la capacidad del vehículo con una demanda acumulada de 29.500 unidades.

3.3.11. Pseudocódigo del Algoritmo de Clarke and Wright

A continuación, la representación de la codificación de un Algoritmo de Clarke and Wright. (Ver Tabla 5).

Tabla 5. Pseudocódigo Clarke and Wright

ALGORITMO: CLARKE AND WRIGHT	
1	Definir ListaTrabajos Heurística del Mejor Vecino
2	While (ListaTrabajos no vacía) do
3	Asignar primer trabajo de ListaTrabajos como primer trabajo de Secuencia
4	Definir ListaSecuencia (todos los trabajos salvo el primer trabajo de la secuencia)
5	while (ListaSecuencia no vacía) do
6	Asignar trabajo de ListaSecuencia que genere menor setup a continuación.
7	Eliminar trabajo asignado de ListaSecuencia
8	endwhile
9	Evaluar Secuencia.
10	Eliminar primer trabajo de ListaTrabajo.
11	endwhile
10	End procedure

Fuente: (Salazar, Medina, 2012).

3.3.12. Soluciones Metaheurística.

En los últimos 20 años estas soluciones algorítmicas han recibido gran popularidad, pues gracias a la efectividad de resolver problemas de gran tamaño permite la eficiencia de su solución y dificultad. (Talbi, 1965, p.23). Por otra parte, Frank et al en el (2012) refiere a las

metaheurística como modelos de solución particularmente eficientes ya que las técnicas de optimización combinatoria surgen como alternativas de gran interés, pues tienen la capacidad de encontrar soluciones de muy buena calidad, en tiempos de cómputo razonables. El enfoque metaheurístico, añade un elemento de aleatoriedad. Taha H. (2012) afirma que:

La metaheurística está diseñada principalmente para escapar del entrapamiento en el óptimo local al permitir movimientos inferiores, si es necesario. Se espera que la flexibilidad agregada a la búsqueda conduzca a una mejor solución. A diferencia de la heurística, la cual siempre termina cuando se llega a un óptimo local. (p. 358).

3.3.12.1. Métodos metaheurísticos.

Los procedimientos metaheurísticos constituyen a la nueva generación de los algoritmos heurísticos, por tal motivo su rendimiento y capacidad de respuesta mejoran notablemente; estos métodos han evolucionado a través de los años y sus aplicaciones demuestran eficiencia a problemas complejos de resolver. Brito et al. (2004) afirman. “Las metaheurística son estrategias para diseñar procedimientos heurísticos. Por tanto, los tipos de metaheurística se establecen, en primer lugar, en función del tipo de procedimiento a los que hace referencia”. (p.6). (Ver Ilustración 8)

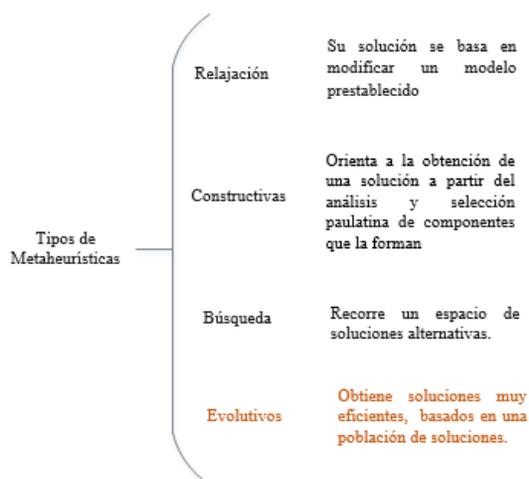


Ilustración 8. Métodos metaheurísticos

3.3.12.2. Métodos evolutivos.

Esta técnica se destaca por la capacidad de combinar la información que proporciona una población, con el propósito de hacerla evolucionar y obtener mejores soluciones; Gracias a este modelo “las iteraciones del algoritmo no se tiene una única solución sino un conjunto de estas”. (Alba, Laguna, Marti, 2003, p.2). En la Ilustración 9, se clasifican algunos de los modelos evolutivos. (Ver Ilustración 9).

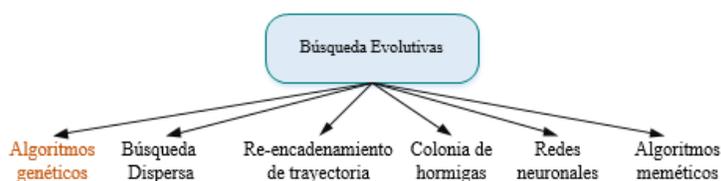


Ilustración 9. Métodos de solución búsqueda basada en poblaciones evolutivas

3.3.13. Algoritmo genético.

Los algoritmos genéticos (AG, por sus siglas) se definen como, “Una técnica de búsqueda y optimización basados en los principios de la genética y selección natural”. (Seck, Medina y

Hernández, 2016, pág. 4). Estos algoritmos realizan una búsqueda factible sin importar que el sistema tenga funciones discontinuas.

Los principios de la genética se definen por la teoría de la evolución introducida por Charles Darwin a mediados de siglo XIX. “Darwin observó que aquellos individuos con variaciones al entorno tienen una posibilidad mayor de sobrevivir en la siguiente generación”. (Hillier y Lieberman, 2015, p.585). Esta idea se acomoda a los problemas de optimización porque los cromosomas que buscan sobrevivir tienen genomas, estos genomas están llenos de información tales como sus características físicas y mentales, es decir que la especie hereda ciertos atributos y la descendencia será la versión mejorada, se demostró que el algoritmo permite adaptarse a esta teoría porque, se tiene una población de datos y la información de cada dato está compuesta por bits, al combinar cada dato se generará uno nuevo y de esta manera se buscará el mejor elemento, después se harán cruce de los datos más eficaces y así se llegaría a la mejor respuesta.

3.3.14. Teoría de Holland.

En 1970 por primera vez se hizo un estudio y descripción del algoritmo por John Holland donde explica que, “Se pueden encontrar soluciones aproximadas a problemas de gran complejidad computacional mediante un proceso de ‘evolución simulada’”; según Holland estos algoritmos presentan ciertas características:

- i). Es un proceso estocástico, los procesos de selección y reproducción son aleatorios y juega un papel importante ya que en la teoría de la evolución cuando se hereda los genes de un individuo, estas también son aleatorias.
- ii). El algoritmo puede *recombinarse* y alcanzar diferentes respuestas donde es posible encontrar la mejor respuesta al problema.

- iii). *La robustez* refiere a la habilidad de ejecutarse de forma consistente a una amplia red de problemas, es decir, no es necesario tener un requerimiento particular para adaptarlo, ya que puede resolver diferentes problemáticas, es por ello que en la decada de los 80's estos Algoritmos con propósitos de Machine-Learning (disciplina en la Inteligencia Artificial) fue de gran impacto, incluso lo aplicaron a videojuegos, y de tal forma para problemas de optimización se ha demostrado que es una herramienta potente.
- iv). *La limitación* que presenta estos algoritmos es que no garantizan el óptimo local, ya que su función es mejorar y llegar a respuestas aceptables y racionales.

3.3.14.1. Terminología los Algoritmos genéticos.

Los elementos distintivos de los Algoritmos Genéticos son los **individuos** o soluciones únicas al problema bajo análisis, las **generaciones** son la cantidad de veces que se generan nuevos individuos. Los **cromosomas** son la información genética cada cromosoma se divide por genes, estos genes son el factor de control único y el alelo es la información que contiene cada gen. (Perea, 2015, p.32). En la Ilustración se puede comprender la terminología y los elementos que componen un AG. (Ver Ilustración 10)

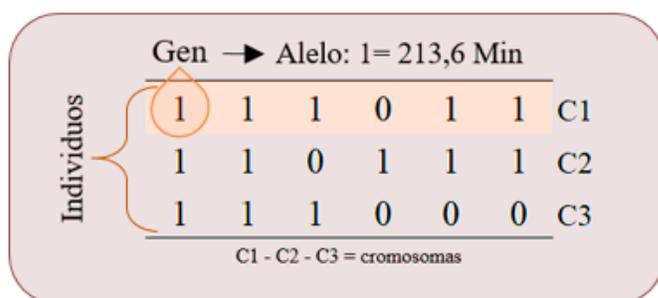


Ilustración 10. Terminología y componentes de un Algoritmos genético.

En la terminología también se encuentra la función de aptitud, donde los individuos son evaluados y comparados.

3.3.15. Operadores de los Algoritmos genéticos.

3.3.15.1. Selección.

A partir de una población, se debe evaluar para escoger a los padres, normalmente esta denominada a una función objetivo para cumplir con un criterio y obtener una probabilidad de ser **elegido**. (Ver Ilustración 11)

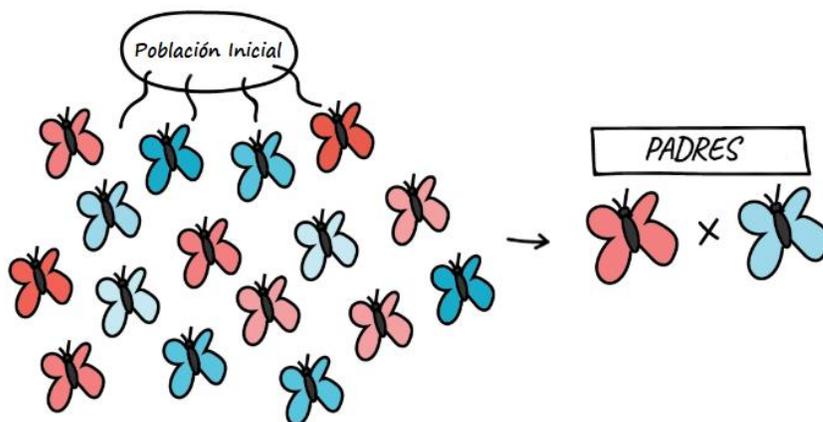


Ilustración 11. Representación de selección natural en la teoría de la evolución. Fuente: (Khan Academy, 2018).

Sin embargo, el proceso de selección tiene otros métodos tales como:

- a) **Basado en el rango:** La población se define por orden de adaptación, y los X menos dignos son eliminados y sustituidos por la descendencia de los X mejores o más aptos, un ejemplo de este método es el muestreo estocástico.

- b) **Rueda de ruleta:** Se escogen genes aleatorios de los cromosomas de la generación actual de la población inicial y se emparejan, sin importar que sean del mismo progenitor.
- c) **Selección de torneo:** se escogen aleatoriamente un número e individuos de la población, y el que tiene puntuación mayor se reproduce, sustituyendo su descendencia al que tiene menor puntuación.

3.3.15.2. Cruzamiento.

Es la parte más crucial del proceso, ya que en este apartado nace un nuevo espécimen con las mejores características de sus descendientes. Es recomendable que la recombinación se enfoque en el cromosoma de mejor puntuación. El intercambio genético de estas permutaciones mejora la solución de la problemática. (Ver Ilustración 12)

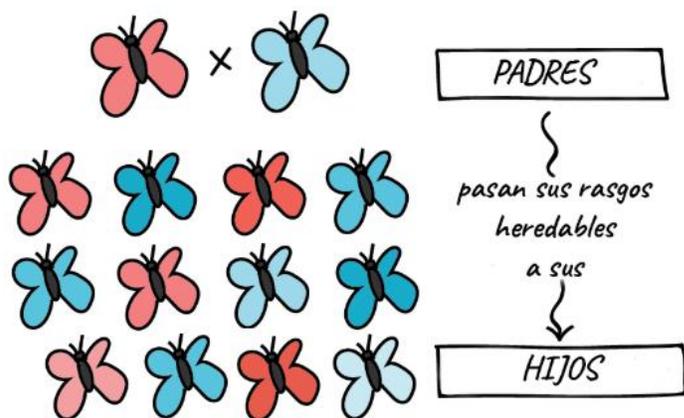


Ilustración 12. Representación del cruce natural en la teoría de la evolución. Fuente: (Khan Academy,2018).

Hay dos formas muy comunes de generar la descendencia en la codificación del AG. Alberte, Málaga e Ibañez (2004), existen puntos de cruce por n-puntos, uniforme y aritméticos.

- a) **Cruce n-puntos:** Los padres se cortan por n puntos y el material genético se intercambia entre ellos, los más usados son de un punto o de dos puntos.
- b) **Cruce uniforme:** Patrón aleatorio en los cromosomas para intercambiar la probabilidad de los genes.
- c) **Cruces especializados:** Aplicar cruces aleatorios codifican soluciones inválidas o poco eficientes. Un ejemplo de estos operadores de cruce es usando en el problema del viajante, en este caso es una limitación del algoritmo ya que es poco recomendable usar este tipo de cruzamiento.

El número de cruces en el AG determina la calidad de la solución a la problemática, pues si la rata de cruces es muy alta puede ocasionar pérdidas en tiempo computacional ya que explora regiones no prometedoras del espacio de búsqueda o si es muy pequeña, ocurre la posibilidad de quedar atrapado en un **óptimo local** y no represente mejoras en la solución.

3.3.15.3. Mutación.

El individuo con las nuevas características tiene un porcentaje de probabilidad de producir cambios, ya sean incrementales o descendientes. Es un suceso inesperado por tal motivo, se recomienda que en el algoritmo la frecuencia de esta operación sea baja, se espera que este en un rango $\{0.05 - 0.1\}$. (Ver Ilustración 13).

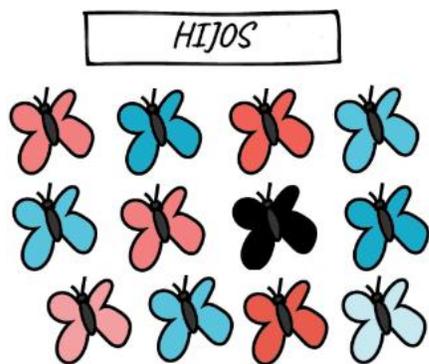


Ilustración 13. Representación de la Mutación en la teoría de la evolución.

3.3.16. Pseudocódigo del Algoritmo Genético.

En la tabla 6, se describe la representación de la codificación de un Algoritmo genético sencillo. (Ver Tabla 6).

Tabla 6. Pseudocódigo Algoritmo Genético.

ALGORITMO: GENETIC ALGORITHM	
1	procedure Algoritmo Genético
2	$t \leftarrow 0$
3	inicializar P_t
4	evaluar P_t
5	while ($t < Ng$) do
6	$t \leftarrow t+1$ seleccionar padres de P_{t-1}
7	formar población P_t
8	evaluar P_t
9	endwhile
10	End procedure

Fuente: (Michalewicz, 1999).

Para comprender la adaptabilidad del algoritmo genético y una vez analizado los procesos de cada componente se desarrolla un diagrama con los componentes de un Algoritmo Genético.

(Ver Ilustración 14)

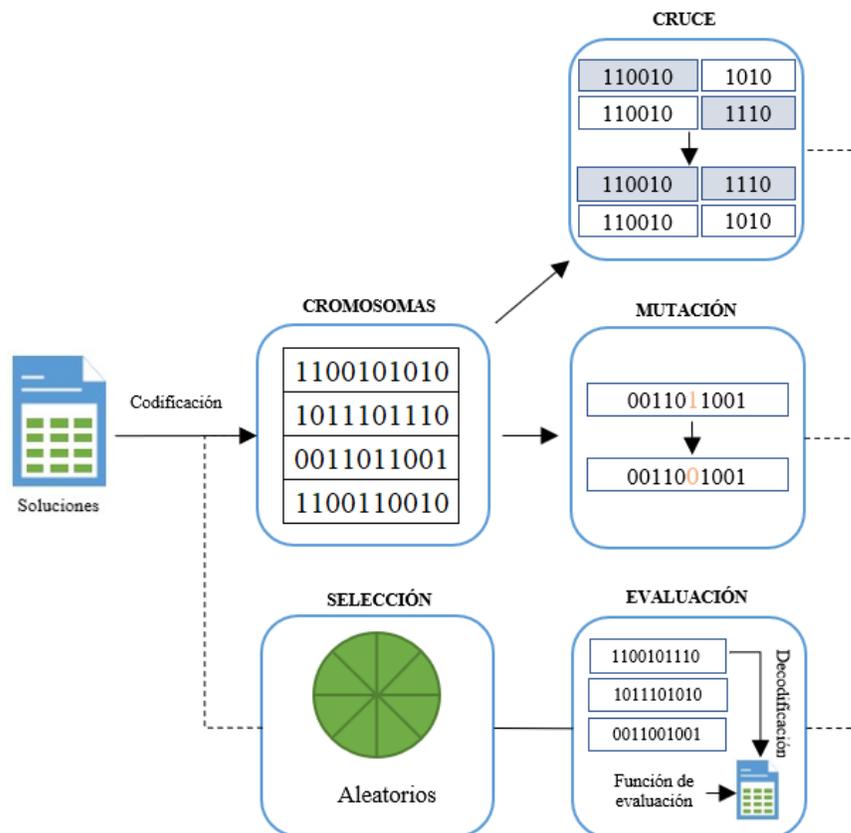


Ilustración 14. Esquema general del funcionamiento de un Algoritmo genético. Fuente: Basado en (Gamboa y Gómez, (s.f), p.23).

3.4. Diseño de Experimentos.

En una investigación experimental, la planeación involucra diseñar un experimento; un diseño estadístico de un experimento permite la posibilidad que este, sea realizado al mínimo en recursos y tiempo. El diseño estadístico es una metodología que permite plantear estrategias para seleccionar, controlar, analizar e interpretar diferentes condiciones de estudio en un fenómeno de una manera objetiva y sistemática. (Domínguez y Castaño, 2016, p.2).

El propósito de un diseño experimental es plantear la necesidad de *evaluar las condiciones de un experimento*, por ejemplo, en una operación de un proceso productivo es necesario identificar

variables que pueden afectar significativamente las respuestas del estudio; la variabilidad de un proceso identifica la calidad de un proceso productivo.

3.5. Estructuras del diseño experimental

3.5.1. Estructura de Tratamiento.

Domínguez y Castaño (2016) afirma. “La estructura de tratamiento, “consiste en el conjunto de tratamientos que el experimentador ha seleccionado para estudiar y/o comparar. Puede haber estructuras con un factor (OFAT) o varios factores; pueden considerarse todos los tratamientos (diseño factorial)”. (p.4).

- a) **Factor:** Es una variable de interés controlada hasta cierto punto por el experimentador, de la que se desea estudiar sus efectos en una o varias respuestas.
- b) **Nivel:** Permite controlar las variables dentro de un factor. Ejemplo: Si el factor es el tiempo de operación, se puede estudiar la respuesta del proceso durante tres periodos diferentes, 10, 30, 50 Segundos. En este caso se dice que el factor tiene tres niveles.
- c) **Tratamiento:** Se refiere a cada una de las combinaciones de los niveles de varios factores aplicados conjuntamente a las unidades experimentales.
- d) **Unidad experimental:** Sujeto al cual se le aplica el tratamiento y se mide la variable respuesta.
- e) **Efecto principal:** Indica la contribución que cada factor tiene sobre las variables de respuesta al modificar los niveles del factor.
- f) **Error Experimental:** Mide la variación que existe entre las observaciones sobre las unidades experimentales, es conveniente llevar a cabo de 2 o más replicas por cada tratamiento para reducir el error.

- g) Interacción:** Implica una relación o dependencia entre dos o más factores, permite precisar la estimación y reducir el error experimental.

El DOE presenta gran variedad de aplicaciones para diferentes disciplinas, este proceso hace parte de la experimentación y es una forma de averiguar cómo funciona el algoritmo propuesto.

La metodología que propone Montgomery (2004) para diseñar un experimento se compone por la selección de las variables, elección del diseño experimental, realización de experimento, análisis del experimento y finalmente es necesario identificar cual y como conviene tomar dichas observaciones(datos).

3.6. Diseño factorial fraccionado 2^k

Inicialmente un diseño factorial se define como, “un tipo de experimento que permite estudiar los efectos que varios factores pueden tener en una respuesta”. (Minitab, 2018). Este diseño factorial considera aquellos factores que tienen efectos grandes, analizando con mayor detalle los factores que obtengan la mejor resolución; según Montgomery (2004) se tiene en cuenta tres ideas clave para el uso del diseño fraccionado con varias variables:

- i) El principio de efectos esparcidos o escasez de efectos, cuando existen varias variables, es posible que el sistema esté dominado principalmente por algunos de los efectos principales y las interacciones de orden inferior.
- ii) La propiedad de proyección, los diseños factoriales fraccionados pueden proyectarse en diseños más fuertes (más grandes) en el subconjunto de los factores significativos
- iii) Experimentación secuencial, es posible combinar las corridas de dos o más diseños factoriales fraccionados para ensamblar secuencialmente un diseño más grande para estimar los efectos de los factores y las interacciones de interés.

3.6.1. La fracción un medio del diseño 2^k

“Consiste en tomar la mitad de las combinaciones de un diseño 2^k completo, es recomendada para aplicar cuando se estudian de 4 – 6 factores”. (Martín, 2015, p.1).

3.6.1.1. Diseños de resolución IV.

Se trata de diseños en los que ninguno de los efectos principales es alias de ningún otro efecto principal m_i de las interacciones de dos factores, pero las interacciones de dos factores son alias entre sí. Un diseño 2^{k-1} con $I=ABCD$ es un diseño de resolución IV (2_{IV}^{k-1}). (Ver Tabla 7).

Tabla 7. Relaciones de alias para diseños factoriales fraccionados 2^{k-p} con $k \leq 15$.

DISEÑO CON 4 FACTORES

Generadores del diseño

$$D = ABC$$

Relación de definición: $I=ABCD$

Alias

$$A = BCD$$

$$B = ACD$$

$$C = ABD$$

$$D = ABC$$

$$AB = CD$$

$$AC = BD$$

$$AD = BC$$

Fuente: (Montgomery, 2004, p.663)

4. DISEÑO METODOLÓGICO

4.1. Diseño de la investigación

Esta investigación es de tipo exploratorio-explicativo experimental transeccional de tecnología formal. El enfoque exploratorio se justifica en la revisión teórica de los elementos referentes a la secuenciación tipo Flowshop y los métodos metaheurísticos que se aplican para el problema de secuenciación. Con respecto al enfoque explicativo, se debe a que se realizará una caracterización de la dinámica de cómo funciona el sistema de producción HAS200, de tal forma que se identifiquen las variables principales para la aplicación del algoritmo –tiempo de procesamiento y tiempo de alistamiento.

Se considera un alcance transeccional, ya que se realizará un registro de la información de los tiempos del sistema producción HAS-200 durante el segundo semestre de 2018. La naturaleza de un diseño experimental se justifica en la forma como se explora la calidad de la solución y el tiempo computacional del algoritmo propuesto, de acuerdo con la parametrización del algoritmo y las posibles combinaciones de parámetros que permitan mejorar la calidad y tiempo computacional frente al problema de secuenciación en el sistema de producción HAS-200. Finalmente, es una investigación aplicada de tecnología formal, debido a que se utilizan tiempos de procesamiento y alistamiento del sistema de producción HAS-200 en las instalaciones del laboratorio de automatización de procesos de la Universidad Pontificia Bolivariana, y se desarrollará una estructura algorítmica en lenguaje de programación Visual Basic, de tal forma que se obtenga un aplicativo computacional de enfoques de solución cuantitativos que hacen parte de las técnicas de Investigación de Operaciones.

4.2. Método de investigación

Con respecto a los pasos o fases de ejecución del proyecto de investigación, Taha (2011), considera la siguiente estructura secuencial metodológica para desarrollar una propuesta de solución en el área de Investigación de Operaciones:

- Definir el problema de optimización: Se puede considerar como el alcance del problema que se desea investigar mediante procedimientos matemáticos. Este paso metodológico debe considerar la descripción de alternativas de decisión, objeto o problemática principal de estudio y las limitaciones del modelo para la solución del problema.
- Construcción del modelo de optimización: Consiste en traducir los elementos clave de la definición del problema en una estructura matemática lineal; considerando el nivel de estructura matemática, se puede resolver el problema mediante un enfoque algorítmico.
- Solución del modelo de optimización: Se establece la técnica de solución para el modelo propuesto considerando herramientas exactas, como resolución de sistemas de ecuaciones; o bien, herramientas metaheurísticas, basadas en lógica de programación.
- Validez del modelo de optimización: Fase final de la estructura metodológica, que permite comparar la solución del modelo propuesto con diferentes instancias o resultados de la literatura, o la comparación de eficiencia computacional.

Para el caso del diseño de un modelo metaheurístico asociado al problema de secuenciación de producción tipo Flowshop con restricción de tiempos de alistamiento en el sistema de producción HAS-200 del laboratorio de automatización, se aplicarán los pasos metodológicos anteriormente descritos.

4.3. Población

Tiempo de producción de las diferentes referencias que se manufactura en el sistema de producción HAS-200 que hace parte del laboratorio de automatización de procesos, de la Universidad Pontificia Bolivariana, Secciona Bucaramanga.

4.3.1. Tamaño de la muestra.

Registro del tiempo de producción de cada referencia que se manufactura en el sistema de producción HAS-200. De acuerdo con la dinámica de la utilización del laboratorio, se determina inicialmente un muestreo de ocho registros de tiempo de producción para cada referencia, de tal forma que se pueda establecer un muestreo de los tiempos que harán parte de la información considerada para la construcción del algoritmo genético asociado problema de secuenciación tipo Flowshop en el sistema de producción HAS-200.

4.4. Definición de las variables de investigación

Se considera como el tiempo de alistamiento de los productos en la HAS-200 y el tiempo de procesamiento bajo ambiente de manufactura de los productos con referencias aleatorias.

4.5. Hipótesis

De acuerdo con el diseño de la investigación y las variables se propone comprobar las siguientes hipótesis para el modelo algorítmico:

H: El modelo metaheurístico para el problema de secuenciación de producción bajo ambiente de Manufactura Flowshop con restricciones de tiempo de alistamiento en secuencia en el sistema de producción HAS-200 de la Universidad Pontificia Bolivariana obtiene mejoras en tiempos computacionales eficientes.

- a) **H₁**: La solución metaheurística se ve afectada por los tiempos de alistamiento o los tiempos de procesamiento de acuerdo con las referencias procesadas.
- b) **H₂**: Cuáles serían los mejores parámetros que permite un mayor rendimiento en tiempo computacional y calidad de solución del problema.

5. RESULTADOS

5.1. Modelamiento del Algoritmo Clarke & Wright

La heurística construye la secuenciación de los trabajos respecto al tiempo de producción, se moldeo el algoritmo teniendo en cuenta las estaciones de trabajo, los tiempos alistamiento y procesamiento; donde los *trabajos* se asocian a las ciudades y los *tiempos de alistamiento* S_{ij} se asocian a las distancias d_{ij} . Por medio de un diagrama de Gantt se plantea la función objetivo (Ecuación 10), donde se evidencia la holgura que representa la situación actual de las dos estaciones de trabajo de la HAS-200. (Ver Ilustración 15).

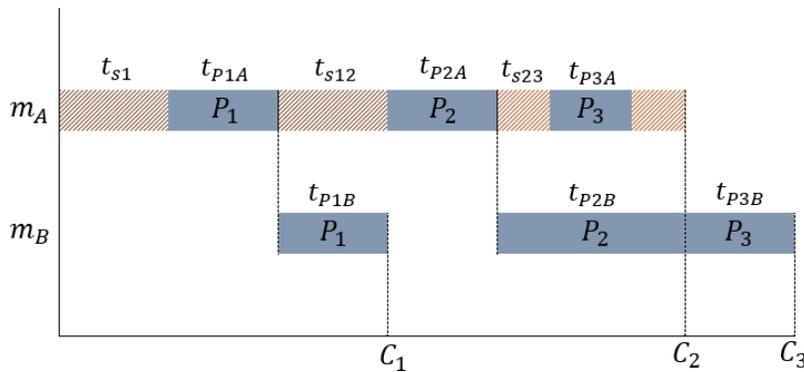


Ilustración 15. Diagrama de Gantt de situación actual del sistema HAS200

El C_{\max} que se espera minimizar (Ecuación 10), está compuesto por la sumatoria de los tiempos de alistamiento, donde $S_{[j-1][i]}$ que se produce entre $(j-1)$ -ésimo e i -ésimo trabajos de la secuenciación representada por $[i-1]$ e $[i]$, más la suma de los tiempos de procesamiento de los n trabajos.

$$\text{Min: } C_{\max} = \sum_{i=1}^n tS[i-1][i] + \sum_{i=1}^n tp \quad (8)$$

Sujeto a:

$$C_{max} = \max\{C_1, C_2, C_3\} \quad (9)$$

$$C_{max} = C_3 \quad (10)$$

$$C_{max} = C_n, \quad n = \text{último trabajo} \quad (11)$$

Para llevar a cabo la solución inicial, se construye la siguiente información que alimenta el algoritmo de la siguiente manera:

- a) Se seleccionaron las tres referencias, que produce la máquina.
- b) Se tomaron los tiempos de alistamiento y procesamiento, de acuerdo con las variaciones de los n elementos $\rightarrow n(n-1)$, donde $n = 6$, que son los tiempos de producción de 6 combinaciones de las tres referencias anteriormente mencionadas.

Con la información de los tiempos de cada estación se procede a determinar la matriz de ahorros. Para los tiempos de procesamiento (T_p) se construye el vector de cada estación de trabajo de la HAS-200 y para la matriz de los tiempos de alistamiento (T_{sij}) se tiene en cuenta, la tardanza de dar una nueva orden a la máquina. (Ver Tabla 7 y 8)

Tabla 8. Vector Tiempos de procesamiento de los trabajos. (Ejemplo)

T_p_{ij}	1	2	3	4	5	6
Est. 1	5	2	1	4	3	1
Est. 2	4	4	3	1	4	3

Tabla 9. Matriz de tiempos de alistamiento. (Ejemplo).

T_{sij}	1	2	3	4	5	6
1	---	3	2	1	2	3
2	2	---	3	4	1	4
3	1	1	---	1	2	2
4	2	2	5	---	1	2
5	4	1	3	2	---	1
6	1	2	1	5	3	---

Utilizando la heurística del método de ahorro, se obtiene un vector solución con su respectivo C_{\max} , este vector solución es la **secuencia factible** de producción en cada una de las referencias, donde la mejor secuencia de producción es: 6 – 4 – 3 – 1 – 2 – 5 $C_{\max} = 35$ Seg. Teniendo en cuenta que el primer trabajo con el menor tiempo de alistamiento es el producto 6; la programación de esta solución se representa en la siguiente Ilustración. (Ver Ilustración 16).

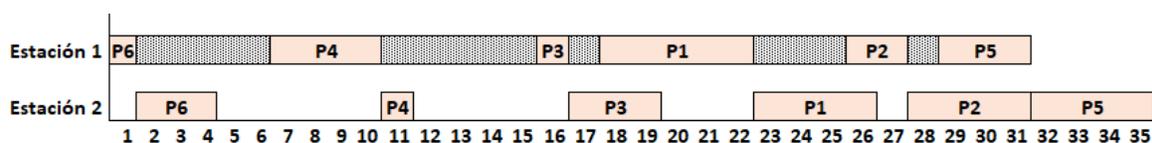


Ilustración 16. Carta Gantt de la solución inicial del algoritmo de Clarke & Wright.

La solución inicial, es un procedimiento analítico de secuenciación, en otras palabras, se determina la situación actual de los tiempos de producción.

5.2. Propuesta del Algoritmo Genético

Para comprender el funcionamiento de la estructura algorítmica de la HAS-200, se describe cada operador que lo compone. Para mejorar el tiempo del proceso de las órdenes lanzadas se definen los siguientes parámetros. (Ver Tabla 10).

Tabla 10. Parámetros del Algoritmo Genético

PARAMETRO	DEFINICIÓN
Estaciones	Maquinas en el proceso de producción
Generaciones	Cantidad de iteraciones con individuos que se ejecutan en el AG
Individuos	Secuenciación de cada vector solución
Porcentaje mutación	Operador de mutación de un individuo por generación
Factor Mutación	Probabilidad de selección de individuos por cada generación

La función objetivo del algoritmo genético (Ecuación 12) minimiza la holgura de cada referencia lanzada en la HAS-200, para encontrar la secuenciación ideal y mejorar la demora de cualquier producto lanzado.

$$\text{Min: } (f(x) = \text{Fitness} = \text{Min Makespan}) \quad (12)$$

5.2.1. Generación de la población inicial.

La población inicial es la secuencia, que se obtiene del método de los ahorros. Para contextualizar, los cromosomas será cada secuencia {6 – 4 – 3 – 1 – 2 – 5} y los genes están dados por la posición de cada producto, en la ilustración 17 el gen se encuentra en la segunda posición, es decir, el producto 4 y el alelo es la información que contiene cada producto {Tiempo de procesamiento + Tiempo de Alistamiento}, cuya información será almacenada internamente en el código VBA para determinar el mejor C_{\max} . (Ver Tabla 11).

Tabla 11. Generación de la población inicial.

SECUENCIA = CROMOSOMA						CMAX
6	4	3	1	2	5	35

Nota: El gen del 4to producto se encuentra en la segunda posición.

5.2.2. Cruzamiento

Para generar la descendencia se procede al cruzamiento, y se definen los siguientes criterios:

- i). El punto de cruce de *n-puntos*, es almacenado por aleatorios donde cada gen tiene una probabilidad acumulada para operar el intercambio de los datos.
- ii). El cruce se hace hacia adelante, por tal razón cada gen empieza su intercambio de derecha a izquierda.

Con base a lo anterior, se puede comprobar la disminución del tiempo de procesamiento, en la primera generación, donde el mejor individuo tiene el mejor C_{max} : 28 Seg y se ejecutó el cruce de 2-puntos. (Ver Tabla 12).

Tabla 12. Cruzamiento del AG.

INDIVIDUO						CMAX INDIVIDUO
6	4	3	1	2	5	35
4	6	2	3	1	5	28

5.2.3. Mutación.

Se denomina el factor mutación para tener la probabilidad de escoger cierta cantidad de individuos por cada generación y aplicarle un rango de cruce de {5% - 10%}. Los parámetros que mejor representan para esta solución son los siguientes. (Ver Tabla 13).

Tabla 13. Parámetros para generar la mejor solución (Ejemplo)

PARÁMETROS	
Generaciones	20
Individuos	5
Porcentaje mutación	0,05
Factor Mutación	0,3

De las 20 generaciones, el 30% de los individuos se cruzarán siempre y cuando el aleatorio cumpla con la condición, por otro lado, si de los individuos seleccionados cumple con una probabilidad por debajo del 0,05. (Ver Tabla 14).

Tabla 14. Mejor secuencia (Ejemplo).

Secuencia	6	4	3	1	2	5
Mejor Secuencia	4	6	3	1	2	5

Como en este caso, las generaciones son muy pocas y el factor mutación es más grande, la probabilidad de que exista una mutación es baja, lo cual implica que la mejor secuencia sea de {4 – 6 – 3 – 1 – 2 – 5}.

5.2.4. Nueva población.

Considerando la condición elitista, se construye la siguiente población, donde el mejor C_{\max} será el más apto para generar la siguiente descendencia. En la ilustración 19 se muestra la mejora de los tiempos de alistamiento. (Ver Ilustración 19).

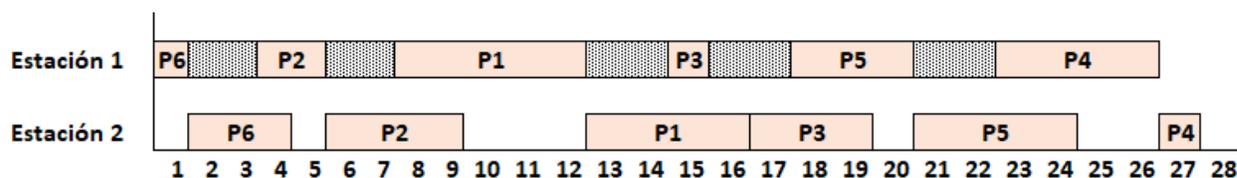


Ilustración 17. Diagrama de Gantt del mejor individuo. (Ejemplo).

5.3. Implementación del AG a la celda de manufactura HAS-200.

Después de caracterizar la estructura algorítmica, se procede a ejecutar el algoritmo en la HAS-200. Para determinar cuál será la mejora solución, remítase a la sección 5.1 párrafo 3, donde se explica la selección de los productos para la toma de los tiempos de procesamiento y alistamiento. A continuación, en las siguientes tablas, se hace un promedio de las muestras que se han tomado por los seis productos. (Ver Tabla 15 y 16).

Tabla 15. Tiempos de procesamiento de la HAS-200.

Productos	1	2	3	4	5	6
Proce. Estación 1	120	126	115	131	113	122
Proce. Estación 2	77	72	72	73	70	71

Tabla 16. Tiempos de Alistamiento de la HAS-200.

T_{sij}	1	2	3	4	5	6
1	---	3,5	3,4	3,3	2,8	3,1
2	3,0	---	3,2	3,9	3,7	3,6
3	3,7	3,6	---	3,5	3,1	3,3
4	3,7	3,6	3,5	---	3,2	3,4
5	2,5	3,5	3,2	3,1	---	3,2
6	3,0	4,0	3,4	3,7	3,5	---

Los tiempos registrados en las tablas 15 y 16, están en minutos. Al ejecutar el algoritmo se obtiene que el C_{max} mejora un 46%, con un tiempo computacional de 9,878 segundos. Para observar a más detalle los resultados de la HAS-200 diríjase al Anexo 1.

5.4. Pseudocódigo del programa propuesto (C&W – AG)

Seguidamente se eslabona el pseudocódigo del programa creado en VBA, que comprende el diseño y su desarrollo para efectos del objeto de estudio. (Ver Ilustración 20 y 21).

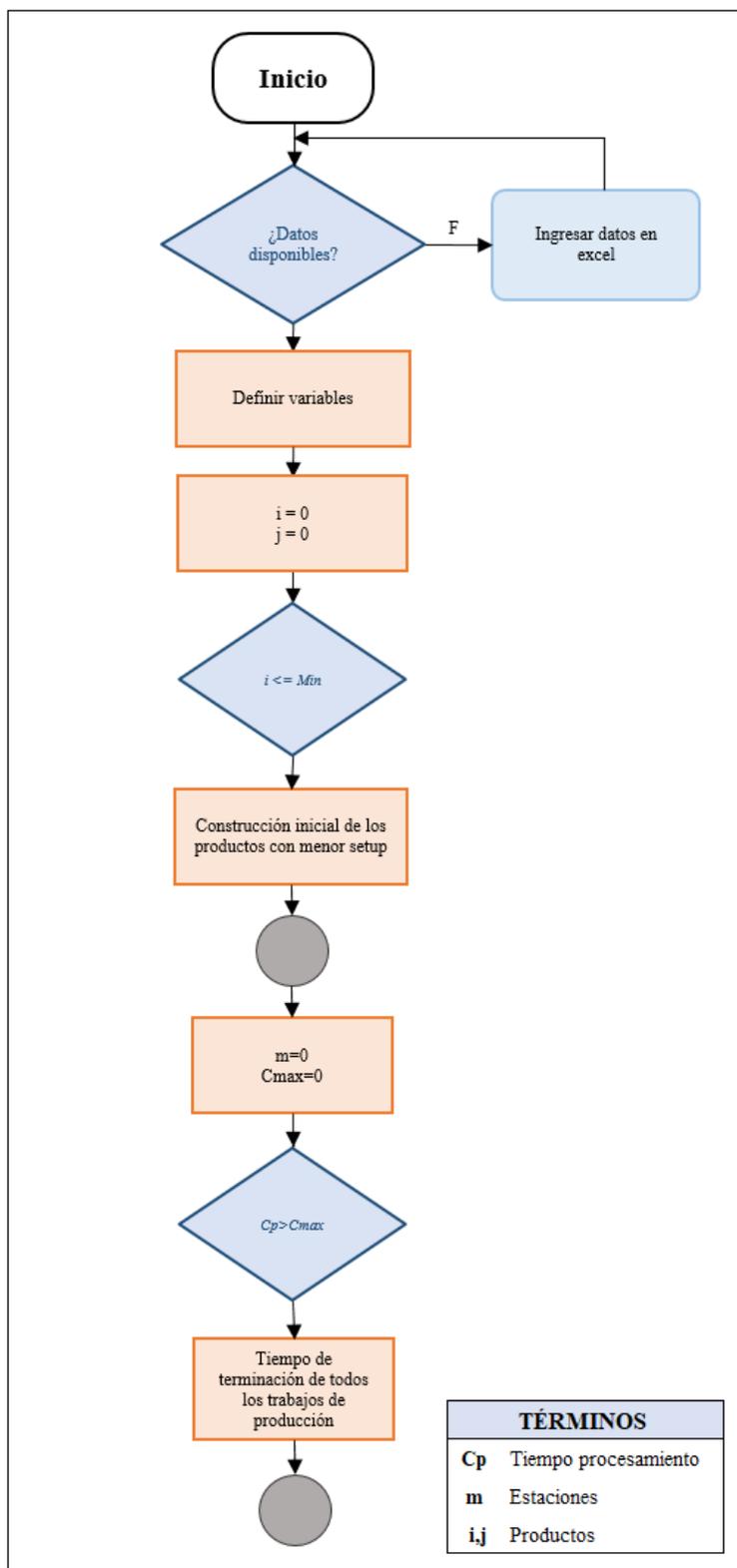


Ilustración 18. Diagrama de flujo general para generación de la solución inicial.

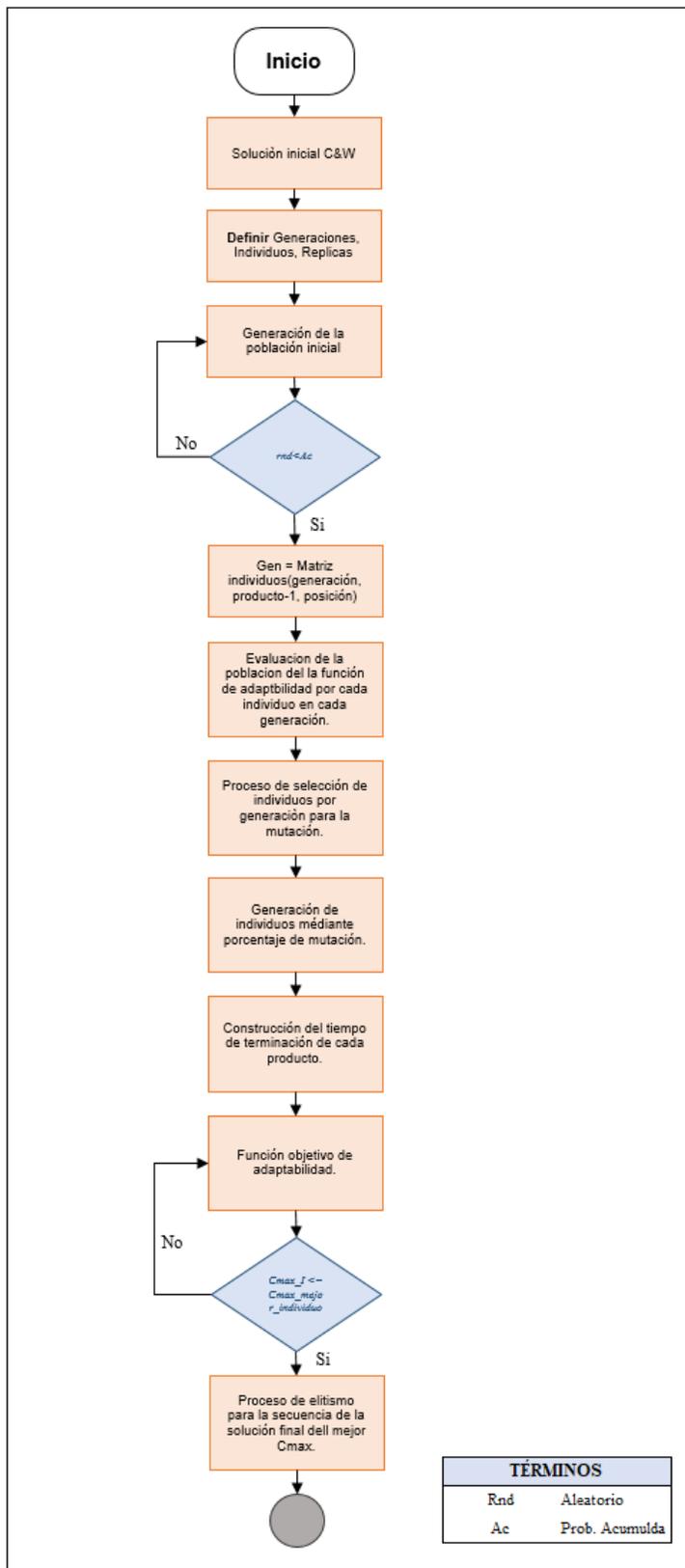


Ilustración 19. Diagrama de flujo general del Algoritmo genético propuesto.

6. ANÁLISIS DE LOS RESULTADOS

En esta sección se evalúa la calidad del algoritmo propuesto en VBA con interfaz en Excel, con un diseño factorial fraccionado 2^k ; se hace una búsqueda en artículos y libros digitales de la base de datos de la Biblioteca de la Universidad Pontificia Bolivariana, tales como: EBSCO HOST, Science Direct, Scopus, Scielo, entre otros.

Se ha compilado una serie de pruebas computacionales compuestas por ocho replicas, en la ecuación 13, se obtiene la cantidad experimental de 128 pruebas por cada instancia y 64 experimentos elementales por cada sesión.

$$\# \text{ Exp Total} = (2^k) * R = 128 \quad (13)$$

$$\# \text{ Exp. Sesión} = (\# \text{ Exp Total}/2) = 64 \quad (14)$$

En la tabla 17 se identifican las variables que representan el diseño experimental. (Ver Tabla 17).

Tabla 17. Variables para el arreglo experimental factorial 2^k fraccionado.

TIPO DE VARIABLE	NOMBRE DEL FACTOR	ABREV.
Dependiente	Makespan	Cmax
	Tiempo Computacional	CPU
	Generaciones	Gen
Independientes	Individuos	Individuos
	Porcentaje Mutación	P_Mutación
	Factor Mutación	F_Mutación

6.1. Diseño de experimentos (DOE) para validar el AG.

En el presente apartado, se espera hallar el parámetro más idóneo que permita mejorar la capacidad de respuesta del makespan y el tiempo computacional, para ello se plantean los siguientes escenarios:

- i). **Flowshop con tiempos de alistamiento en cero**, en la librería OR-Library desde <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/flowshopinfo.html>, "Benchmarks for basic scheduling problems" de Taillard en 1993, los grupos contienen diferentes combinaciones de la cantidad de trabajos n y número de máquinas m . Las combinaciones $n \times m$ son: $\{20, 20, 50, 50\} \times \{5, 20, 5, 20\}$; en la tabla 18, se exponen las matrices seleccionadas que compone cada instancia. (Ver Tabla 18).

Tabla 18. Instancias con tiempos de alistamiento en cero.

INSTANCIAS	MATRIZ	UPPER BOUND	LOWER BOUND
20X5	1	1278	1232
	3	1081	1073
	9	1230	1206
20X20	1	2297	1911
	3	2326	1844
	10	2178	1900
	2	2834	2808
50X5	4	2751	2740
	6	2829	2793
	10	3091	3046

- ii). **Flowshop con tiempos de alistamiento**, de la librería SOA: (Ruiz, Rubén, Stützele, 2006), evalúan el rendimiento del algoritmo con instancias de referencia que se pueden descargar desde <http://www.upv.es/gio/r Ruiz>, estas instancias se dividen en cuatro conjuntos y se ha seleccionado la instancia SDST10 que tienen los tiempos de preparación distribuidos uniformemente en el rango $[1, 9]$, estas instancias cuentan con el tiempo de procesamiento de Taillard y también toman como referencia las

muestras de las instancias de Taillard. A continuación, se demuestra las matrices seleccionadas para los tiempos de alistamiento. (Ver Tabla 19).

Tabla 19. Instancias con tiempos de alistamiento.

INSTANCIAS SDST10	MATRIZ	UPPER BOUND	LOWER BOUND
M1_Ta001	1	1278	1232
M4_Ta003	3	1081	1073
M4_Ta009	9	1230	1206
M0_Ta021	1	2297	1911
M12_Ta023	3	2326	1844
M9_Ta030	10	2178	1900
M0_Ta032	2	2834	2808
M2_Ta034	4	2751	2740
M1_Ta036	6	2829	2793

6.1.1. Resultados experimentales con tiempos de alistamiento en cero.

En esta sección en adelante se evalúan las instancias propuestas en la literatura para determinar, si el algoritmo presenta respuestas dentro del rango aprobado en las instancias que proponen los autores anteriormente mencionados.

6.1.1.1. Análisis de varianza para instancias Tai_SDST.

Para evaluar la eficiencia y calidad del AG_UPB. Se han realizado 3 experimentos por cada instancia con las siguientes variables de entrada: Generaciones {200 – 400 – 1295}, Individuos {150 – 446 - 854}, Porcentaje {0,05 – 0,1}, Factor {0,2 – 0,3 – 0,4 – 0,7}.

6.1.1.2. Análisis de varianza para instancias 20X5_1.

En la Tabla 20 se resume el análisis de varianza del experimento, los efectos que tienen estimaciones grandes están incluidos en el modelo. (Ver Tabla 20).

Tabla 20. Análisis de varianza de la instancia 20X5_1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	560,7	560,67	18,26	0,001
Individuos	1	3850,7	3850,67	125,39	0,000
Porcentaje	1	7210,7	7210,67	234,81	0,000
Factor	1	10,7	10,67	0,35	0,564
Generaciones*Individuos	1	16,7	16,67	0,54	0,472
Generaciones*Porcentaje	1	170,7	170,67	5,56	0,031
Generaciones*Factor	1	1472,7	1472,67	47,96	0,000
Error	16	491,3	30,71		
Total	23	13784			

Los factores que influyen en la respuesta se determinan por el porcentaje de mutación, la cantidad de individuos generados y la interacción entre las generaciones*factor de mutación, por otro lado, las generaciones y los individuos no presenta un efecto significativo. (Ver Figura 3)

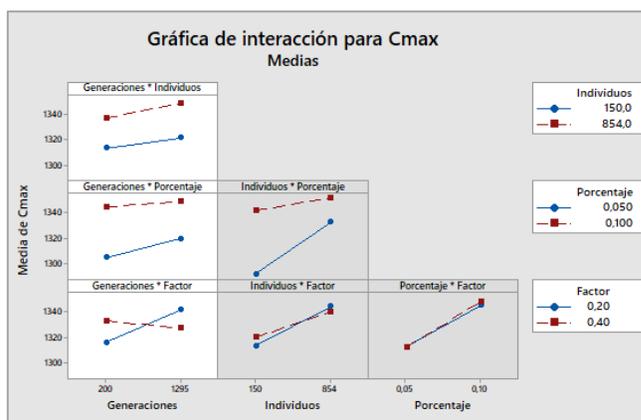


Figura 3. Interacciones del c_{max} para las instancias 20X5_1.

Además, se estima que entre mayor sea el número de generaciones, la interacción con el factor mutación aumenta. Por otro lado, Minitab presenta una limitación a la hora de realizar un diseño experimental pues, confunde efectos principales con las interacciones y tiende a eliminar los factores no significativos, para el caso de la interacción generación*individuos se tiene en

cuenta, los individuos los cuales generan mayor efecto significativo. Los efectos principales que se esperan tomar para esta instancia están representados en la figura. (Ver figura 4).

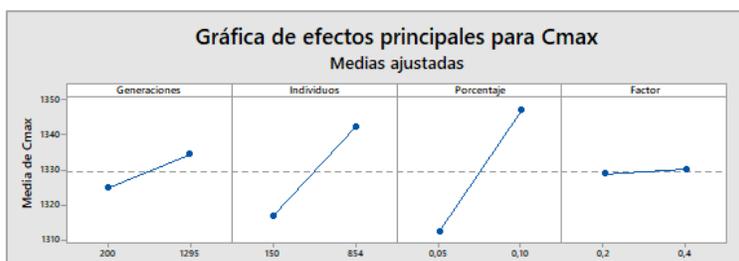


Figura 4. Efectores principales para el C_{max} de la instancia 20X5_1

En cuanto al análisis de varianza del tiempo computacional, se comprueba que no existe efectos significativos, es decir que no afecta al C_{max} , en la siguiente figura se puede comprobar que el tiempo de respuesta es directamente proporcional a la cantidad de individuos*generaciones, pero gracias al factor mutante se reduce el tiempo de operación. (Ver Figura 5)

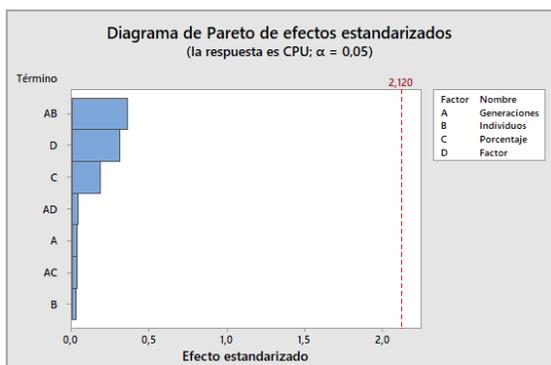


Figura 5. Diagrama de Pareto efectos del tiempo computacional de instancia 20X5_1.

6.1.1.3. Análisis de varianza para instancias 20X5_3.

En la tabla 21, se analiza la varianza para la instancia de 20 máquinas y 5 trabajos. (Ver tabla 21).

Tabla 21. Análisis de varianza para la instancia 20X5_3.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	3901,5	3901,5	11,99	0,003
Individuos	1	7072,7	7072,7	21,73	0,000
Porcentaje	1	30960,2	30960,2	95,13	0,000
Factor	1	560,7	560,7	1,72	0,208
Generaciones*Individuos	1	661,5	661,5	2,03	0,173
Generaciones*Porcentaje	1	42,7	42,7	0,13	0,722
Generaciones*Factor	1	793,5	793,5	2,44	0,138
Error	16	5207,3	325,5		
Total	23	49200			

Para esta instancia las generaciones presentan un nivel de significancia muy alto, es decir que entre más generaciones ingresen en el programa, se pueden hallar mejores respuestas (makespan). (Ver figura 6).

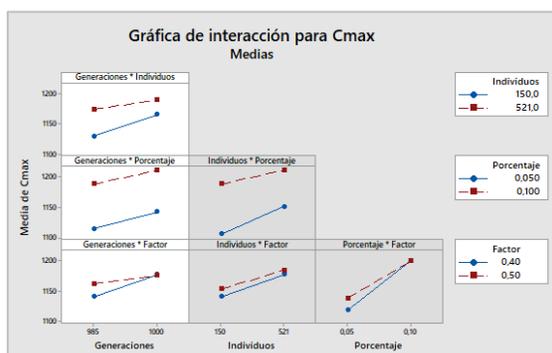


Figura 6. Interacciones del c_{max} para las instancias 20X5_3.

Los efectos principales para esta instancia deben estar por debajo de la media para llegar a la respuesta propuesta. (Ver Figura 7).

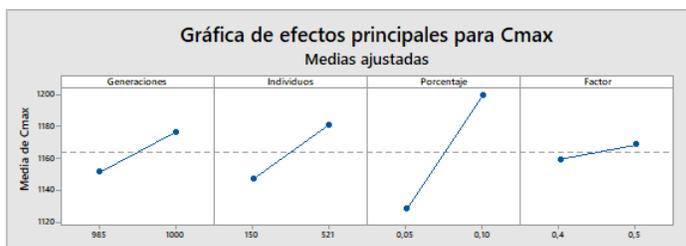


Figura 7. Efectos principales para el C_{max} de la instancia 20X5_3.

Para el rendimiento del programa, se evidencia que a mayor porcentaje de mutación la capacidad del programa mejora notablemente. (Ver figura 8).

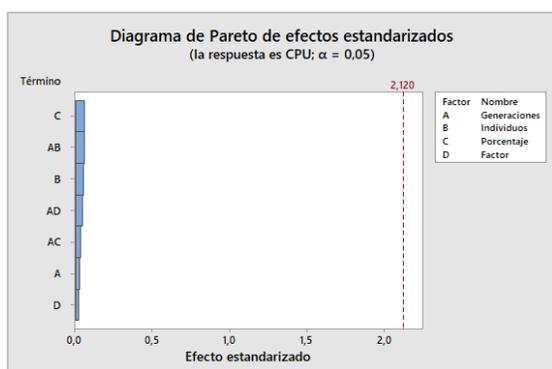


Figura 8. Diagrama de Pareto efectos del tiempo computacional de instancia 20X5_3.

6.1.1.4. Análisis de varianza para instancias 20X5_9.

En la tabla 22, se muestra el último grupo de instancias con 20 máquinas y 5 trabajos, se determina lo siguiente. (Ver Tabla 22).

Tabla 22. Análisis de varianza para instancias 20X5_9.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	18,4	18,37	0,03	0,876
Individuos	1	495	495,04	0,68	0,423
Porcentaje	1	4788,4	4788,37	6,55	0,021
Factor	1	805	805,04	1,1	0,031
Generaciones*Individuos	1	442	442,04	0,6	0,448
Generaciones*Porcentaje	1	442	442,04	0,6	0,448

Generaciones*Factor	1	925	925,04	1,27	0,277
Error	16	11696,7	731,04		
Total	23	19612,6			

Se puede observar e interpretar que los efectos significativos influyen en los dos últimos factores, pues a medida que se deja un porcentaje de mutación del 5% y un porcentaje del 40%, se hallan mejores respuestas. (Ver figura 9).

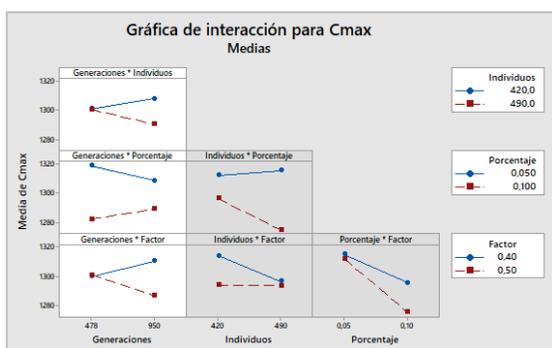


Figura 9. Interacciones del c_{max} para las instancias 20X5_9

Los efectos principales que se consideran para mejorar la respuesta deben estar por debajo de la media. (Ver figura 10).

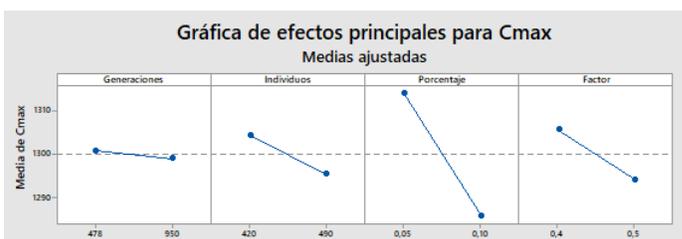


Figura 10. Efectos principales para el C_{max} de las instancias 20X5_9.

Para este caso se han realizado experimentos con una cantidad mayor de individuos que generaciones, por tal razón el tiempo computacional se ve afectado en dicho factor. (Ver figura 11).

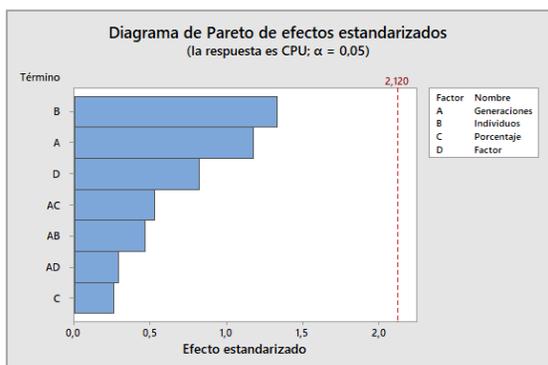


Figura 11. Diagrama de Pareto efectos del tiempo computacional de instancia 20X5_9

Para las instancias de 20 trabajos con 5 máquinas se comprueba que las variables que ejercen mayor efecto en la ejecución del algoritmo, es el porcentaje de mutación y la cantidad de individuos que se generen, por ser el resultado más cercano al C_{\max} propuesto en la librería.

6.1.1.5. Análisis de varianza para instancia 20X20_1.

En la tabla 23, se presenta el análisis de varianza para las instancias de 20 trabajos con 20 máquinas. (Ver Tabla 23).

Tabla 23. Análisis de varianza para instancia 20X20_1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	13443	13442,7	2,16	0,161
Individuos	1	384	384	0,06	0,807
Porcentaje	1	2204	2204,2	0,35	0,056
Factor	1	42504	42504,2	6,83	0,019
Generaciones*Individuos	1	34504	34504,2	5,54	0,032
Generaciones*Porcentaje	1	25091	25090,7	4,03	0,062
Generaciones*Factor	1	171	170,7	0,03	0,871
Error	16	99639	6227,5		
Total	23	217940			

En este punto el factor de mutación se estabiliza a un 40%, valor que da el mínimo C_{\max} , si se obtiene un factor mutación del 40%, es considerable tener más individuos que generaciones, es

por ello, que el nivel de significancia está en el factor mutación y las generaciones*individuos. (Ver figura 12).

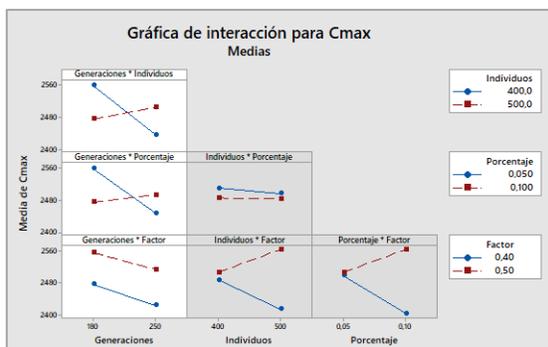


Figura 12. Interacciones del c_{max} para las instancias 20X20_1

Los efectos principales también determinan que, a menor porcentaje y factor de mutación, se puede hallar fácilmente la respuesta. (Ver figura 13).

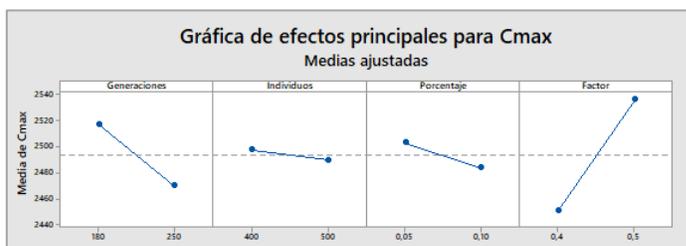


Figura 13. Efectos principales para instancias 20X20_1

Se ha comprobado que, al incrementar los individuos se debe tener más control en el factor mutación y porcentaje mutación, al controlar estos parámetros se puede reducir el tiempo computacional considerablemente. (Ver figura 14)

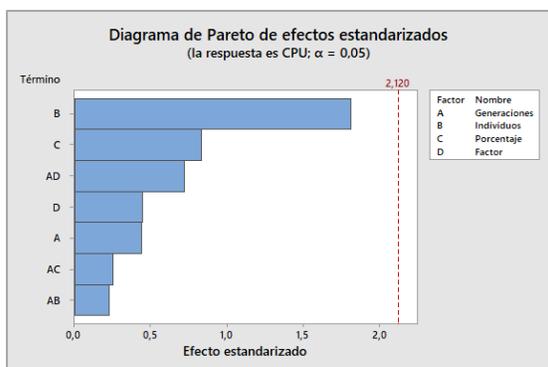


Figura 14. Diagrama de Pareto efectos del tiempo computacional de instancia 20X20_1.

6.1.1.6. Análisis de varianza instancias 20X20_3.

En la tabla 20, se emite la instancia de 20 máquinas y 20 trabajos, con tiempos de procesamiento de la tercera matriz de la librería. (Ver Tabla 24).

Tabla 24. Análisis de varianza para las instancias 20X20_3.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	39701	39700,6	8,17	0,021
Individuos	1	1139	1139,1	0,23	0,641
Porcentaje	1	2186	2185,6	0,45	0,521
Factor	1	2730	2730,1	0,56	0,475
Generaciones*Individuos	1	5	5,1	0	0,975
Generaciones*Porcentaje	1	885	885,1	0,18	0,681
Generaciones*Factor	1	16066	16065,6	3,31	0,107
Error	8	38877	4859,6		
Total	15	101587			

Para esta instancia se incrementan los tiempos de procesamiento y aun así, como anteriormente se ha mencionado, el factor mutación juega un papel importante, pues es quien mejor representa la calidad de respuesta del makespan, por otro se empieza a notar que un alto porcentaje de mutación la respuesta aún puede mejorar. (Ver Figura 15).

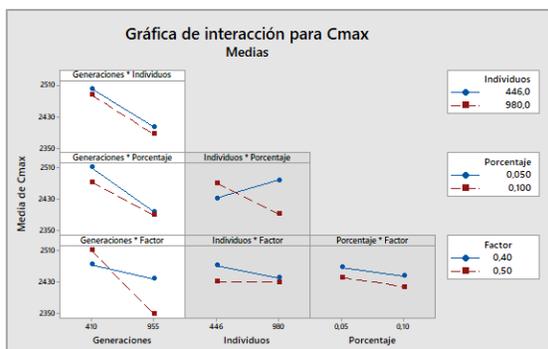


Figura 15. Interacciones del c_{\max} para las instancias 20X20_3

Por tal razón existe un grado de significancia entre individuos y los dos factores anteriormente mencionados; según lo anterior, para tener la mejor respuesta para esta instancia los efectos principales recomendados se exponen en la siguiente figura. (Ver figura 16).

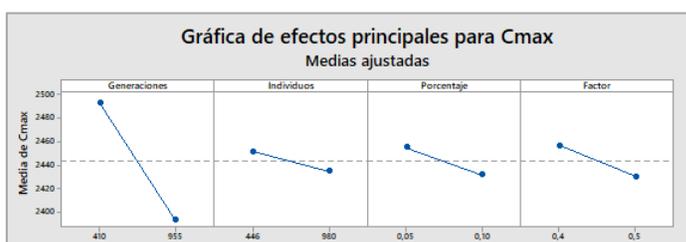


Figura 16. Efectos principales para las instancias 20X20_3.

Cuando se aumentan las generaciones, se aumentan los individuos, por tanto, es normal que el tiempo computacional se vea afectado. (Ver Figura 17).

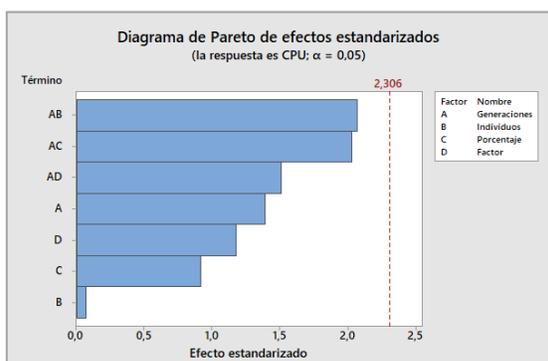


Figura 17. Diagrama de Pareto efectos del tiempo computacional de instancia 20X20_3

6.1.1.7. Análisis de varianza para instancia 20X20_10

De acuerdo con el inciso 6.1.1.4 y 6.1.1.5, los resultados en el análisis de varianza no cambian, pues se siguen empleado los factores que mejor representan el makespan y los resultados son similares. (Ver Tabla 25).

Tabla 25. Análisis de varianza para instancia 20X20_10.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	715	715	0,14	0,717
Individuos	1	2625	2625	0,5	0,049
Porcentaje	1	40590	40590,4	7,73	0,013
Factor	1	5859	5859,4	1,12	0,307
Generaciones*Individuos	1	8251	8251	1,57	0,228
Generaciones*Porcentaje	1	4082	4082	0,78	0,391
Generaciones*Factor	1	5735	5735	1,09	0,312
Error	16	84040	5252,5		
Total	23	151898			

Como se muestra en la figura, la interacción entre generaciones*individuos es quien mejor representaría al parámetro de entrada, después de calibrar los demás factores. Las generaciones determinan que, a menor cantidad de datos, es más seguro un alto grado de generaciones y bajo individuos. (Ver figura 18).

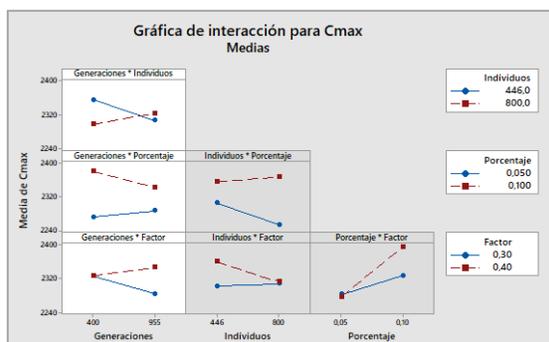


Figura 18. Interacciones del c_{max} para las instancias 20X20_10

Cuando se aumentan los individuos, es recomendable bajar el porcentaje de mutación, de esta manera también la respuesta obtiene grandes resultados. (Ver figura 19).

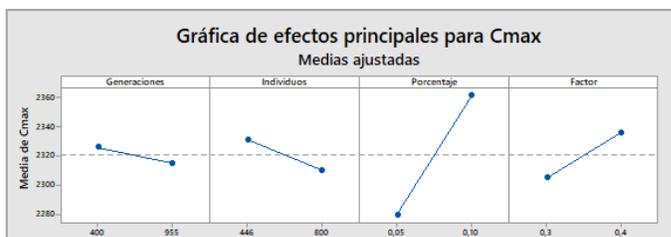


Figura 19. Gráfica de efectos principales para el C_{max} de instancias 20X20_10

Como se experimentó el aumento de individuos, el tiempo computacional aumento, sin embargo, con un factor de mutación del 40%, se puede ejecutar sin ningún inconveniente. (Ver Figura 20).

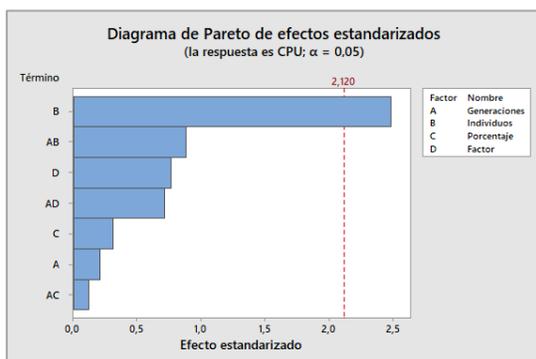


Figura 20. Diagrama de Pareto para tiempo computacional

6.1.1.8. Análisis de varianza para instancia 50X5_2.

En la tabla 26, se hace el análisis de varianza con 50 trabajos y 5 máquinas. (Ver tabla 26).

Tabla 26. Análisis de varianza para instancias 50X5_2.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	247	247	0,06	0,804
Individuos	1	4788,4	4788,4	1,23	0,284

Porcentaje	1	737	737	0,19	0,050
Factor	1	1335	1335	0,34	0,567
Generaciones*Individuos	1	477	477	0,12	0,731
Generaciones*Porcentaje	1	7385	7385	1,89	0,188
Generaciones*Factor	1	1520	1520	0,39	0,541
Error	16	62363,3	3897,7		
Total	23	78853			

A igual que el anterior inciso a pesar de aumentar la cantidad de trabajos, y reducir a cantidad de las maquinas, el factor que predomina en la respuesta es el porcentaje de mutación, en esta sección se comprueba que al utilizar un porcentaje de mutación del 10% mejora la calidad de respuesta. (Ver figura 21)

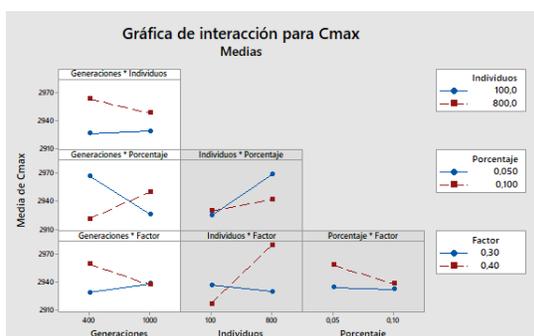


Figura 21. Interacciones del c_{\max} para las instancias 50X5_2.

En la siguiente figura, se representan los efectos principales que se recomiendan para la instancia. (Ver Figura 22).

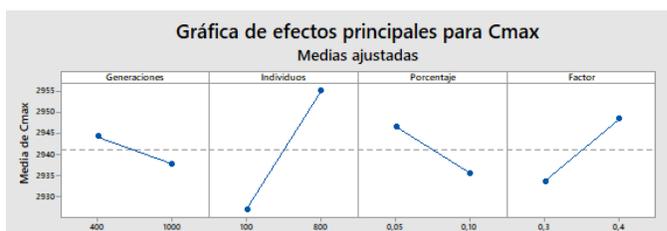


Figura 22. Grafica de efectos principales para el C_{\max} de instancias 50X5_2

Como anteriormente se ha mencionado, el tiempo computacional se ve afectado por la cantidad de generaciones y el porcentaje de mutación, a mayores generaciones y bajo nivel de mutación más tarda el programa en ejecutarse. (Ver figura 23).

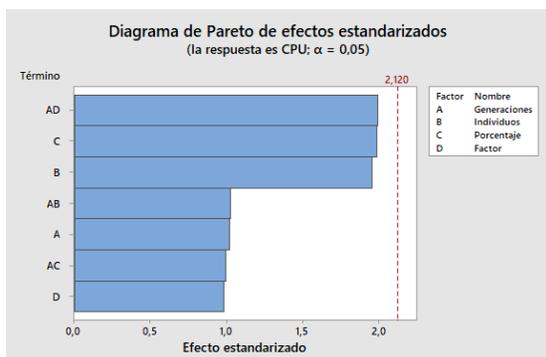


Figura 23. Diagrama de Pareto para tiempo computacional de la instancia 50X5_2

6.1.1.9. Análisis de varianza de instancia 50X5_4

En la tabla 27, se demuestra el análisis de varianza con un aumento en los tiempos de procesamiento de 50 trabajos con 5 máquinas. (Ver tabla 27).

Tabla 27. Análisis de varianza de la instancia 50X5_4.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	35721	35721	1,81	0,216
Individuos	1	14042	14042	0,71	0,424
Porcentaje	1	8930	8930	0,45	0,521
Factor	1	43264	43264	2,19	0,177
Generaciones*Individuos	1	8930	8930	0,45	0,521
Generaciones*Porcentaje	1	1892	1892	0,1	0,765
Generaciones*Factor	1	94864	94864	4,79	0,06
Error	8	158287	19786		
Total	15	365931			

A pesar de aumentar los tiempos de procesamiento, las variables de entrada no afectan mucho en la respuesta, pues se siguen empleando los parámetros de un factor del 40% y un porcentaje de 1% a una cantidad moderada de generaciones, pero un aumento significativo en los individuos. (Ver figura 24).

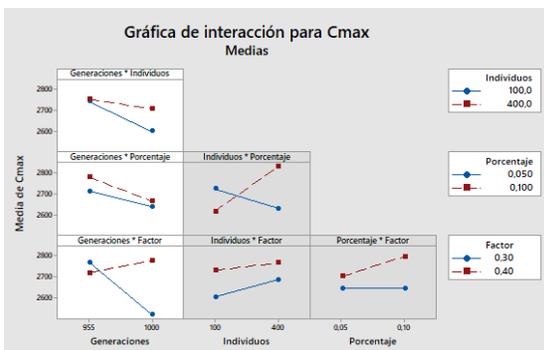


Figura 24. Interacción para el C_{max} de la instancia 50X5_4.

Sin embargo, también se comprueba que se puede llegar a las mismas soluciones con los siguientes parámetros de entrada. (Ver Figura 25).

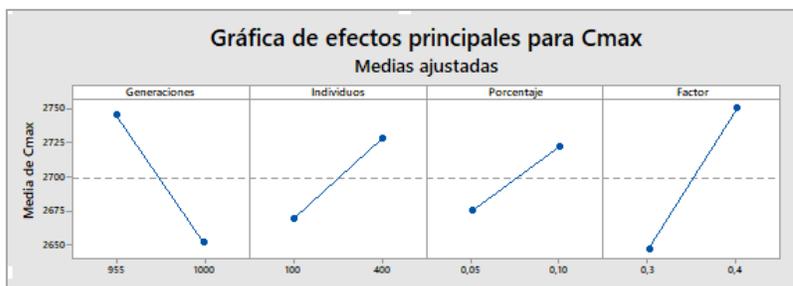


Figura 25. Efectos principales de la instancia 50X5_4.

El problema que presentan estas variables de entrada, son los individuos y el factor de mutación, ya que el tiempo computacional tarda un poco más. (Ver figura 26).

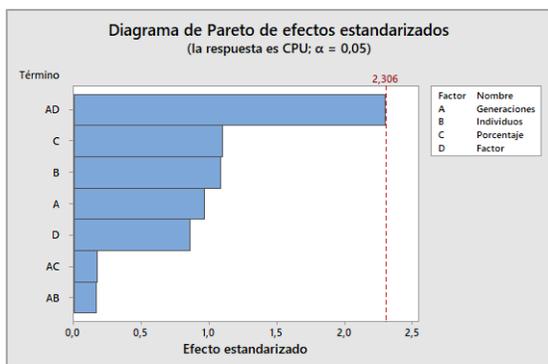


Figura 26. Diagrama de Pareto para el Tiempo computacional de las instancias 50X5_4.

6.1.1.10. Análisis de varianza de instancia 50X5_6

En la tabla 28, se demuestra el análisis de varianza para 50 trabajos y 5 máquinas con tiempos de procesamiento, inferiores al anterior. (Ver Tabla 28).

Tabla 28. Análisis de varianza para la instancia 50X5_6.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	85,6	85,6	0,03	0,876
Individuos	1	8145,1	8145,1	2,47	0,155
Porcentaje	1	29326,6	29326,6	8,89	0,018
Factor	1	10,6	10,6	0	0,956
Generaciones*Individuos	1	5439,1	5439,1	1,65	0,235
Generaciones*Porcentaje	1	7439,1	7439,1	2,25	0,172
Generaciones*Factor	1	1501,6	1501,6	0,45	0,519
Error	8	26401,5	3300,2		
Total	15	78348,9			

Cuando se disminuyen los tiempos de procesamiento se determina que el porcentaje tiene un efecto significativo, sin embargo, la respuesta no tiene gran diferencia cuando los tiempos cambian. (Ver figura 27).

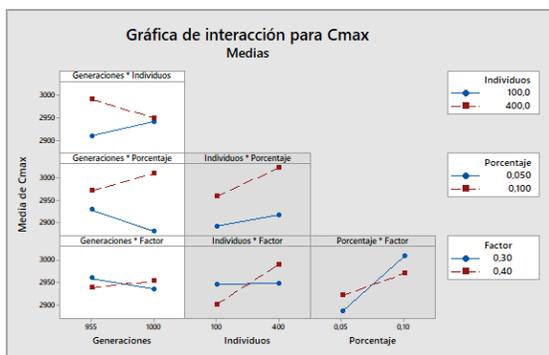


Figura 27. Interacción para el C_{max} de la instancia 50X5_6.

Por tanto, se sugiere disminuir los individuos y el porcentaje de mutación a 5%, en la siguiente graficas se evidencian los efectos principales para el C_{max} . (Ver Figura 28).

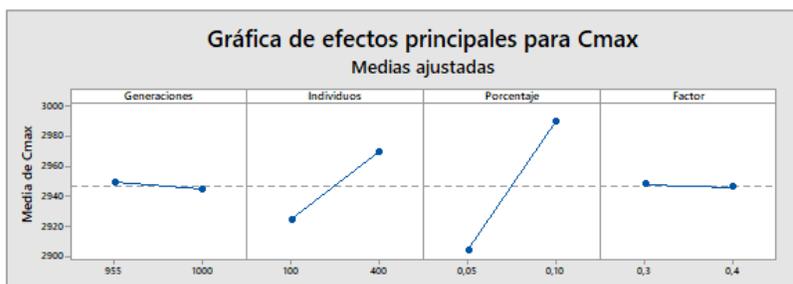


Figura 28. Grafica de efectos principales para las instancias de 50X5_6.

Como se aumentaron las generaciones, es como la que mayor influencia representa en el tiempo computacional. (Ver Figura 29).

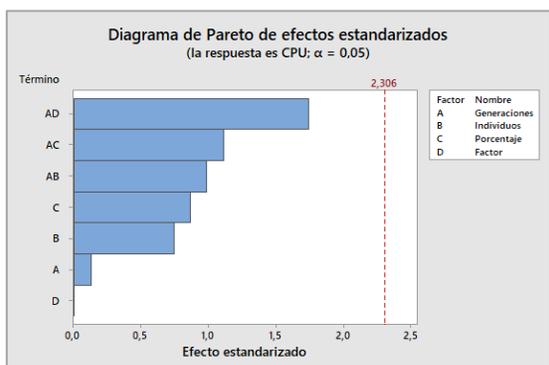


Figura 29. Diagrama de Pareto para la instancia 50X5_6

6.1.2. Resultados experimentales con tiempos de alistamiento.

En esta sección se evalúa las instancias propuestas en la literatura para determinar si el algoritmo presenta respuestas dentro del rango aprobado en las instancias que proponen los autores anteriormente mencionados.

6.1.2.1. Análisis de varianza para instancias SDST10.

Se ha seleccionado aleatoriamente la instancia SDST10, para evaluar la robustez del AG_UPB. Se han realizado 3 experimentos por cada instancia con las siguientes variables de entrada: Generaciones {354 – 950 – 1200}, Individuos {420 – 610 - 852}, Porcentaje {0,05 – 0,1}, Factor {0,2 – 0,3 – 0,4 – 0,7}.

6.1.2.2. Análisis de varianza para instancias Ta001_M1.

En la Tabla 29, se presenta el análisis de varianza del experimento, los efectos que tienen estimaciones grandes están incluidos en el modelo. (Ver Tabla 29).

Tabla 29. Análisis de varianza para la instancia Ta001_M1.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	7,6	7,6	0	0,949
Individuos	1	203,1	203,1	0,12	0,74
Porcentaje	1	7788,1	7788,1	4,55	0,066
Factor	1	1580,1	1580,1	0,92	0,365
Generación*Individuos	1	5967,6	5967,6	3,48	0,099
Generación*Porcentaje	1	14701,6	14701,6	8,58	0,019
Generación*Factor	1	1278,1	1278,1	0,75	0,413
Error	8	13704,5	1713,1		
Total	15	45230,4			

Para esta instancia se identifica que las generaciones y el porcentaje de mutación incide en el criterio de minimización del makespan, además se establece que la respuesta mejora cuando, los individuos son mayores que las generaciones. (Ver figura 30).

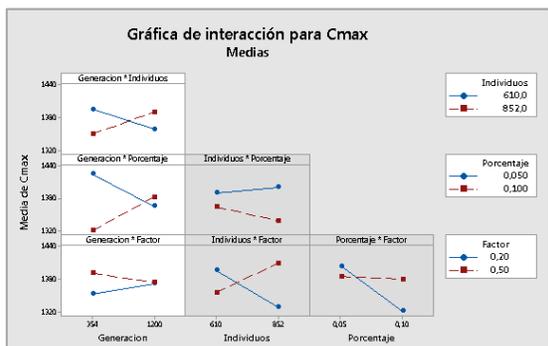


Figura 30. Interacciones para el C_{max} de la instancia Ta001_M1

Los efectos principales para estas instancias están dados por los siguientes. Donde los parámetros más recomendados para la ejecución del programa deben estar por debajo de la media. (Ver Figura 31).

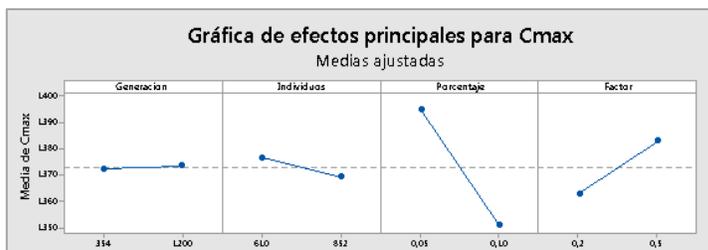


Figura 31. Efectos principales de la instancia Ta001_M1.

6.1.2.3. Análisis de varianza para instancias Ta003_M4.

En la tabla 30, se hace un análisis de varianza para 20 trabajos y 5 máquinas con tiempos de alistamiento. (Ver tabla 30).

Tabla 30. Análisis de varianza para la instancia Ta003_M4.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	2773,5	2773,5	0,99	0,335
Individuos	1	3037,5	3037,5	1,08	0,314
Porcentaje	1	20,2	20,2	0,01	0,933
Factor	1	280,2	280,2	0,1	0,756
Generaciones*Individuos	1	1120,7	1120,7	0,4	0,536
Generaciones*Porcentaje	1	1802,7	1802,7	0,64	0,435
Generaciones*Factor	1	12330,7	12330,7	4,39	0,052
Error	16	44898,7	2806,2		
Total	23	66264			

A pesar de agregar los tiempos de alistamiento a la secuencia, es posible identificar que el parámetro que tiene mayor influencia es el factor de mutación y los individuos que se ingresen por cada generación al igual que si los tiempos de alistamiento estuviesen en cero. (Ver figura 32).

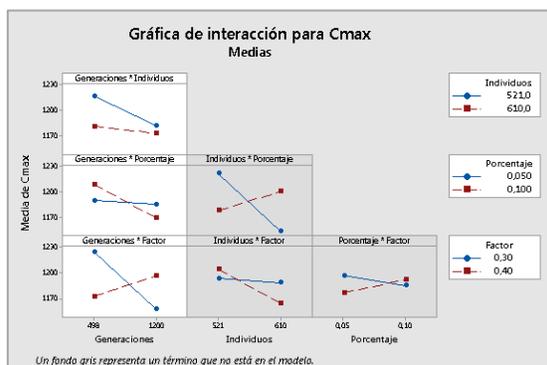


Figura 32. Interacciones para el C_{max} de la instancia Ta003_M4.

Además, se identifica que, al ingresar tiempos de alistamiento, la respuesta es efectiva al tener los parámetros propuestos por la sección anterior. Los efectos principales que se pueden ejecutar en esta instancia se demuestran en la siguiente figura. (Ver Figura 33).



Figura 33. Efectos principales de la instancia Ta003_M4.

6.1.2.4. Análisis de varianza para instancia Ta009_M4.

En la tabla 31, se analiza la varianza para 20 trabajos con 5 máquinas con tiempos de alistamiento. (Ver Tabla 31).

Tabla 31. Análisis de varianza para instancias Ta009_M4.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	7182,6	7182,56	3,32	0,106
Individuos	1	7,6	7,56	0	0,954
Porcentaje	1	3813,1	3813,06	1,76	0,221
Factor	1	5,1	5,06	0	0,963
Generaciones*Individuos	1	826,6	826,56	0,38	0,554
Generaciones*Porcentaje	1	410,1	410,06	0,19	0,675
Generaciones*Factor	1	6765,1	6765,06	3,12	0,115
Error	8	17329,5	2166,19		
Total	15	36339,4			

Al tener un factor mutación del 40%, un porcentaje de mutación de 1%, una reducción en las generaciones y un leve aumento en los individuos, el C_{max} mejora notablemente; por tanto, se puede inferir que no existe efectos significativos entre generaciones*porcentajes, porque los individuos se aumentan cada vez que se aumentan los tiempos de alistamiento. (Ver Figura 34).

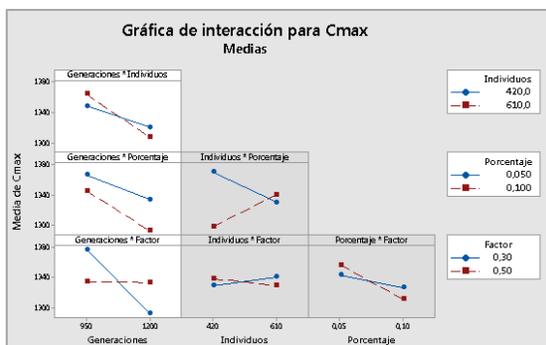


Figura 34. Interacciones para el C_{max} de la instancia Ta009_M4.

También se demuestra que, con los siguientes efectos es posible mejorar el tiempo de ejecución. (Ver Figura 35).



Figura 35. Efectos principales para el C_{max} de la instancia Ta009_M4.

Para el tiempo computacional se identifica que, al aumentar el factor de mutación se incrementa el nivel de ejecución del programa, sin embargo, para estabilizar es recomendable dejar un porcentaje de 5%; por otro lado, se evidencia que a mayores tiempos de alistamiento un factor mutación del 60% es lo ideal.

6.1.2.5. Análisis de varianza instancias Ta021_M0.

En tabla 32, se demuestra el análisis de varianza para 20 trabajos y 20 máquinas, con tiempos de alistamiento. (Ver Tabla 32).

Tabla 32. Análisis de varianza para la instancia Ta021_M0.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	702	702,3	0,06	0,082
Individuos	1	43472	43472,3	3,43	0,101
Porcentaje	1	12656	12656,2	1	0,047
Factor	1	3540	3540,2	0,28	0,011
Generaciones*Individuos	1	4489	4489	0,35	0,568
Generaciones*Porcentaje	1	11449	11449	0,9	0,369
Generaciones*Factor	1	3136	3136	0,25	0,632
Error	8	101257	12657,1		
Total	15	180702			

Al aumentar la cantidad de máquinas se presenta una mejora con los mismos parámetros del inciso 6.1.3.4, sin embargo, es recomendable dejar un factor mutante del 70% y un porcentaje de mutación del 5% para disminuir el tiempo computacional. (Ver Figura 36).

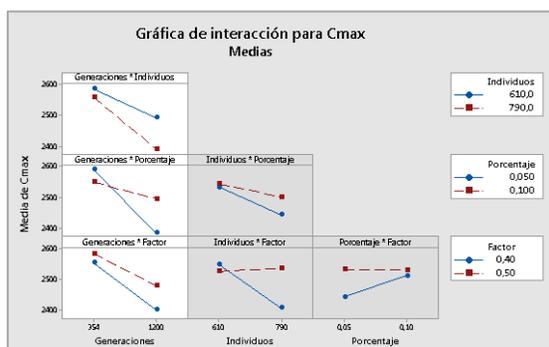


Figura 36. Interacciones para el C_{max} de la instancia Ta021_M0.

Los efectos principales que permiten mejorar el tiempo computacional en esta instancia se evidencian en la siguiente figura. (Ver Figura 37).

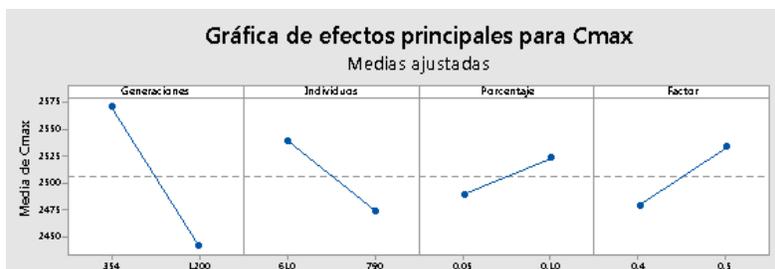


Figura 37. Efectos principales para el C_{max} de la instancia Ta021_M0.

6.1.2.6. Análisis de varianza para instancia Ta023_M12.

En la tabla 33, se demuestra el análisis de varianza para 20 trabajos y 20 máquinas, con tiempos de alistamiento. (Ver Tabla 33).

Tabla 33. Análisis de varianza para instancia Ta023_M12.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	324	324	0,03	0,876
Individuos	1	3906	3906,3	0,31	0,592
Porcentaje	1	1722	1722,3	0,14	0,021
Factor	1	5184	5184	0,41	0,038
Generaciones*Individuos	1	8010	8010,2	0,64	0,447
Generaciones*Porcentaje	1	14520	14520,3	1,16	0,313
Generaciones*Factor	1	17956	17956	1,43	0,266
Error	8	100306	12538,3		
Total	15	151929			

Como anteriormente se mencionó al aumentar la cantidad de máquinas, el CPU es más pausado, pero se puede solucionar con el aumento de individuos, por tal motivo el análisis arroja gran significancia sobre este valor. (Ver Figura 38).

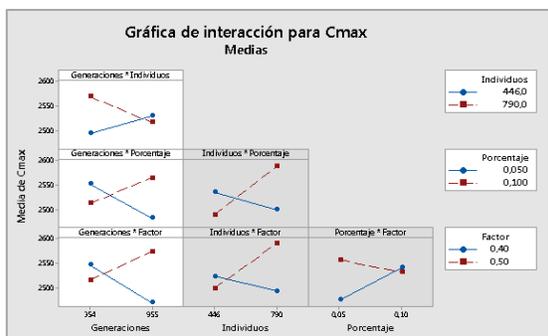


Figura 38. Interacciones de C_{max} para instancias Ta023_M12.

Los efectos principales que se recomienda en el programa para estas instancias se representan en la siguiente figura. (Ver figura 39).

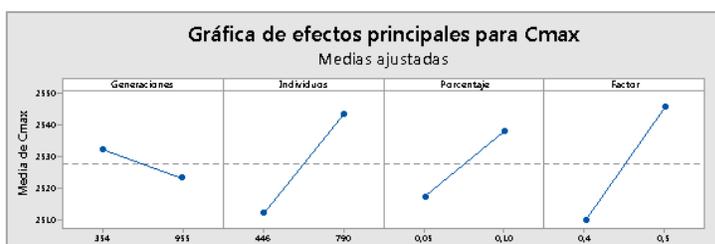


Figura 39. Efectos principales para el C_{max} de la instancia Ta023_M12.

6.1.2.7. Análisis de varianza para instancia Ta030_M9.

En la tabla 34, se demuestra el análisis de varianza para 20 trabajos y 20 máquinas, con tiempos de alistamiento. (Ver Tabla 34).

Tabla 34. Análisis de varianza para instancia Ta030_M9.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	43160	43160,1	5	0,056
Individuos	1	371	370,6	0,04	0,841
Porcentaje	1	10455	10455,1	1,21	0,303
Factor	1	39303	39303,1	4,55	0,065
Generaciones*Individuos	1	3938	3937,6	0,46	0,518
Generaciones*Porcentaje	1	6683	6683,1	0,77	0,405
Generaciones*Factor	1	13398	13398,1	1,55	0,248

Error	8	69053	8631,6
Total	15	186360	

. En este caso las generaciones y los individuos representan significancia, el motivo es que se dejan los parámetros que ya representan un equilibrio en el sistema y se empieza a evaluar los dos primeros factores. (Ver Figura 40).

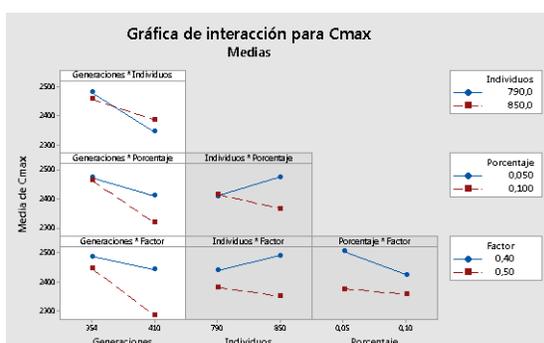


Figura 40. Interacciones para el C_{max} de la instancia Ta030_M9.

Los efectos principales para obtener la mejor respuesta, en esta instancia se representa en la siguiente figura, recordar que siempre deben estar por debajo de la media. (Ver Figura 41).

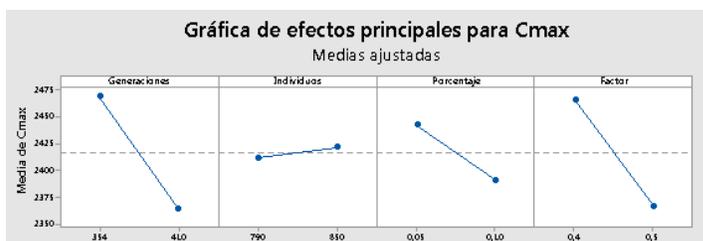


Figura 41. Efectos principales para la instancia Ta030_M9.

6.1.2.8. Análisis de varianza de instancia Ta032_M2.

En la tabla 35, se determina el análisis de varianza para 50 trabajos y 5 máquinas, con tiempos de alistamiento. (Ver Tabla 35).

Tabla 35. Análisis de varianza para la instancia Ta032_M2.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	29929	29929	2,73	0,137
Individuos	1	4900	4900	0,45	0,522
Porcentaje	1	8556	8556,2	0,78	0,402
Factor	1	90	90,3	0,01	0,039
Generaciones*Individuos	1	121	121	0,01	0,919
Generaciones*Porcentaje	1	32942	32942,3	3,01	0,121
Generaciones*Factor	1	2450	2450,2	0,22	0,649
Error	8	87582	10947,8		
Total	15	166571			

Al agregar trabajos y reducir la cantidad de máquinas, el factor de mutación del 40% ya no es conveniente, porque el tiempo computacional aumenta notablemente; se han tomado muestras con porcentajes y factores con bajos niveles y se comprueba que la respuesta y el CPU mejora. (Ver Figura 42).

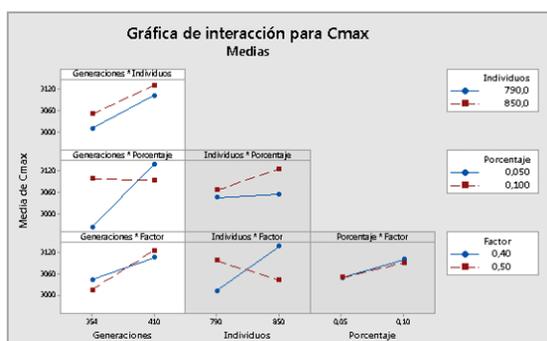


Figura 42. Interacciones para el C_{max} de la instancia Ta032_M2.

Los efectos principales que se recomiendan en la instancia se evidencian en la siguiente figura. (Ver Figura 43).

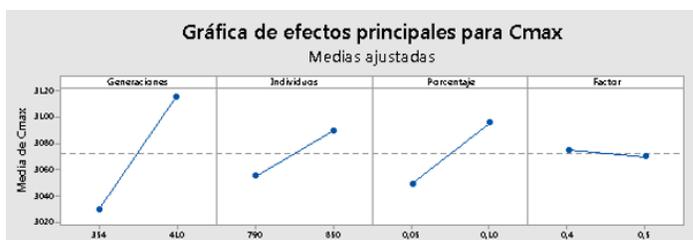


Figura 43. Efectos principales para la instancia Ta032_M2.

6.1.2.9. Análisis de varianza de instancia Ta034_M0.

En la tabla 26, se hace el análisis de varianza para 50 trabajos y 5 máquinas, con tiempos de alistamiento. (Ver Tabla 36).

Tabla 36. Análisis de varianza para la instancia Ta034_M0.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	4970,3	4970,3	1,35	0,279
Individuos	1	11881	11881	3,22	0,11
Porcentaje	1	24025	24025	6,51	0,034
Factor	1	462,3	462,3	0,13	0,732
Generaciones*Individuos	1	576	576	0,16	0,703
Generaciones*Porcentaje	1	4900	4900	1,33	0,282
Generaciones*Factor	1	3422,3	3422,3	0,93	0,364
Error	8	29503	3687,9		
Total	15	79739,8			

En esta instancia se verifico que, al aumentar los porcentajes y el factor, la respuesta tomara más tiempo, el efecto significativo se centra en el porcentaje mutación e individuos. (Ver Figura 44).

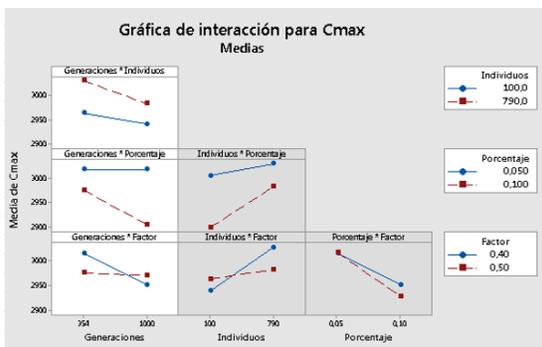


Figura 44. Interacción del C_{max} para la instancias Ta034_M0.

Al disminuir el factor mutante y el porcentaje, se pueden mejorar las respuestas y el tiempo computacional. (Ver Figura 45).

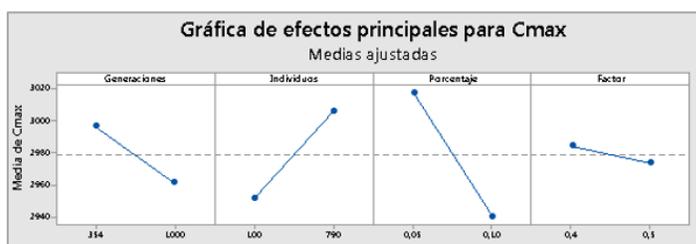


Figura 45. Efectos principales para la instancia Ta034_M0

6.1.2.10. Análisis de varianza de instancia Ta036_M6.

En la tabla 37, se presenta el análisis de varianza para 50 trabajos y 5 máquinas, con tiempos de alistamiento. (Ver Tabla 37).

Tabla 37. Análisis de varianza para instancia Ta036_M6.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Generaciones	1	4064	4064,1	0,43	0,531
Individuos	1	203	203,1	0,02	0,887
Porcentaje	1	4456	4455,6	0,47	0,013
Factor	1	4258	4257,6	0,45	0,522
Generaciones*Individuos	1	7439	7439,1	0,78	0,402
Generaciones*Porcentaje	1	5968	5967,6	0,63	0,451

Generaciones*Factor	1	431	430,6	0,05	0,837
Error	8	75991	9498,9		
Total	15	102809			

Podemos ver que a medida que disminuye la tasa de mutación, la efectividad del algoritmo genético aumenta hasta alcanzar mejores respuestas. (Ver Figura 46).

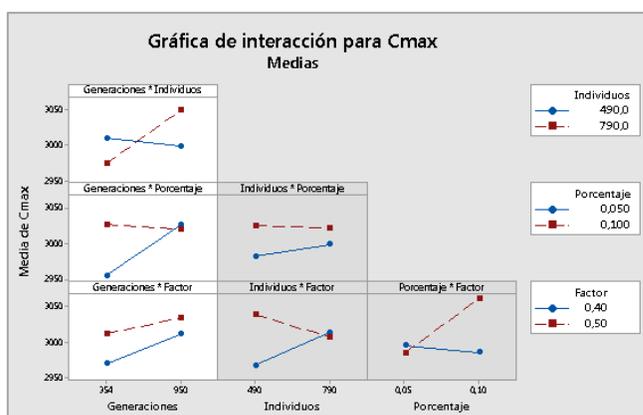


Figura 46. Interacción del C_{max} para la instancias Ta036_M6.

Finalmente se establece que los efectos principales para esta instancia se demuestran en la siguiente figura. (Ver Figura 47).



Figura 47. Efectos principales para la instancia Ta036_M6.

Después de realizar las pruebas computacionales en el programa, se analiza los límites del C_{max_Ub} y el C_{max_Lb} (Límites superiores e inferiores), donde se compara el algoritmo genético propuesto, con la metaheurística de Ruiz, Maroto y Alcaraz (2004) que se denotaran como

SDST_HGA, donde adaptan dos algoritmos genéticos con metaheurísticas existentes, para el algoritmo híbrido (HGA por sus siglas en inglés), Todas las pruebas fueron realizadas en un grupo de cuatro computadoras PC / AT Con Athlon XP 1600+ procesadores y 512 MBytes de memoria principal; por otro lado también se encuentra Taillard (1993), donde menciona en una de sus publicaciones los método heurísticos para dar solución a las instancias que propone. Como en los artículos se ha buscado encontrar la misma solución o mejor solución a las de referencia que son las de Taillard, en la siguiente ilustración se puede diferenciar en términos del makespan las instancias propuestas, donde las instancias SDST-Ta refieren a las de Taillard y las AG_UPB a las del proyecto de investigación de pregrado. (Ver Ilustración 22).

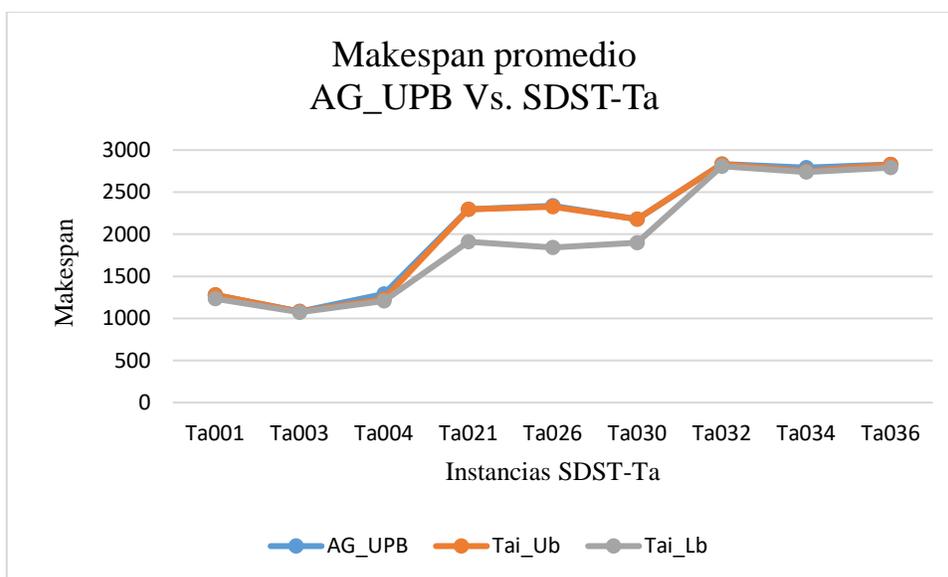


Ilustración 20. Makespan promedio por cada instancia AG_UPB Vs. SDST-Ta

Se puede observar que el algoritmo propuesto en el presente proyecto está por debajo de la instancia Tai_Ub. Las ejecuciones del AG_UPB se realizaron en un procesador Inter® Core™i3 – 2328M CPU de 2.20GHZ y memoria (RAM) de 4 GB.

7. CONCLUSIONES

Se identifico la dinámica que comprende la estructura algorítmica y los problemas de programación tipo Flowshop con restricciones de tiempos de alistamiento, lo cual permitió hacer uso de la metaheurística propuesta.

Se caracterizo la línea de producción que compone la HAS-200 del laboratorio de automatización de la Universidad Pontificia Bolivariana, donde se analizan los tiempos de procesamiento y alistamiento por cada combinación de los productos que la componen.

Se materializa la propuesta del diseño del algoritmo evolutivo, donde internamente fue modificado para la estructura de la celda de manufactura la HAS-200, que permite encontrar mejores soluciones de calidad con menores costos y tiempo computacional.

Se implemento un DOE, para instancias con tiempos de alistamiento en cero denominadas SDST-Ta e instancias con tiempos de alistamiento bautizadas SDST-HGA donde se puede afirmar con seguridad la robustez del AG propuesto en este proyecto de investigación, ya que presentan respuestas aproximadas o iguales al intervalo propuesto por los autores mencionados.

Se evidencia mediante el diseño experimental las variables de entradas idóneas para el AG_UPB, donde se demuestra que la variable más significativa se nombra Factor_mutación, seguidamente por el Porcentaje_mutación ya que estos factores se encargan de dar el equilibrio al sistema.

Se observo que al incrementar los tiempos de alistamiento es más conveniente aumentar el factor mutación, para mejorar la calidad de respuesta, en cuanto al tiempo computacional es recomendable reducir el porcentaje de mutación, para disminuir su tiempo de ejecución.

Se acredita que el modelo metaheurístico para el problema de secuenciación de producción bajo ambiente de manufactura Flowshop con restricciones de tiempo de alistamiento en

secuencia en el sistema de producción HAS-200 de la Universidad Pontificia Bolivariana obtiene mejoras en tiempos computacionales eficientes, puede procesar las respuestas propuestas por los autores en las instancias a pesar de ser ejecutado en un interfaz sencillo, además la solución que brinda el algoritmo para la HAS-200 del laboratorio de automatización mejora a un 46%.

Para que el modelo obtenga un mayor rendimiento en tiempo computacional y calidad de solución al problema con tiempos de alistamiento se sugiere, un porcentaje de mutación del 5%, factor mutación en un rango del 50% - 70%, con un 70% disminuye el tiempo de ejecución, pero si los tiempos de alistamiento están en cero se recomienda un porcentaje de mutación del 10% y un factor de mutación del 40% por último, las generaciones deben ser menores que los individuos.

8. RECOMENDACIONES

Se recomienda al aspirante investigador tener conocimientos previos en la programación, para facilitar el proceso de la estructuración algorítmica.

Se sugiere implementar otro método metaheurístico para las HAS-200, ya que, sobre la base de este, se pueden llegar a mejores conclusiones y mejoramiento continuo.

Es necesario analizar los recursos disponibles tanto para la fase de investigación como la fase experimental, ya que un procesador de poca memoria dificulta el muestreo.

REFERENCIAS

- Abellanas, M., Lodares, D. (1991). Análisis de algoritmos y teoría de grafos. México D.F: Editorial Macrobit Editores.
- Alba, E, Laguna., Martí, R. (2003). Métodos Evolutivos. España. <https://www.uv.es/rmarti/paper/docs/heur3.pdf>
- Alberte, A, Málaga, R., Ibáñez, A. (2004). Implementación de Juegos usando Algoritmos Evolutivos. Madrid. España. <https://eprints.ucm.es/9144/1/TC2004-4.pdf>
- Baker, K., Trietsch, D., (2006). Principles of sequencing and scheduling. John Wiley & Sons, Inc., Hoboken, New Jersey.
- Brito, S. J., Campos, R. C., García, L. C., García, T. M., Mellan, B. B., Moreno, P. J., Moreno, V. J. (2004). Metaheurísticas: una revisión actualizada. Departamento de Estadística, Investigación Operativa y Computación. Universidad de La Laguna. 38271 La Laguna. <http://jamoreno.webs.ull.es/www/papers/paper37.pdf>
- Brucker P., (2006), Scheduling Algorithms. Berlin, Germany: Springer.
- Damodaran, S. K., & Sunil Kumar, T. K. (2018). Hydro-Thermal-Wind Generation Scheduling Considering Economic and Environmental Factors Using Heuristic Algorithms. *Energies* (19961073), 11(2), 1-19. doi: 10.3390/en11020353
- Daniels, R. L., & Mazzola, J. B. (1993). A tabu-search heuristic for the flexible-resource flow shop scheduling problem. *Annals of Operations Research*, 41(1-4), 207-230.
- De Lima, J. F. (2010). Metaheuristics Using Reinforcement Learning Applied to Salesman Traveling Problem and Applications, December 2010. P 213-236. ISBN 978-953-307-426-9.
- Flores, M, A. (2012). Optimización en la entrega de productos para una cadena de abastecimiento. : Determinación de Rutas de Entregas. Universidad Nacional Mayor de San Marcos. Perú. Recuperado de: http://sisbib.unmsm.edu.pe/bibvirtualdata/tesis/ingenie/flores_tm/cap3.pdf
- Frank, S., Steponavice, I., Rebennack, S. (2012) Optimal power flow: a bibliographic survey I, *Energy Syst.* 3 (September (3)) (2012) 221–258
- Gamboa, J, M. Gómez, A, J. (s.f). Formulación de un algoritmo genético para el problema de programación de órdenes de trabajo de una empresa de artes gráficas. Bogotá. Recuperado de: http://www.konradlorenz.edu.co/images/investigaciones/matematicas/proyecto_de_grado_ag1.pdf

- Graham, R. (1966). Bounds for certain multiprocessor anomalies. *Bell System Technical Journal*, 45(9):156-1581.
- Galera Prieto, M. (2017). Algoritmos para la programación de la producción en un entorno de flujo regular distribuido de permutación. (Trabajo Fin de Grado Inédito). Universidad de Sevilla, Sevilla.
- Gonzales., T., (Ed). (2007). Handbook of approximation algorithms and metaheuristics. Universidad de California. U.S.A: Champman & Hall/CRC Includes bibliographical references and index. ISBN-13: 978-1-58488-550-4 ISBN-10: 1-58488-550-5
- Gupta, J. D., & Maykut, A. R. (1973). Flow-shop scheduling by heuristic decomposition. *International Journal Of Production Research*, 11(2), 105.
- Glass, C. A., & Potts, C. N. (1996). A comparison of local search methods for flow shop scheduling. *Annals Of Operations Research*, 63(1-4), 489-509.
- Hayes, R.H. & Wheelwright, S.C. (1979). "Link Manufacturing Process and Product Life Cycles". En: *Harvard Business Review*, January February. pp. 133-140.
- Hillier, F., Lieberman, G. *Introducción a la investigación de operaciones*. Mexico: The McGraw-Hill Companies, Inc.
- Hein, F., Almeder, C., Figueira, G., & Almada-Lobo, B. (2018). Designing new heuristics for the capacitated lot sizing problem by genetic programming. *Computers and Operations Research*, 96, 1-14. doi:10.1016/j.cor.2018.03.006
- Holland-I. H. (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbo
- Honghan Ye, Wei Li, Enming Miao. An effective heuristic for no-wait flow shop production to minimize makespan, *Journal of Manufacturing Systems*, Volume 40, Part 2, 2016, Pages 2-7, ISSN 0278-6125, <https://doi.org/10.1016/j.jmsy.2016.05.001>. (<http://www.sciencedirect.com/science/article/pii/S0278612516300140>) Keywords: Makespan minimization; No-wait flow shop; Heuristics; Computational complexity
- Hwang Lioa, Ochiny Burne, (2007). *la recherche algorithmique* Faculté Universitaire Catholique de Mons, Mons, Belgium Institut d'Economie Scientifique et de Gestion, Lille, France.
- Ignall, E., & Schrage, L. (1965). Application of the branch and bound technique to some flow-shop scheduling problems. *Operations Research*, 13(3), 400.
- Jian Lin, Shuai Zhang (2016). "An effective hybrid biogeography-based optimization algorithm for the distributed assembly permutation flow-shop scheduling problem". *Computers & Industrial Engineering*, 97, 128-136.

- Laribi, I., Yalaoui, F., Belkaid, F., & Sari, Z. (2016). Heuristics for solving flow shop scheduling problem under resources constraints. *IFAC-PapersOnLine*, 49(12), 1478-1483. <https://doi.org/10.1016/j.ifacol.2016.07.780>
- López, J. C., Giraldo, J. A., & Arango, J. A. (2015). Reducción del Tiempo de Terminación en la Programación de la Producción de una Línea de Flujo Híbrida Flexible (HFS). *Información Tecnológica*, 26(3), 157-172. Doi: 10.4067/S0718-07642015000300019
- Luan, L. (2008). Diseño de un sistema para la optimización de la planificación logística en una empresa constructora. Universidad de Chile. Santiago de Chile. Recuperado en: http://www.tesis.uchile.cl/tesis/uchile/2008/lee_cl/sources/lee_cl.pdf
- Manual de usuario y prácticas. (2017). Manual de usuario y prácticas de la HAS200. Recuperado en: https://upbeducomy.sharepoint.com/personal/leidy_olarte_upb_edu_co/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fleidy_olarte_upb_edu_co%2FDocuments%2FSAI5085_ManualUsuario%26PracticasHAS-200_ES_WEB%2Epdf&parent=%2Fpersonal%2Fleidy_olarte_upb_edu_co%2FDocuments&slid=9b1d7f9e-9094-6000-afed-ea2736275a76
- Mercado, R. Z. R., & Bard, J. F. (2001). Secuenciando óptimamente líneas de flujo en sistemas de manufactura, *IV*(10), 38–45.
- McMahon, G. B. (1969). Optimal production schedules for flow shops. *CORS Journal*, 7(2), 141-151.
- Michalewicz, Z. (1999). *Genetic Algorithms + Data Structures = Evolution Programs*. Editorial Springer. ISBN 3-540-60676-9 Springer-Verlag Berlin Heideberg New York. Disponible en: <http://web.ist.utl.pt/adriano.simoes/tese/referencias/Michalewicz%20Z.%20Genetic%20Algorithms%20+%20Data%20Structures%20=%20Evolution%20Programs%20%283ed%29.PDF>
- Morillo, D., Moreno, L., Díaz, J. (2013). Metodologías Analíticas y Heurísticas para la Solución del Problema de Programación de Tareas con Recursos Restringidos (RCPS): una revisión. *Ingeniería y Ciencia*, 10(19), 11-20. doi:10.17230/ingciencia.10.19.12
- Mosheiov, G., Oron, D., & Ritov, Y. (2004). Flow-shop batching scheduling with identical processing-time jobs. *Naval Research Logistics*, 51, 783–799
- Morton, T. E y Pentico, D., *Heuristic Scheduling Systems*, Jon Wiley & Sons, Nueva York, 1993.
- Muñoz, J. A. (2017). Restricciones de capacidad, exportación e I+D: Un análisis empírico con datos de empresas manufactureras españolas. *Estudios De Economía Aplicada*, 35(1), 133-151.

- Noor, S., Ikram, M., Ullah, I y Saqib, N. (2015). Solving Jobshop Scheduling problem with genetic Algorithm. *Sclint*. 27(4),3367-3371
- Patro, R., Suero, D., Guzman, O. (2015). Ruteo de vehículos desde un centro de distribución a una línea de supermercados en Barranquilla, Colombia. Colombia. Universidad Libre de Barranquilla. Recuperado en: <http://www.unilibrebaq.edu.co/ojsinvestigacion/index.php/ingeniare/article/view/423>
- Perea., V. (2015). Diseño de un procedimiento basado en algoritmos genéticos para dar solución al problema de scheduling admitiendo rechazos en escenarios flexible flow shop. Bogotá. Colombia. Universidad Distrital Francisco José de Caldas. <http://repository.udistrital.edu.co/bitstream/11349/2785/1/PereaGonz%C3%A1lezVivianaAlexandra2015.pdf>
- Pérez, C. (2016). Problemas de Rutas de Vehículos algunas de sus variantes más conocidas. Universidad de Laguna. Recuperado en: <https://riull.ull.es/xmlui/bitstream/handle/915/3046/Problemas%20de%20Rutas%20de%20Vehiculos.%20Algunas%20de%20sus%20variantes%20mas%20conocidas..pdf?sequence=1>.
- Pérez, O. (2009). Elaboración de un cuadernillo de apuntes: Sistemas de Manufactura. México. Recuperado en: <http://www.tesoem.edu.mx/alumnos/cuadernillos/2009.020.pdf>
- Producción industrial colombiana cayó 0,6% en el 2017. (2018). *Revista Dinero*. Recuperado en: <https://www.dinero.com/economia/articulo/produccion-industrial-en-colombia-2017/255294>.
- Phanden, Rakesh. (2016). Multi Agents approach for Job Shop Scheduling Problem using Genetic Algorithm and Variable Neighborhood Search method. https://www.researchgate.net/publication/321807729_Multi_Agents_approach_for_Job_Shop_Scheduling_Problem_using_Genetic_Algorithm_and_Variable_Neighborhood_Search_method.
- Priale. I (2012). *Investigación de operaciones*, 9na. Edición – Hamdy A. Taha | FreeLibros. Recuperado 28 de agosto de 2016, a partir de <http://www.freelibros.org/investigacion-de-operaciones-9na-edicion-hamdy-a-taha.html>
- Quispe., R. (2007) ¿Que es un Algoritmo? Blog de Rodolfo Quispe-Otazu. Disponible en: <http://www.rodolfoquispe.org/blog/que-es-un-algoritmo.php>
- Ribas, I., Companys, R., & Tort-Martorell, X. (2013). An efficient iterated local search algorithm for the total tardiness blocking flow shop problem. *International Journal Of Production Research*, 51(17), 5238-5252. doi:10.1080/00207543.2013.802390.
- Ruiz, R., Maroto, C., & Alcaraz, J. (2005). Solving the flowshop scheduling problem with

- sequence dependent setup times using advanced metaheuristics. *European Journal of Operational Research*, 165(1), 34–54. <https://doi.org/10.1016/j.ejor.2004.01.022>
- Ruiz, Rubén , Stützle, T. (2006). An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *Industrial and Engineering Chemistry Product Research and Development*, 19(4), 551–556. <https://doi.org/10.1016/j.ejor.2006.07.029>
- Salazar, H, E. Medina, S, J, C. (2012). Minimización del makespan en máquinas paralelas idénticas con tiempos de preparación dependientes de la secuencia utilizando un algoritmo genético. *Ingeniería Investigación y Tecnología*, volumen XIV (número1). Recuperado en: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-77432013000100005
- Seck, J., Medina. J.,Hernández. (2016). Introducción a los Algoritmos Genéticos con Matlab. México. Universidad del estado de hidalgo. https://www.uaeh.edu.mx/docencia/P_Lectura/icbi/ asignatura/introduccion_a_los_algoritmos_geneticos_con_matlab.pdf
- Sippper, D., Bulfin, R. (1998). Planeación y control de la producción -- México: McGraw-Hill, c1998
- Smits, A. M., & Baker, K. R. (1981). An experimental investigation of the occurrence of special cases in the three-machine flowshop problem. *International Journal of Production Research*, 19(6), 737.
- Taillard. E. (1993). Benchmarks for basic scheduling problems, *European Journal of Operational Research*, EJOR 64(2):278-285, 1993. Disponible en: <http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>
- Taha. H (2012) *Investigación de operaciones*. 9 edición.(p.1-8), PEARSON EDUCATION, Mexico: <https://jrvargas.files.wordpress.com/2009/01/investigacic3b3n-de-operaciones-9na-edicic3b3n-hamdy-a-taha-fl.pdf>
- Talbi, E. G. (2009). Metaheuristics from design to implementation. Editorial Wiley. New Jersey
- Vidal, E. A. (2013). Optimización de algoritmos heurísticos. Santiago. Universidad de Santiago de Compostela. http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_782.pdf
- Vega, C. A., & Caballero, J. P. (2010). Combined use of GRASP and path-relinking during production scheduling to minimize total weighted tardiness in a machine. [Uso combinado de GRASP y Path-Relinking en la programación de producción para minimizar la tardanza total ponderada en una máquina] *Ingeniería y Universidad*, 14(1), 79-96. Retrieved from www.scopus.com
- Ying, K. C., & Lin, S. W. (2018). Minimizing makespan for no-wait flowshop scheduling problems with setup times. *Computers and Industrial Engineering*, 121(May), 73–81.

<https://doi.org/10.1016/j.cie.2018.05.030>.

ANEXOS

Anexo A. Parametros de la HAS-200.

En la tabla 38, se muestra los parametros que se establecen en la HAS-200. (Ver Tabla 38).

Tabla 38. Variables de entrada para el AG de la HAS-200.

Generaciones	354
Individuos	790
Porcentaje Mutación	0,1
Factor Mutación	0,4
Replicas	6

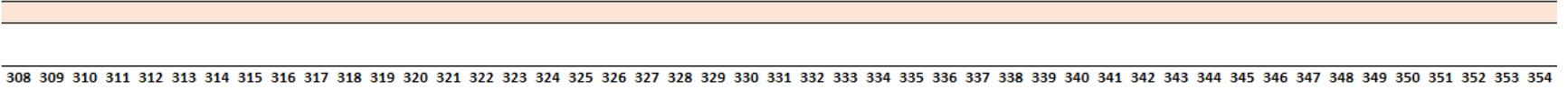
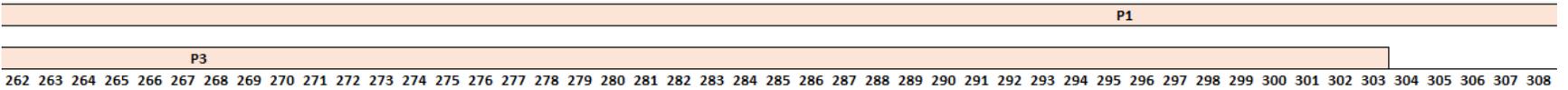
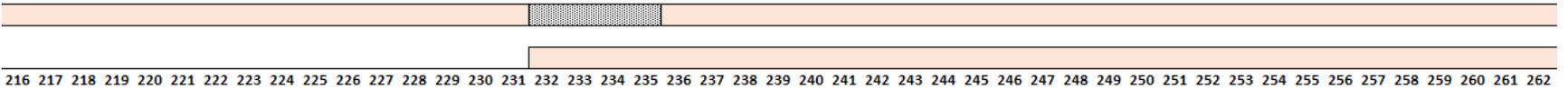
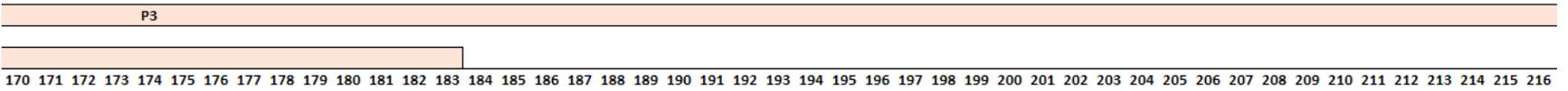
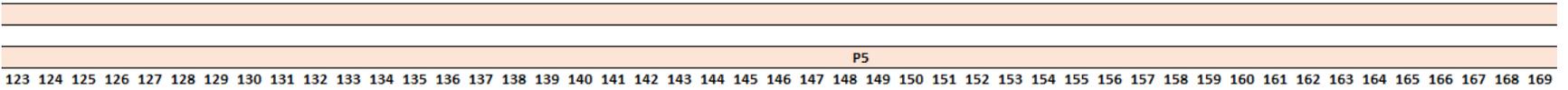
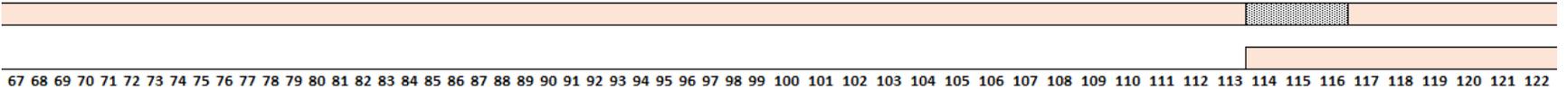
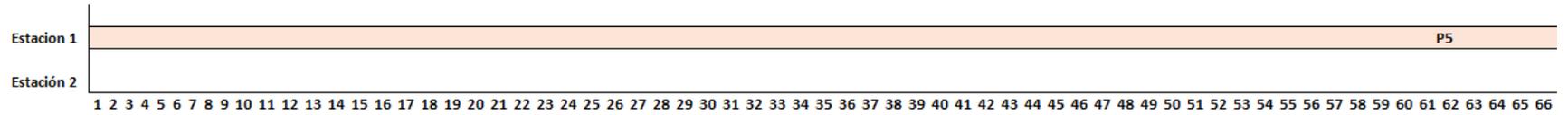
Anexo B. Secuenciación de la HAS-200.

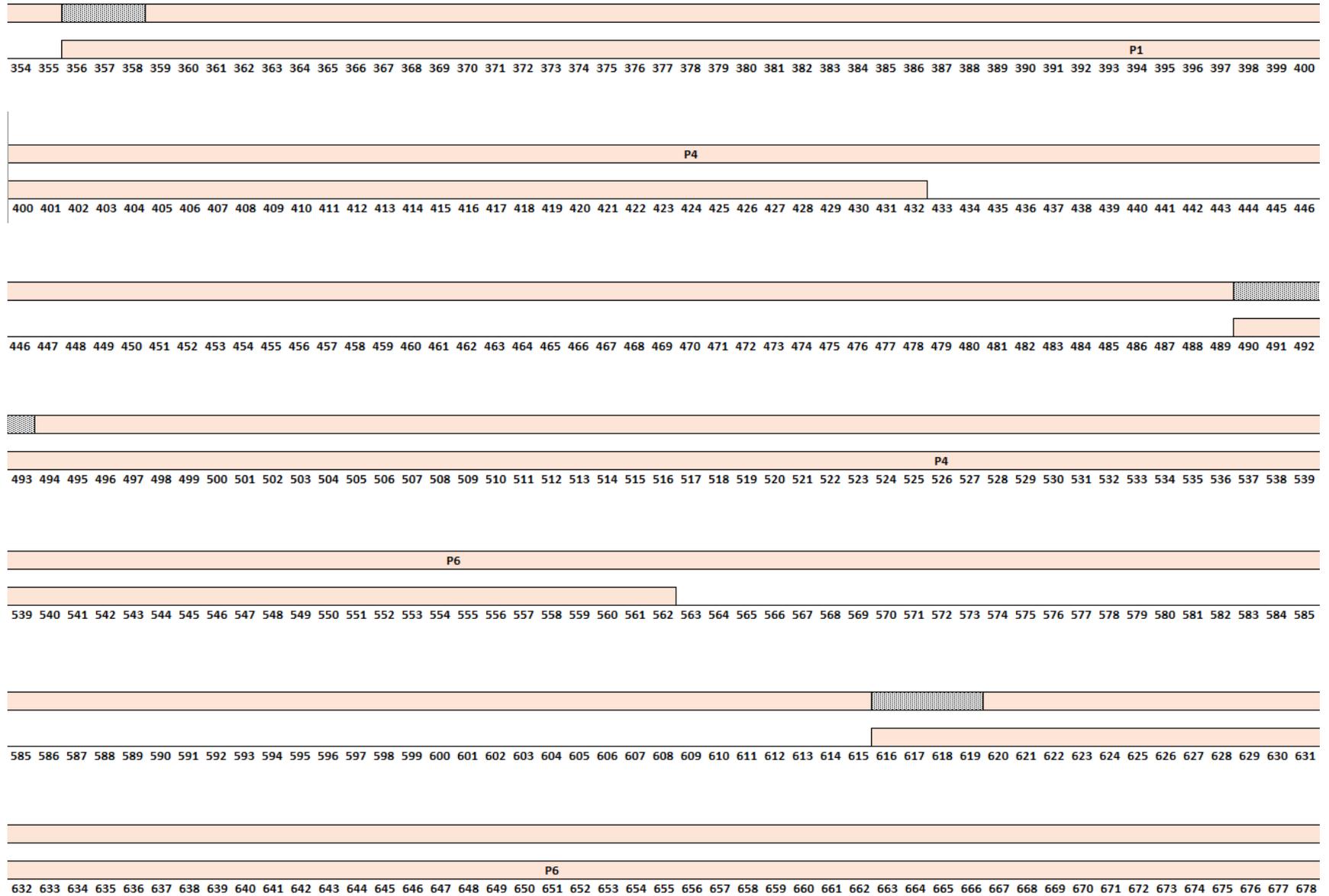
En la tabla 39, se evidencia la secuenciación inicial en la HAS-200 y la secuencia que recomienda el programa para mejorar el tiempo del proceso. (Ver Tabla 39).

Tabla 39. Secuenciación propuesta para la HAS-200.

Secuencia inicial	5	3	1	4	6	2
Mejor Secuencia	4	2	3	6	1	5

Se hace la ilustración de cada situación por medio de un diagrama de gantt donde se expone los tiempos de programación de los seis productos, estos tiempos tienen una unidad en minutos. (Ver Figura 48 y 49).





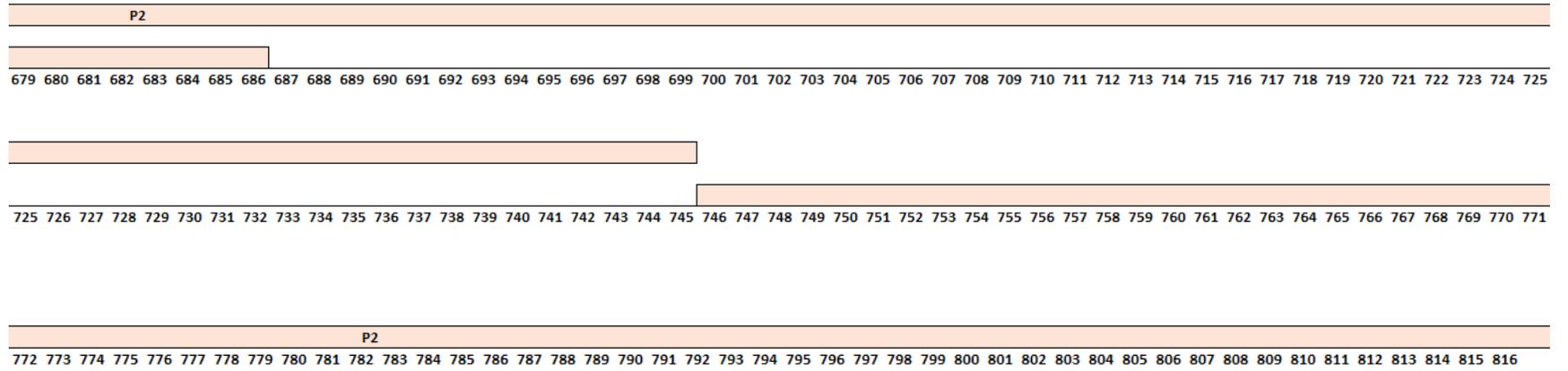
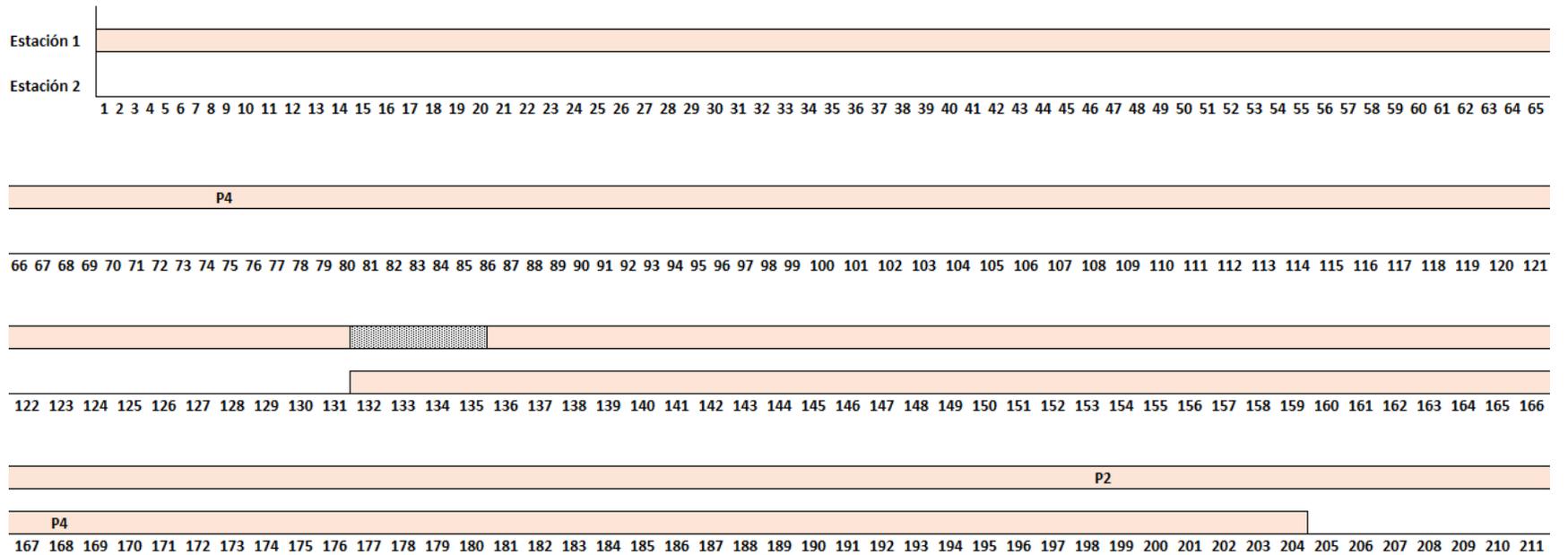
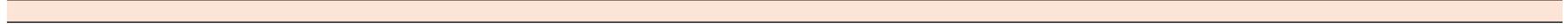


Figura 48. Diagrama de Gantt inicial de la HAS-200.





212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256

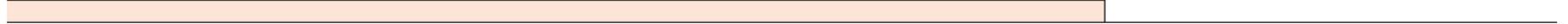


257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301

P2



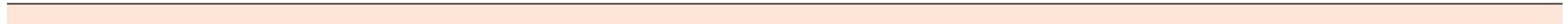
P3



302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346

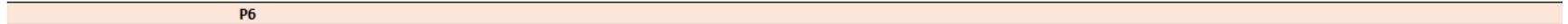


347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391



P3

392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436



P6



437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481



481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525



P1



P6

526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570

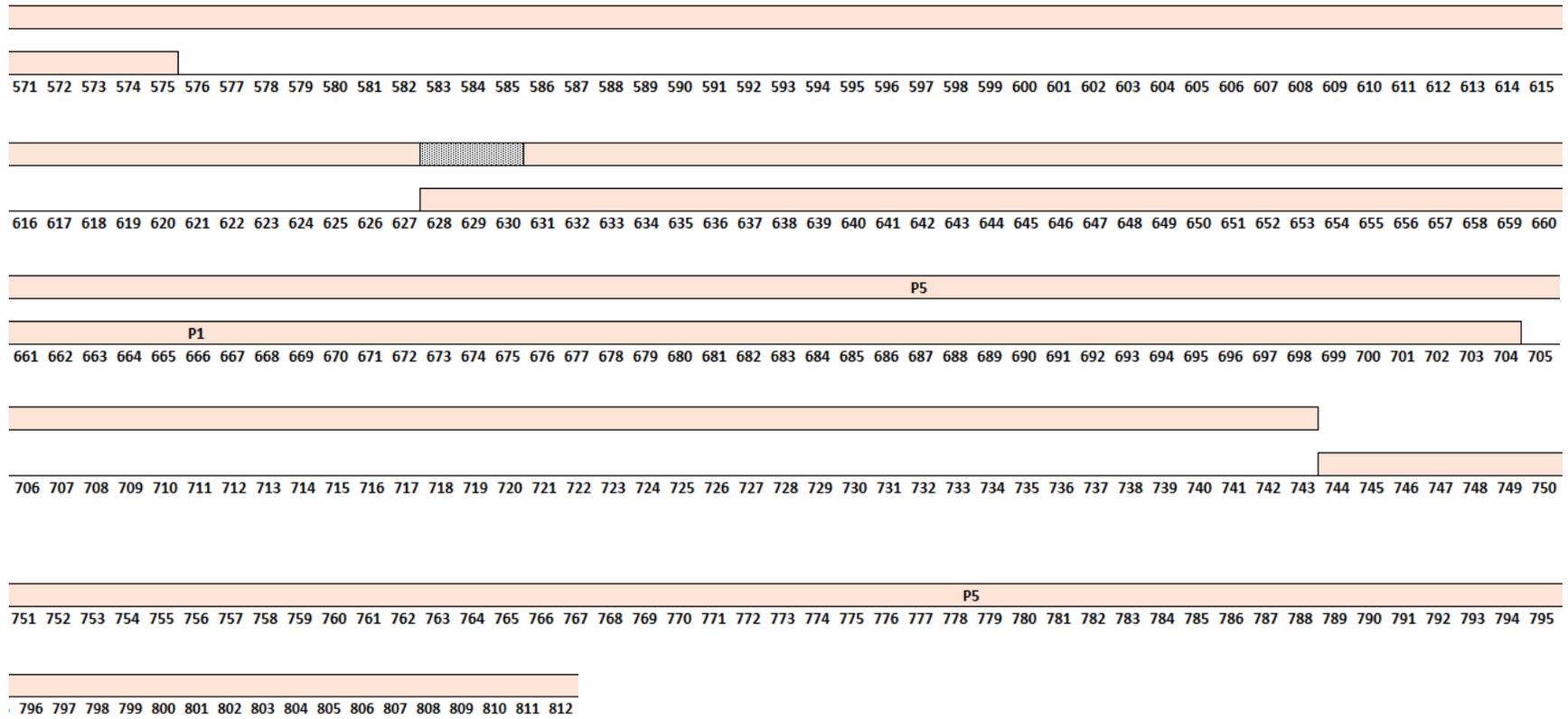


Figura 49. Diagrama de Gantt situación actual de la HAS-200.