

**METAHEURÍSTICA BASADA EN EL ALGORITMO COMPETITIVO
IMPERIALISTA (ICA) APLICADA A LA SOLUCIÓN DEL PROBLEMA DE *FLOW*
SHOP HÍBRIDO (HFS) CON MÁQUINAS PARALELAS NO RELACIONADAS**

**ESLENDIS MORATTO CHIMENTY
LUIS EDUARDO PÉREZ FIGUEREDO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
BUCARAMANGA**

2016

**METAHEURISTICA BASADA EN EL ALGORITMO COMPETITIVO
IMPERIALISTA (ICA) APLICADA A LA SOLUCIÓN DEL PROBLEMA DE *FLOW
SHOP HÍBRIDO (HFS)* CON MÁQUINAS PARALELAS NO RELACIONADAS**

**ELENDIS MORATTO CHIMENTY
LUIS EDUARDO PÉREZ FIGUEREDO**

Trabajo de Grado para optar el título de Ingeniero Industrial

Director

ELIANA MARCELA PEÑA TIBADUIZA

M.s.c. Ingeniería Industrial

Codirector

EDWIN ALBERTO GARAVITO HERNÁNDEZ

Ingeniero Industrial

Esp. Producción - Mejoramiento continuo.

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
BUCARAMANGA**

2016

AGRADECIMIENTOS

A Dios por darnos la sabiduría y permitirnos culminar esta etapa de nuestras vidas.

A nuestros padres por su esfuerzo, dedicación y apoyo incondicional durante toda nuestra formación universitaria.

A la Universidad Industrial de Santander por permitirnos ser parte de esta gran familia, y por todos los espacios académicos y culturales que contribuyeron a nuestra formación y crecimiento personal.

A los profesores Eliana Peña y Edwin Garavito por brindarnos el apoyo, orientación y acompañamiento en todo este proceso de aprendizaje, el cual fue clave en el desarrollo de la presente investigación.

Al profesor Rolando Acosta por su asesoría y disposición para resolver nuestras inquietudes en aspectos claves del proyecto.

A Daniel Martínez, Karin Aguilar y Tatiana Rodríguez por su apoyo en el desarrollo de esta investigación. Asimismo al grupo de investigación Ópalo por brindarnos los espacios y abrirnos sus puertas.

A Paola y Alejandra (las gemelas), María Alejandra, Sandra, Nidia, Laura, Angie, Yessica y Diego Alejandro por ofrecernos su amistad y aconsejarnos en momentos difíciles.

Quiero dedicar este trabajo a todos aquellos que han contribuido a su desarrollo, en primer lugar a Dios, por guiarme y protegerme en todo momento, dándome las fuerzas necesarias para afrontar el reto.

A mis padres Ramiro Moratto y Nurleydis Chimenty, quienes siempre han confiado en mis capacidades y me han facilitado la mejor formación a su alcance.

A mis hermanas Sulay, Margie y Karen y a mis abuelas Genny y Amilde quienes siempre estuvieron allí para darme su apoyo incondicional en este proceso.

A David Espejo por creer en mí y acompañarme en todo este tiempo permitiéndome compartir este logro con él.

A mi familia y amigos por brindarme palabras de aliento y esperanza.

Eslendis Moratto Chimenty

A Dios, por su infinito amor y por todas las bendiciones que me ha dado, gracias a su ayuda es que fue posible alcanzar esta meta. A ÉL sea toda la gloria.

A mi Madre, mi principal motivación, mi ejemplo de vida y mi tesoro más preciado, por su amor, apoyo incondicional, por todo el esfuerzo que hizo para que esta meta hoy sea realidad y por enseñarme y demostrarme que a pesar de que las cosas parezcan difíciles de alcanzar, con esfuerzo y disciplina todo es posible, a ella dedico éste y todos los triunfos venideros.

A Isabella, mi “hermanita”, por llenar con su amor y alegría mi hogar.

A mi familia por siempre estar a mi lado apoyándome y motivándome en mi proceso de formación universitaria.

A mis amigos: Andrea Carolina, Diego P., Byron, Keyla, Jhon Hernandez, Alberto, María Alejandra Ana Mercedes y Eslendis por permitirme compartir con ustedes estos años, por estar conmigo en los momentos de dificultad y en los momentos felices y por enseñarme tantas cosas.

Luis Eduardo Pérez Figueredo

CONTENIDO

	Pág.
INTRODUCCIÓN	20
1. GENERALIDADES DEL PROYECTO	23
1.1. PLANTEAMIENTO DEL PROBLEMA	23
1.2. JUSTIFICACIÓN DEL PROYECTO	24
1.3. OBJETIVOS	27
1.3.1. Objetivo general	27
1.3.2. Objetivos específicos.....	27
2. MARCO TEÓRICO.....	28
2.1. OPTIMIZACIÓN	28
2.2. COMPLEJIDAD COMPUTACIONAL	30
2.3. MÉTODOS DE SOLUCIÓN	32
2.3.1. Algoritmos exactos.	32
2.3.2. Algoritmos heurísticos.	33
2.3.3. Algoritmos Metaheurísticos.	35
2.3.4. Metaheurísticas utilizadas para solucionar problemas de optimización combinatoria.	37
2.3.5. El algoritmo competitivo imperialista (Imperialist Competitive Algorithm, ICA).....	41
2.4. PROGRAMACIÓN DE OPERACIONES	47
2.4.1. Clases de Schedule.	47
2.4.2. Tipos de problema de <i>scheduling</i>	48
2.5. FLOW SHOP HÍBRIDO.....	50
3. REVISIÓN DE LA LITERATURA	53
4. DISEÑO DEL MODELO MATEMÁTICO	61
4.1. DESCRIPCIÓN DEL SISTEMA PRODUCTIVO BAJO ESTUDIO.....	61
4.2. MODELO MATEMÁTICO.....	62
4.2.1. MODELO DE PROGRAMACIÓN LINEAL ENTERA MIXTA.....	63
4.3. ANÁLISIS DE RESULTADOS.....	67

5.	DISEÑO DEL ALGORITMO	70
5.1.	DESCRIPCIÓN DETALLADA DEL ALGORITMO	70
5.1.1.	Generación de la población inicial.....	70
5.1.2.	Movimiento de las colonias hacia su imperialista.....	75
5.1.3.	Información compartida entre colonias.....	76
5.1.4.	Revolución.....	77
5.1.5.	Guerra entre imperios.....	79
6.	DISEÑO EXPERIMENTAL PARA LA DETERMINACIÓN DE LOS FACTORES INCIDENTES EN EL DESEMPEÑO DEL ALGORITMO.....	82
6.1.	FACTORES PRINCIPALES.....	82
6.2.	ANÁLISIS DE LOS EFECTOS PRINCIPALES DE LOS PARÁMETROS DEL ALGORITMO.	84
6.2.1.	Análisis de varianza para la instancia k5m2i3	84
6.2.2.	Análisis de varianza para la instancia k5m2i5	86
6.2.3.	Análisis de varianza para la instancia k5m3i3	87
6.2.4.	Análisis de varianza para la instancia k5m3i5	89
6.2.5.	Análisis de varianza para la instancia k7m2i3	90
6.2.6.	Análisis de varianza para la instancia k7m2i5	92
6.2.7.	Análisis de varianza para la instancia k7m3i3	94
6.2.8.	Análisis de varianza para la instancia k7m3i5	96
6.2.9.	Análisis de varianza para la instancia k9m2i3	97
6.2.10.	Análisis de varianza para la instancia k9m2i5.....	99
6.2.11.	Análisis de varianza para la instancia k9m3i3.....	101
6.2.12.	Análisis de varianza para la instancia k9m3i5.....	102
7.	VALIDACIÓN DEL ALGORITMO	105
7.1.	ANÁLISIS DE RESULTADOS.....	106
8.	CONCLUSIONES.....	120
9.	RECOMENDACIONES	122
	BIBLIOGRAFIA.....	123
	ANEXOS.....	132

LISTA DE TABLAS

	Pág.
Tabla 1. Porcentaje soluciones óptimas y gap.....	68
Tabla 2. Tiempos (u.t) de procesamiento para k5m2i3.....	71
Tabla 3. Tiempo (u.t) de alistamiento para k5m2i3.....	72
Tabla 4. Cálculo del costo de un país.....	73
Tabla 5. Combinaciones de los tratamientos del diseño factorial 2^{5-1}	83
Tabla 6. ANOVA para k5m2i3.....	85
Tabla 7. ANOVA para k5m2i5.....	87
Tabla 8. ANOVA para k5m3i3.....	88
Tabla 9. ANOVA para k5m3i5.....	90
Tabla 10. ANOVA para k7m2i3.....	91
Tabla 11. ANOVA para k7m2i5.....	93
Tabla 12. ANOVA para k7m3i3.....	95
Tabla 13. ANOVA para k7m3i5.....	97
Tabla 14. ANOVA para k9m2i3.....	98
Tabla 15. ANOVA para k9m2i5.....	100
Tabla 16. ANOVA para k9m3i3.....	102
Tabla 17. ANOVA para k9m3i5.....	103
Tabla 18. Factores y niveles.....	106
Tabla 19. Resultados instancia k5m2i3.....	107
Tabla 20. Resultados instancia k5m2i5.....	108
Tabla 21. Resultados instancia k5m3i3.....	109
Tabla 22. Resultados instancia k5m3i5.....	110
Tabla 23. Resultados instancia k7m2i3.....	111
Tabla 24. Resultados instancia k7m2i5.....	112
Tabla 25. Resultados instancia k7m3i3.....	113
Tabla 26. Resultados instancia k7m3i5.....	114
Tabla 27. Resultados instancia k9m2i3.....	115

Tabla 28. Resultados instancia k9m2i5	116
Tabla 29. Resultados instancia k9m3i3	117
Tabla 30. Resultados instancia k9m3i5	118
Tabla 31. Instancia <i>k5m2i3</i>	139
Tabla 32. Instancia <i>k5m2i5</i>	140
Tabla 33. Instancia <i>k5m3i3</i>	141
Tabla 34. Instancia <i>k5m3i5</i>	142
Tabla 35. Instancia <i>k7m2i3</i>	143
Tabla 36. Instancia <i>k7m2i5</i>	144
Tabla 37. Instancia <i>k7m3i3</i>	145
Tabla 38. Instancia <i>k7m3i5</i>	146
Tabla 39. Instancia <i>k9m2i3</i>	147
Tabla 40. Instancia <i>k9m2i5</i>	148
Tabla 41. Instancia <i>k9m3i3</i>	150
Tabla 42. Instancia <i>k9m3i5</i>	151
Tabla 43. Tiempos de procesamiento y alistamiento de la instancia k2m2i2.....	153

LISTA DE GRÁFICAS

	Pág.
Gráfica 1. Porcentaje soluciones óptimas para instancias con 2 máquinas.....	68
Gráfica 2. Porcentaje soluciones óptimas para instancias con 3 máquinas.....	69
Gráfica 3. <i>Makespan</i> o costo total del país	73
Gráfica 4. Diagrama Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k5m2i3.....	84
Gráfica 5. Plot error gráfica normalidad e independencia k5m2i3.....	84
Gráfica 6. Diagrama Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k5m2i5.....	86
Gráfica 7. Plot error gráfica normalidad e independencia k5m2i5.....	86
Gráfica 8. Diagrama de Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k5m3i3.....	87
Gráfica 9. Plot error gráfica normalidad e independencia k5m3i3.....	88
Gráfica 10. Diagrama Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k5m3i5.....	89
Gráfica 11. Plot error gráfica normalidad e independencia k5m3i5.....	89
Gráfica 12. Diagrama Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k7m2i3.....	90
Gráfica 13. Plot error gráfica normalidad e independencia k7m2i3.....	91
Gráfica 14. Diagrama Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k7m2i5.....	92
Gráfica 15. Plot error gráfica normalidad e independencia k7m2i5.....	92
Gráfica 16. Diagrama Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k7m3i3.....	94
Gráfica 17. Plot error gráfica normalidad e independencia k7m3i3.....	94
Gráfica 18. Diagrama Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k7m3i5.....	96
Gráfica 19. Plot error gráfica normalidad e independencia k7m3i5.....	96

Gráfica 20. Diagrama Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k9m2i3.....	97
Gráfica 21. Plot error gráfica normalidad e independencia k9m2i3.....	98
Gráfica 22. Diagrama Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k9m2i5.....	99
Gráfica 23. Plot error gráfica normalidad e independencia k9m2i5.....	99
Gráfica 24. Diagrama de Pareto de efectos estandarizados para <i>makespan</i> de k9m3i3.	101
Gráfica 25. Plot error gráfica normalidad e independencia k9m3i3.....	101
Gráfica 26. Diagrama Pareto y gráfica normal de efectos estandarizados para <i>makespan</i> de k9m3i5.....	102
Gráfica 27. Plot error gráfica normalidad e independencia k9m3i5.....	103

LISTA DE FIGURAS

	Pág.
Figura 1. Diagrama de flujo del algoritmo ICA.....	43
Figura 2. Población inicial de imperialistas y sus colonias.....	45
Figura 3. Competencia entre los imperialistas.....	46
Figura 4. Sistema productivo bajo estudio.....	61
Figura 5. Caso $C_{j,i} \geq C_{k,i-1}$	66
Figura 6. Caso 2 $C_{j,i} \leq C_{k,i-1}$	66
Figura 7. Ejemplo de generación de un país para $k5m2i3$	71
Figura 8. Procedimiento de asimilación.....	76
Figura 9. Información compartida entre colonias.....	77
Figura 10. Revolución.....	78
Figura 11. Diagrama de flujo algoritmo propuesto.....	81
Figura 13. Habilidad de diccionario en GAMS.....	159
Figura 14. Creación de archivos gams.gms y dict.txt.....	159
Figura 15. Ruta del archivo dict.txt.....	160
Figura 16. Archivo gams.gms.....	161
Figura 17. Archivo dict.txt.....	162
Figura 18. Asignación de trabajos para la instancia $k2m2i2$	167
Figura 19. Tiempos de finalización y makespan de la instancia $k2m2i2$	167
Figura 20. Diagrama de Gantt para instancia $k2m2i2$	168

LISTA DE ANEXOS

Anexo A. Formulación del modelo matemático en la estructura del lenguaje GAMS	132
Anexo B. Instancias utilizadas para la validación del modelo	138
Anexo C. Guía para la validación de un modelo matemático en GAMS	153
Anexo D. Códigos de programación del algoritmo competitivo imperialista.....	169

Los siguientes anexos se encuentran en carpeta adjunta

Anexo E. Datos para el diseño experimental aplicado al algoritmo propuesto (ver archivo excel)

Anexo F. Datos para la validación del algoritmo propuesto (ver archivo excel)

RESUMEN

TÍTULO: “METAHEURISTICA BASADA EN EL ALGORITMO COMPETITIVO IMPERIALISTA (ICA) APLICADA A LA SOLUCIÓN DEL PROBLEMA DE FLOW SHOP HÍBRIDO (HFS) CON MÁQUINAS PARALELAS NO RELACIONADAS”¹

AUTORES:

MORATTO CHIMENTY, Eslendis
PÉREZ FIGUEREDO, Luis Eduardo².

PALABRAS CLAVES:

Flow shop híbrido, máquinas paralelas, algoritmo competitivo imperialista, algoritmo genético, metaheurística.

DESCRIPCIÓN

En la presente investigación se estudia el problema de *flow shop* híbrido con máquinas paralelas no relacionadas con el propósito de minimizar el makespan, el cual es considerado un problema de optimización combinatoria y clasificado como NP-hard debido a su complejidad computacional representada en el tiempo y memoria requerida para encontrar una solución óptima.

Diversos autores han implementado métodos heurísticos y metaheurísticos para resolver este problema, los cuales si bien no garantizan una solución óptima, permiten obtener una respuesta aproximada en un tiempo computacional razonable. Entre los métodos de solución comúnmente utilizados se encuentran el algoritmo genético, recocido simulado y combinación de reglas de secuenciación tales como la fecha de entrega más cercana y el tiempo de procesamiento más corto.

En este trabajo se propone una metaheurística basada en el algoritmo competitivo imperialista para la minimización del makespan, considerando tiempos de alistamiento dependientes de la secuencia y de la máquina. En varias de las etapas del algoritmo competitivo imperialista se utilizan operadores característicos del algoritmo genético (mutación y cruce). La eficiencia de la metaheurística se mide mediante doce instancias generadas por los autores y validadas en lenguaje de programación Matlab® y se compara con la solución obtenida por un método exacto utilizando el software GAMS®.

Los resultados muestran que el algoritmo propuesto logra obtener soluciones de mejor calidad que las obtenidas mediante el método exacto para algunas de las instancias al combinar en sus niveles altos dos de los parámetros considerados en el diseño experimental.

¹ Proyecto de Grado

² Facultad de Ingenierías Fisicomecánicas. Escuela de Estudios Industriales y Empresariales. Directora: M.s.c. Eliana M. Peña Tibaduiza. Codirector: Esp. Edwin A. Garavito Hernández.

ABSTRACT

TITLE: “METAHEURISTIC BASED ON IMPERIALIST COMPETITIVE ALGORITHM (ICA) APPLIED TO THE SOLUTION OF HYBRID FLOW SHOP PROBLEM (HFS) WITH UNRELATED PARALLEL MACHINES”³

AUTHORS

MORATTO CHIMENTY, Eslendis
PÉREZ FIGUEREDO, Luis Eduardo⁴.

KEYWORDS

Hybrid flow shop, parallel machine, imperialist competitive algorithm, genetic algorithm, metaheuristic.

DESCRIPTION

In this research the hybrid flow shop problem with unrelated parallel machines was studied in order to minimize the makespan, which is considered a combinatorial optimization problem and it is classified as NP-hard due to its computational complexity represented in time and memory required to find an optimal solution.

Several authors have implemented heuristics and metaheuristics to solve this problem, eventhough they do not guarantee an optimal solution, allow to obtain an approximate answer in a reasonable computational time. Among the methods of solution commonly used are the genetic algorithm, simulated annealing and combination of dispatching rules such as Earliest Due Date and Short Processing Time.

A metaheuristic solution procedure was proposed based on the imperialist competitive algorithm to minimize the makespan, considering that set-up times depend on the sequence and the machine. In various stages of imperialist competitive algorithm characteristic genetic algorithm operators (mutation and crossover) are used. Metaheuristic efficiency is measured by 12 instances, which were generated by the authors validated in Matlab®, and were compared with the results obtained using the exact method using GAMS® software.

The results show that the proposed algorithm obtains a better quality solutions than those obtained through the exact method for some of the instances when were combined two of the parameters considered in the experimental design in their high levels.

³ Proyecto de Grado

⁴ Facultad de Ingenierías Fisicomecánicas. Escuela de Estudios Industriales y Empresariales. Directora: M.s.c. Eliana M. Peña Tibaduiza. Codirector: Esp. Edwin A. Garavito Hernández.

INTRODUCCIÓN

«La Dirección de Operaciones, tiene el reto de adaptar los procesos de gestión de los sistemas productivos y logísticos a las actuales necesidades de las empresas; en este proceso de cambio es habitual partir de sistemas productivos poco flexibles y orientados a la producción en masa en los que es fundamental emplear el mejor “saber-hacer” para procurar obtener el rendimiento más adecuado de los recursos disponibles»⁵. Es aquí, donde se reconoce la importancia de los procesos de programación, puesto que estos permiten organizar la producción para reducir en gran medida los tiempos de respuesta al cliente.

Tradicionalmente los problemas de programación de la producción son evaluados como sistemas ideales, es decir, pocos son los trabajos realizados acordes con lo que realmente sucede en el entorno manufacturero, en donde se presenta gran variedad de procesos productivos. En cualquier tipo de sistema es de gran utilidad una adecuada programación para lograr posicionar a las organizaciones.

Uno de los sistemas productivos presentes en la industria es el *flow shop*, en el cual un conjunto de tareas debe seguir un flujo unidireccional para ser procesado a través de múltiples estaciones, donde cada estación consta de una sola máquina. Sin embargo, la necesidad de aumentar o equilibrar la capacidad debida a aumentos de demanda ha llevado a la adquisición de nuevos procesadores en algunas etapas⁶, que pueden coexistir con los ya disponibles y generalmente

⁵ GÓMEZ, Pedro. Programación de la producción en un taller de flujo híbrido sujeto a incertidumbre: arquitectura y algoritmos. Aplicación a la industria cerámica. 2010. Tesis Doctoral.

⁶ RIBAS, Imma, et al. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. En: Computers & Operations Research, 2010, vol. 37, no 8, p. 1439-1454.

difieren en las características de operación: son más rápidos, tienen un menor tiempo de alistamiento, pueden procesar más trabajos, etc.

Este tipo de sistemas es frecuente encontrarlos en industrias como las de fabricación de baldosas cerámicas, automotriz, metalmecánicas, de fabricación de televisores con circuito impreso a bordo, entre otras; en las cuales existe la necesidad de asignar n trabajos a s etapas de procesamiento con el propósito de encontrar una programación óptima que permita atender una función objetivo en particular y suele ser denominado como “*flow shop*” híbrido o “*flow shop*” flexible.

Dicho problema ha sido del interés de numerosos investigadores desde que aparecieran los primeros trabajos en 1970, dada su complejidad y aplicación práctica pertenece al área de la optimización combinatoria y se han planteado diferentes métodos tanto heurísticos como exactos para hallar una solución factible considerando diferentes restricciones.

La presente investigación se enfoca en dar una solución al problema de “*flow shop*” híbrido con máquinas paralelas no relacionadas, a través de una metaheurística basada en el algoritmo competitivo imperialista, considerando los supuestos de tiempos de alistamiento dependientes de la secuencia y tiempos de alistamientos dependientes de la máquina. La eficiencia de la metaheurística se mide mediante 12 instancias generadas por los autores y sus resultados son comparados con los obtenidos por un método exacto.

Los aportes de este proyecto son de relevancia para el grupo de investigación OPALO y la Escuela de Estudios Industriales y Empresariales en la consolidación y el fortalecimiento de la línea de investigación de programación de operaciones.

Cumplimiento de objetivos	
Objetivos específicos	Numerales relacionados
Realizar una revisión de la literatura acerca del problema de <i>flow shop</i> híbrido con máquinas paralelas no relacionadas.	3
Definir el modelo de programación matemática que se ajuste al problema.	4.2.1
Diseñar la estructura del algoritmo acorde con las necesidades del problema de <i>flow shop</i> híbrido.	5.1
Implementar el algoritmo en el software <i>MATLAB</i> y validarlo a través de diferentes instancias.	Anexo D, 7
Comparar los resultados obtenidos mediante el algoritmo con soluciones encontradas a través de métodos exactos.	7.1

1. GENERALIDADES DEL PROYECTO

1.1. PLANTEAMIENTO DEL PROBLEMA

El problema de *flow shop* híbrido (*hybrid flow shop, HFS*) tiene como objetivo determinar de forma adecuada la secuencia y asignación de un conjunto de tareas a un conjunto de m máquinas, para lograr optimizar una función en particular; dicha función considerada por lo general es minimizar el *makespan* o la cantidad de trabajos tardíos. En este problema se considera que los trabajos tienen el mismo orden de procesamiento y el flujo del proceso es unidireccional.

El *HFS* como se ha venido trabajando, no contempla aspectos de la producción en las industrias reales, debido a que los estudios que generalmente se hacen no consideran parámetros y supuestos presentes en la realidad (tiempos dependientes de la secuencia, elegibilidad de máquinas, existencia de zonas de almacenamiento, entre otras), esta situación ha generado que los estudios realizados sobre este problema sean pocos.

El presente proyecto basará sus estudios en el problema de *flow shop* híbrido con máquinas no relacionadas y tiempos de alistamiento dependientes de la secuencia y de la máquina, la cual es una de las restricciones que mayor dificultad le aporta al problema, convirtiéndolo en *NP-HARD*, es decir, de difícil solución dado que no se cuenta con algoritmos de orden polinomial a partir de los cuales se pueda resolver el problema de forma óptima y en el menor tiempo posible⁷.

Cuando se habla de máquinas no relacionadas se asume que los tiempos de

⁷ JIMENEZ, Ángela Patricia. Solución del problema de flow shop flexible aplicando el algoritmo genético de Chu-Beasley. En: Entre Ciencia e Ingeniería, 2013, no 13, pp, 34-40.

procesamiento de las tareas dependen de cada procesador específico, debido a las diferencias existentes en las velocidades de procesar de cada uno. Desde que el problema fue abordado por primera vez, los investigadores han intentado desarrollar modelos y estrategias para darle solución, “tanto las estrategias como los modelos han evolucionado a través del tiempo, convirtiéndose en la actualidad en un reto para las técnicas computacionales”⁸. Esto ha generado el desarrollo de metaheurísticas basadas en combinación de diversos algoritmos, siendo los más comunes: el algoritmo genético, la combinación de reglas de secuenciación (*EDD*, *SPT*, *LPT*, entre otras), el recocido simulado (*SA*, por sus siglas en inglés) y la búsqueda tabú.

En esta ocasión la metaheurística a desarrollar será basada en el algoritmo competitivo imperialista (*ICA*, por sus siglas en inglés) y luego los resultados obtenidos serán comparados con los logrados mediante la implementación de un método exacto.

1.2. JUSTIFICACIÓN DEL PROYECTO

Dada la situación de globalización de los mercados en donde se ve reflejado un aumento en la competitividad y el afán por satisfacer de forma más rápida todas las necesidades y expectativas del cliente, se hace imprescindible el uso óptimo de los recursos y la búsqueda de nuevas metodologías que permitan hacer eficientes los procesos de manufactura, siendo uno de los intereses comunes en las organizaciones mejorar la flexibilidad de sus procesos para adaptarse a fluctuaciones de la demanda y disminuir los tiempos de procesamiento de los diferentes trabajos. Esta flexibilidad se logra mediante una adecuada programación de las operaciones.

⁸ Ibid., p. 35.

“En las empresas del pasado, el problema de la programación de operaciones ha sido resuelto típicamente de forma manual, utilizando criterios basados en la experiencia del personal”⁹ pues los sistemas utilizados, en su mayoría contaban con una sola máquina en cada una de las etapas del proceso. Pero, con el paso del tiempo y con las nuevas exigencias de los clientes, estos sistemas fueron cambiando haciendo necesaria la adquisición de nueva maquinaria en cada una de las etapas. De ahí surge el *flow shop* híbrido, un sistema compuesto de varias etapas en serie, cada una de las cuales puede constar de más de una máquina en paralelo y donde los trabajos siguen un flujo unidireccional, y a su vez permite modelar diferentes procesos productivos comunes en el mundo industrial.¹⁰

Tradicionalmente los trabajos que se han realizado sobre éste problema han estudiado la restricción de máquinas paralelas idénticas, caso que no es muy común en las industrias, pues las especificaciones de la nueva maquinaria adquirida cambian haciendo que difieran en características de operación con las iniciales¹¹. A partir de lo anterior, se ha generado interés alrededor de esta problemática, llevando a diversos autores a incorporar en el problema la restricción de máquinas paralelas no relacionadas, tal es el caso de Ruiz y

⁹ DE MICHELI, Mauro, et al. Algoritmo de búsqueda local iterativa para la programación de piezas en un sistema flow shop híbrido. En: UPCCommons. Universitat Politècnica de Catalunya. 2009., p.28

¹⁰ RIBAS, Imma y PASCUAL, Ramón. Programación de la Producción en un sistema flow shop híbrido sin esperas y tiempos de preparación dependientes de la secuencia. En: ETSEIB-Departament d'Organització d'Empreses, 2006., p. 21-32

¹¹ RIBAS, Imma, et al.. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. En: Computers & Operations Research, 2010, vol. 37, no 8, p. 1439-1454.

Maroto¹² que en 2006 estudiaron el problema aplicado en la industria de fabricación de baldosas de cerámica, Jitti et al.,¹³ en 2009 en la producción de la industria textil y de montajes de vehículos, y Almeder y Hartl¹⁴ quienes en 2013 consideran el problema de dos etapas en un caso de aplicación en la industria metalúrgica.

Teniendo en cuenta que los trabajos realizados sobre *flow shop* híbrido con máquinas paralelas no relacionadas son pocos en comparación a los que consideran los procesadores idénticos, es propósito fundamental de esta investigación analizar el problema de secuenciación de trabajos considerando las características descritas previamente, mediante el desarrollo de una metaheurística basada en el algoritmo *ICA* para mejorar la asignación de los recursos.

El *ICA*, según Mendoza et al.,¹⁵ es una técnica basada en el imperialismo y el proceso de la competencia imperialista. Especialmente, este algoritmo considera el imperialismo como un nivel de evolución social humana, compuesto por modelos matemáticos y por los complicados procesos históricos y políticos, como

¹² RUIZ, Ruben y MAROTO, Concepción. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. En: European Journal of Operational Research, 2006, vol. 169, no 3, p. 781-800.

¹³ JUNGWATTANAKIT, Jitti, et al. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: Computers & Operations Research, 2009, vol. 36, no 2, p. 358-378

¹⁴ ALMEDER, Christian y HARTL, Richard F. A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer. En: International Journal of Production Economics, 2013, vol. 145, no 1, p. 88-95.

¹⁵ MENDOZA, Franklin, et al. Herramientas computacionales "CI-API-1": una alternativa para la solución de problemas en sistemas de distribución de energía eléctrica. En: Memorias de las IX Jornadas de Investigación 2011-UNEXPO Puerto Ordaz, 2011, p. 10-22.

la herramienta para la optimización de la evolución, lo que garantiza que la calidad de las soluciones obtenidas sea alta.

Otra de las acciones que se desea, es incentivar a los estudiantes de Ingeniería Industrial al estudio de estos sistemas, pues es un problema que no ha sido abordado como tema de investigación en la Escuela de Estudios Industriales y Empresariales y pertenece a la línea de programación de operaciones del Grupo de Optimización y Organización de Sistemas Productivos, Administrativos y Logísticos (OPALO), en el cual se consideran distintas variables que pueden ser aplicadas para mejorar el desempeño de las industrias manufactureras.

1.3. OBJETIVOS

1.3.1. Objetivo general

Desarrollar una metaheurística basada en el algoritmo competitivo imperialista (ICA), para solucionar el problema de *flow shop* híbrido con máquinas paralelas no relacionadas.

1.3.2. Objetivos específicos

- Realizar una revisión de la literatura acerca del problema de *flow shop* híbrido con máquinas paralelas no relacionadas.
- Construir el modelo de programación matemática que se ajuste al problema.
- Diseñar la estructura del algoritmo acorde con las necesidades del problema de *flow shop* híbrido.
- Implementar el algoritmo en el software *MATLAB* y validarlo a través de diferentes instancias.
- Comparar los resultados obtenidos mediante el algoritmo con soluciones encontradas a través de métodos exactos.

2. MARCO TEÓRICO

2.1. OPTIMIZACIÓN

La optimización es una disciplina fundamental en campos de la ciencia tales como la informática, la inteligencia artificial o la investigación operativa y es concebida como “el proceso de intentar encontrar la mejor solución posible a un problema generalmente en un tiempo limitado”.¹⁶ Un problema de optimización puede ser definido como aquel en el que hay muchas posibles soluciones y alguna forma clara de comparación entre ellas, de tal forma que éste existe si y sólo si se dispone de un conjunto de soluciones candidatas diferentes que pueden ser comparadas y puede ser formulado matemáticamente como se muestra a continuación en la ecuación 1:

$$P = \begin{cases} \text{opt } f(x) \\ \text{s. a:} \\ x \in F \subset SS \end{cases} \quad (1)$$

Donde $f(x)$ es la función a optimizar, F el conjunto de soluciones factibles y SS el espacio de soluciones.

Este tipo de problema se puede dividir en dos categorías: aquellos en los que la solución está codificada mediante valores reales y aquellos cuyas soluciones están codificadas por valores enteros, en esta última categoría se encuentran un tipo particular de problema denominado problema de optimización combinatoria, que consiste en encontrar el máximo (o el mínimo) de una determinada función sobre un conjunto finito de soluciones.

¹⁶ DUARTE, Abraham, et al. Metaheurísticas. Madrid: Dykinson, 2007. p 1

La optimización combinatoria se deriva de la optimización matemática discreta¹⁷, por lo cual es usual encontrar los conceptos de enumeración, combinación y permutación. Partiendo de que se establece un conjunto S cuyo tamaño es $|S| = n$, se define la enumeración de dicho conjunto como una lista exacta de todos los elementos que lo componen.

Una combinación es una colección desordenada de elementos distintos, por lo general de un tamaño k tomados de S . Por otra parte, una permutación es una secuencia que contiene elementos distintos de k .

Un problema de optimización relacionado con la combinación óptima de los elementos se denomina un problema de agrupación. En cuanto a la optimización relacionada con la permutación óptima de los elementos se llama un problema de programación, entre ellos se encuentra el problema de programación *flow-shop*, el cual busca determinar el tiempo mínimo de procesamiento total mediante el ajuste de la secuencia de tareas en los puestos de trabajo.

Los problemas de optimización combinatoria presentan como una particularidad que siempre existe un algoritmo exacto que permite obtener la solución óptima¹⁸; sin embargo, éste suele ser ineficiente, ya que en la mayoría de los casos, el tiempo que emplearía en encontrar una solución crece de forma exponencial con el tamaño del problema. Con base en lo anterior los problemas de optimización combinatoria suelen ser clasificados según su complejidad computacional.

¹⁷ YU, Xinjie y GEN, Mitsuo. Introduction to evolutionary algorithms. Springer Science & Business Media, 2010. pagina 263

¹⁸ DUARTE, et al. Op. Cit. p 2.

2.2. COMPLEJIDAD COMPUTACIONAL

Un problema de optimización puede caracterizarse matemáticamente de acuerdo a la dificultad que representa su solución para un computador, la cual influye directamente en la eficiencia y la eficacia del algoritmo utilizado. Desde el punto de vista de la teoría de complejidad, cuyo objetivo es “clasificar los problemas en función de la resolubilidad algorítmica práctica de los mismos”¹⁹, se puede definir complejidad como “la cantidad de recursos necesarios para efectuar un cálculo”²⁰, cuyo uso puede ser medido en función de dos indicadores²¹:

- Memoria ocupada: que muestra la cantidad de memoria que ocupan todas las variables utilizadas por el algoritmo, es decir la suma del almacenamiento necesario para ejecutarlo, y está compuesta por la memoria estática y la memoria dinámica.
- Tiempo de ejecución: depende del número de datos de entrada del problema y de la velocidad que el equipo de cómputo posea y es medido mediante la generación de una función denominada función temporal, que define el número de instrucciones ejecutadas por el programa cuando se resuelve un problema. Se dice que un algoritmo es eficiente si su costo es mínimo en cuanto al costo espacial y al costo temporal.

Tipos de complejidad. La complejidad de un problema de optimización vista desde la teoría de la complejidad está definida por los siguientes elementos²²:

¹⁹ JIMÉNEZ, Mario y CAPARRINI, Fernando. Máquinas moleculares basadas en ADN. Universidad de Sevilla, 2003.pag 23

²⁰ CRUZ, Marco, et al. Aplicación de la teoría de la complejidad en optimización combinatoria. Inventio, la génesis de la cultura universitaria en Morelos, 2014, no 20, p. 35.

²¹ Ibid. p.37

²² JIMÉNEZ y CAPARRINI, Op.cit., p. 24

- Un modelo de computación que proporcione los dispositivos sobre los que se van a resolver los problemas, generalmente como modelo de computación se consideran las máquinas de Turing, las cuales pueden ser deterministas o no deterministas.
- Un modo de computación que precise el concepto de aceptación de un dato de entrada y que a su vez fije el significado de resolución de un problema.
- Una medida de complejidad que permita cuantificar los recursos usados por los dispositivos computacionales en la resolución de problemas.
- Una función total computable entre números naturales que sirva de cota superior de los recursos usados.

Mediante el modelo computacional de Turing es posible clasificar los problemas por su grado de complejidad para resolverlos, a través de éste se han identificado problemas intratables, clasificados como *NP* (*nondeterministic polynomial time*) los cuales son imposibles de resolver en un tiempo razonable cuando el número de variables que los componen es una cantidad extremadamente grande²³, a esta clase pertenecen los que se listan a continuación.

- Problemas P: aquellos problemas que pueden ser resueltos en un tiempo polinómico utilizando una máquina de Turing determinista.
- Problemas NP-completos. Son aquellos cuya solución no ha sido resuelta de forma exacta por medio de algoritmos deterministas en tiempo polinomial, por consiguiente se tratan de resolver mediante algoritmos no deterministas, cuya solución deseable sea de complejidad polinomial. Se dice que un problema es *NP*- Completo si es posible convertir el problema en otro de tal forma que la

²³ CRUZ et al. Op. Cit., 39

solución al segundo problema se pueda utilizar para resolver el primero²⁴.

- Problemas NP-duros. Son igual o incluso más difíciles de resolver que los problemas NP-Completos puesto que no existe un algoritmo polinómico que permita verificar una solución.

2.3. MÉTODOS DE SOLUCIÓN

Los algoritmos utilizados para resolver los problemas de optimización combinatoria pueden ser clasificados en las siguientes categorías:

2.3.1. Algoritmos exactos. Son aquellos que garantizan la optimalidad de los resultados, pero no tienen tiempo de complejidad polinómica para resolver los problemas de tipo *NP*-completo o *NP*-duro. A esta categoría pertenecen:

- Método Simplex.

Desarrollado por G.B Dantzig en 1947 quien lo utilizó para solucionar problemas logísticos de la fuerza aérea como el despliegue de tropas y la entrega de suministros²⁵. Se fundamenta en el método gráfico, que establece que la solución óptima siempre está asociada con un punto de esquina del espacio de soluciones. El objetivo del método simplex es mover la solución a una esquina, la cual está en capacidad de mejorar el valor de la función objetivo en cada iteración, este procedimiento se detiene cuando se encuentra la solución óptima.

²⁴ IGLESIAS, José et al. Complejidad Computacional. Grado Ingeniería informática. Teoría de Autómatas y Lenguas formales.

²⁵FISHBACK, Paul E. Linear and nonlinear programming with Maple: an interactive, applications-based approach. CRC Press, 2009.

- Algoritmo de ramificación y acotamiento.

Desarrollado por G. Doig en 1960, es un esquema sistemático para enumerar implícitamente los números finitos de soluciones factibles de un problema de programación lineal. Aunque, teóricamente el tamaño del árbol de enumeración es exponencial en los parámetros del problema, en la mayoría de los casos, el método elimina un gran número de soluciones factibles.

2.3.2. Algoritmos heurísticos. Todos aquellos procedimientos que pueden proporcionar una buena solución la cual no necesariamente es la óptima, ya que el proceso iterativo puede detenerse al encontrar un óptimo local y por consiguiente no continuar la búsqueda del óptimo global²⁶. Algunas de las razones por las cuales se implementan los métodos heurísticos son las siguientes:

- Dada la complejidad del problema, no existe un método exacto que pueda encontrar solución al problema en tiempo razonable.
- A pesar de que exista un método exacto para resolver el problema, su implementación es computacionalmente muy costosa.
- La flexibilidad de un método heurístico es mayor a la de un método exacto, lo que posibilita la inclusión de condiciones difíciles de modelación.

Dado que los algoritmos heurísticos son de diferente naturaleza, algunos incluso diseñados para un problema en específico, no existe una clasificación exacta, sin embargo, pueden ser agrupados en diferentes categorías las cuales se describen a continuación²⁷:

²⁶ MARTÍ, Rafael y REINELT, Gerhard. The linear ordering problem: exact and heuristic methods in combinatorial optimization. Springer Science & Business Media, 2011.

²⁷ DUARTE, et al. Op. Cit. p.,27

- Métodos constructivos

Son métodos deterministas capaces de construir una solución a un problema en particular, que depende de la estrategia utilizada, las cuales pueden ser:²⁸

- a. Estrategia voraz: denominada también como algoritmo ávido o algoritmo miope, construye paso a paso una solución factible a partir de una semilla.
- b. Estrategia de Descomposición: divide el problema principal en subproblemas más sencillos de resolver, la división se realiza hasta obtener un subproblema cuya solución sea trivial.
- c. Métodos de reducción: identifican propiedades o características que contienen las buenas soluciones para convertirlas en restricciones del problema con el propósito de limitar el espacio de soluciones y de esta forma simplificar el problema.
- d. Métodos de manipulación del modelo: conocidos también como métodos inductivos, buscan simplificar el problema original para obtener una solución al problema simplificado, a partir de la cual es calculada la solución al problema original. entre estos métodos se encuentran: la linealización, la agrupación de variables, introducción de nuevas restricciones entre otras.

- Métodos de búsqueda

Se caracterizan porque parten de una solución factible la cual intentan mejorar.

Entre las estrategias de búsqueda se encuentran:

- a. Estrategia de búsqueda local 1: mejora progresivamente la solución factible inicial, mediante un análisis de la vecindad que permite determinar cuál es el primer movimiento que mejora la solución actual.

²⁸ Ibid., p. 28.

- b. Estrategia de Búsqueda local 2: al igual que la estrategia anterior mejora progresivamente la solución factible inicial, sin embargo al examinar la vecindad de dicha solución se evalúan todos los posibles movimientos de los cuales se selecciona el que produzca una mayor mejora en la función objetivo.
- c. Estrategia aleatorizada: a partir de la solución factible dada y de una vecindad asociada a esa solución, se seleccionan aleatoriamente soluciones vecinas a esa vecindad

2.3.3. Algoritmos Metaheurísticos. El término metaheurístico fue utilizado por primera vez por Fred Glover en 1986 y de acuerdo a Osman y Kelly²⁹ citado por Rafael Martí, son “procesos de generación iterativos que guían una heurística subordinada combinando inteligentemente diferentes conceptos para explorar y aprovechar los espacios de búsqueda usando las estrategias de aprendizaje para estructurar la información con el fin de encontrar soluciones casi óptimas, diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos”.

A pesar que no existen características que sean comunes a todas las metaheurísticas y que permitan realizar una clasificación exacta de las mismas, éstas pueden ser clasificadas desde su enfoque conceptual (o estado puro) y su implementación real, la cual es más complicada dado que considera las inhomogeneidades existentes entre ellas³⁰.

A continuación se describen algunas características usadas para clasificar las metaheurísticas desde la taxonomía clásica. Cabe destacar que las alternativas de clasificación descritas no permiten establecer una separación clara entre todas las

²⁹ MARTI, Rafael. Procedimientos metaheurísticos en optimización combinatoria. *Matemáticas*, 2003, vol. 1, no 1, p. 3-62.

³⁰ DUARTE et al. Op. cit. p., 35

metaheurísticas, por lo cual algunos autores las toman como referencia para realizar una clasificación más robusta.

- Inspiración.
 - a. Natural: aquellos algoritmos que toman como referencia un comportamiento biológico, social, cultural, etc.
 - b. Sin inspiración: algoritmos que se obtienen directamente de sus propiedades matemáticas.
- Número de soluciones.
 - a. Poblacionales: la búsqueda de una solución óptima de un problema es realizada a través de un conjunto de soluciones.
 - b. Trayectoriales: la búsqueda del óptimo se lleva a cabo utilizando una sola solución la cual se mejora iterativamente.
- Función objetivo.
 - a. Estáticas: no se realiza ninguna modificación sobre la función objetivo durante la búsqueda.
 - b. Dinámicas: modifican la función objetivo durante la búsqueda.
- Vecindad.
 - a. Una vecindad: para realizar la búsqueda de la solución óptima solamente se utiliza una estructura de vecindad.
 - b. Varias vecindades: durante la búsqueda se modifican las estructuras de vecindades.
- Uso de memoria.
 - a. Sin memoria: se basan en el estado anterior.
 - b. Con memoria: utilizan una estructura de memoria para recordar la historia pasada.

2.3.4. Metaheurísticas utilizadas para solucionar problemas de optimización combinatoria. A continuación se describen algunas de las metaheurísticas comúnmente utilizadas.

- Búsquedas prohibidas (*Tabú Search*).

Introducida por primera vez por Glover en 1986, consiste en un procedimiento de mejora, donde se considera el salto a la mejor solución de una vecindad, incluso cuando ésta tenga un costo algo peor que la solución actual, lo que permite la opción de salir de óptimos locales en favor de otros mejores. La nueva vecindad denominada “vecindad legal”, está definida por una relación de vecindad de una solución y por una lista de movimientos prohibidos que recuerda los últimos movimientos realizados para evitar que se repitan soluciones anteriores y permite intensificar o diversificar su búsqueda de espacio de la solución de un problema determinado, en un esfuerzo para evitar el atrapamiento en óptimos locales.

- Recocido simulado (*Simulated Annealing*).

Conocido también como temple simulado, fue concebida por Kirkpatrick et al. en la década de 1980, es propuesto como una analogía con el fenómeno físico del temple de un metal en una fragua. Es un algoritmo aleatorizado de búsqueda local por vecindades, donde movimientos que mejoran la función objetivo se aceptan sin duda, y movimientos que empeoran la función objetivo se aceptan con una probabilidad dependiente de un parámetro controlable (llamado temperatura) que disminuye a medida que avanza el proceso iterativo. Al esquema que controla la temperatura se le llama esquema de enfriamiento (*cooling schedule*) que influye en la calidad de la solución final.

- Búsqueda de vecindad variable (*Variable Neighborhood Search*).

Propuesta por P. Hansen y N. Mladenovic (1999, 2001), es un algoritmo en el que la búsqueda de una solución óptima se fundamenta en un cambio sistemático de las estructuras de vecindades dentro de un procedimiento de búsqueda local. La metaheurística se basa en tres principios descritos a continuación:

- a. Un óptimo local respecto a una vecindad no tiene por qué serlo con respecto a otra vecindad.
- b. Un óptimo global es un óptimo local con respecto a todas las posibles estructuras de vecindad.
- c. Para muchos problemas, los óptimos locales con respecto a una o varias estructuras de vecindad está relativamente cerca.

Existen varias categorías del algoritmo *VNS*, que dependen de la forma en las que se combinan los principios descritos previamente, entre ellas se encuentran:

1. Búsqueda del entorno variable decente (*Variable Neighborhood Descent*): el cambio de vecindad se realiza de forma determinista, tal que si el proceso de búsqueda queda atrapado en un óptimo local, se realiza un cambio en la vecindad a fin de escapar de dicho óptimo.
2. Búsqueda del entorno variable reducido (*Restricted Variable Neighborhood Search*): combina los tres principios de forma estocástica, en este método se selecciona aleatoriamente un punto que pertenece a una vecindad dada, sin realizar ningún tipo de optimización local.
3. Búsqueda del entorno variable básico (*Basic Variable Neighborhood Search*): en esta estrategia se combinan aspectos deterministas y estocásticos, al igual que en la búsqueda del entorno variable reducido, se selecciona un punto que pertenece a una vecindad, sin embargo realiza una mejora mediante una estrategia de búsqueda local.

- Algoritmos evolutivos (*Evolutionary Algorithms*).

Pueden alcanzar una solución casi óptima, dado que su complejidad de tiempo no está relacionada con la dimensión del problema y por consiguiente requerirá de un tiempo relativamente corto. Los algoritmos evolutivos están conformados por: la población, que es un conjunto de soluciones candidatas de un problema dado (individuos); selección, es decir, el método para realizar la elección de individuos de forma que los seleccionados sean los mejores y a la vez transmitan su código genético a la descendencia y la alteración que es el mecanismo que permite crear nuevos individuos al modificar ciertas características de forma estocástica.

Los algoritmos evolutivos se dividen de acuerdo a las características de implementación en cuatro grupos, los cuales serán descritos a continuación.

1. Programación evolutiva: se caracteriza por no realizar el cruce de los individuos, razón por la cual el mecanismo de mutación es más complejo.
2. Estrategias de evolución: éste algoritmo inicializa y evalúa una población para seleccionar los padres de forma aleatoria y uniforme. Al combinar dos padres se genera una cantidad de hijos superior a los padres, los cuales se modifican implementando una etapa de mutación.
3. Programación genética: busca construir programas informáticos mediante el uso de programas generados aleatoriamente y de los principios de los algoritmos evolucionarios.
4. Algoritmos genéticos (*Genetic Algorithm*). Desarrollados por J. Holland en 1975, se basan en la teoría de evolución de las especies enunciada por Darwin, son procedimientos iterativos que consisten en una población de individuos, cada uno de los cuales representa una solución y recibe el nombre de cromosoma el cual suele estar compuesto por unidades discretas denominadas genes.

Algunos de los elementos que conforman los algoritmos genéticos y que a su vez los diferencian de otros algoritmos evolutivos son los siguientes:

- a. Población inicial: corresponde a una generación aleatoria de soluciones al problema dado.
- b. Representación: relación entre las soluciones factibles (fenotipo) y la codificación de las variables (genotipo), estas pueden ser mediante cadenas binarias, cadenas de números o cadenas de palabras.
- c. Función de evaluación: permite cuantificar la calidad de los individuos de la población. Se denomina también como *fitness*.
- d. Operadores genéticos: permiten generar nuevos individuos bajo cierta probabilidad, comúnmente los más utilizados son:

Cruce: la generación de la descendencia, es decir nuevos individuos, se efectúa al seleccionar dos individuos de la población inicial, quienes reciben el nombre de padres e intercambiar parte de sus genes.

Mutación: es el cambio de forma aleatoria de parte de un individuo, se emplea para preservar la diversidad de las soluciones y realizar una exploración más amplia del espacio de solución.

- e. Selección: mecanismo que permite elegir con una mayor probabilidad a los individuos que posean el mayor valor de la función de evaluación. Algunos de los métodos comúnmente utilizados son los siguientes:

Método de la ruleta: cada individuo de la población es asignado a una porción de la ruleta, la cual es proporcional al valor de la función de evaluación y del promedio de las funciones de evaluación de los demás individuos.

Torneo: este método consiste en escoger q individuos de una población de tamaño n , seleccionando de ellos el que posea el mejor valor de la función de evaluación.

2.3.5. El algoritmo competitivo imperialista (Imperialist Competitive Algorithm, ICA).

Es uno de los algoritmos evolutivos más recientes, los primeros aportes fueron realizados por Athaspaz-Gargari y Lucas en 2007 y por Rajabioun et al. en 2008, se basa en la evolución socio-política de los humanos, específicamente en el imperialismo, las formas de conquista y la competencia imperialista como estrategias de optimización.

El imperialismo es definido como la política de extender el poder y las características socio-culturales de un país, lo cual se da mediante el dominio y conquista de otros países llamados colonias, quienes tienden a adoptar las características del país conquistador. Cuanto más fuerte sea el país imperialista, más oportunidades tiene de conquistar a otros países, mientras que si el imperialista es débil, tiene un mayor riesgo de perder sus colonias.

Por otra parte, un país imperialista junto con sus colonias forma un imperio, cuyo poder es la suma del poder del país imperialista con el poder de sus colonias, así pues los imperios con mayor poder tienden a conquistar los imperios con menor poder.

El algoritmo³¹ comienza con una población inicial de individuos (denominados países), los países con mejor valor de aptitud son seleccionados para ser los imperialistas, y el resto serán colonias, las cuales serán divididas entre los países imperialistas basadas en el poder de cada uno de ellos.

³¹ ATASHPAZ-GARGARI, Esmail y LUCAS, Caro. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. En Evolutionary computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007. p. 4661-4667.

Tras dividir las colonias entre los países imperialistas y de que éstas se muevan hacia su respectivo imperialista, comienza la competencia entre imperios, aquel imperio que no sea capaz de incrementar su poder o de por lo menos no dejar que éste decrezca, será eliminado de la competencia. El desarrollo de la competencia resultará en un aumento gradual del poder de los imperios más fuertes y en la caída de los más débiles, en la figura 1 es posible observar las etapas que presenta el algoritmo competitivo imperialista. A continuación se describirá de forma más detallada cada uno de los pasos del algoritmo.

1. Generación de imperios iniciales.

En un problema de optimización cuya dimensión es de N variables, un país (que en la terminología de los algoritmos genéticos recibe el nombre de cromosoma), es una formación de $1 \times N_{var}$ y se define así:

$$pais = [p_1, p_2, p_3 \dots \dots p_N] \quad (2)$$

El costo de un país se halla al evaluar la función de costo f en las variables mediante la siguiente fórmula:

$$costo = f(p_1, p_2, p_3 \dots \dots p_N) \quad (3)$$

Inicialmente una población de tamaño N_{pop} es generada aleatoriamente, de ésta se seleccionarán los N_{imp} países imperialistas y las N_{col} colonias, las cuales se asignan a los imperialistas teniendo en cuenta el costo normalizado y el poder normalizado de cada uno de ellos, calculados mediante el uso de las siguientes ecuaciones:

$$C_n = c_n - \max\{c_i\} \quad (4)$$

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (5)$$

Donde c_i es el costo del enésimo imperialista y C_n es el costo normalizado y p_n es el poder normalizado. De acuerdo con lo anterior, el número inicial de colonias de un imperio será:

$$N.C_n = round\{p_n * N_{col}\} \quad (6)$$

Para asignar las colonias a cada imperialista, se seleccionan aleatoriamente $N.cn$ colonias de toda la población. Dichas colonias junto con el imperialista formarán la población inicial de un imperio. En la figura 2 es posible observar que el imperialista con mayor poder es el que mayor número de colonias posee.

2. Movimiento de las colonias hacia el imperialista.

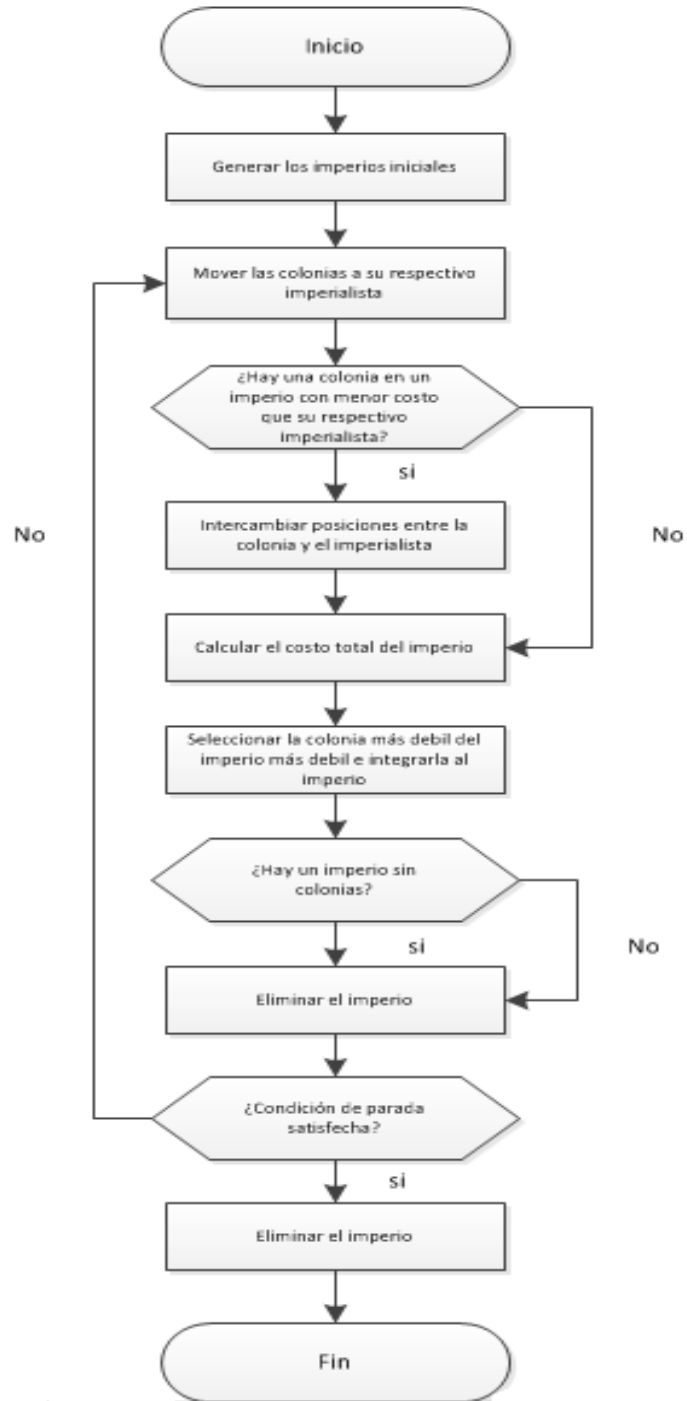
Hace referencia al mejoramiento de las colonias que emprende el imperialista, en él, una colonia se desplaza x unidades hacia su imperialista, la dirección del movimiento se define mediante un vector desde la colonia al imperialista. x es una variable aleatoria que sigue una distribución uniforme como se muestra a continuación:

$$x \sim U(0, \beta * d) \quad (7)$$

Donde β es un número mayor que 1 y d es la distancia entre la colonia y el imperialista. El hecho de que $\beta > 1$ causa que las colonias se acerquen al estado imperialista. Para ampliar la búsqueda de diferentes puntos alrededor del imperialista, se agrega una cantidad aleatoria de desviación de la dirección del movimiento denominada θ , la cual sigue una distribución uniforme entre $-\gamma \leq \theta \leq \gamma$, que corresponde a un parámetro que ajusta la desviación desde la dirección

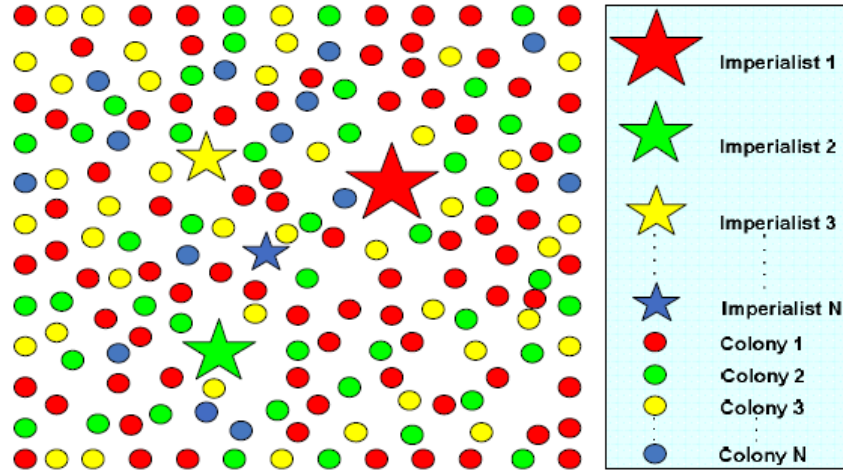
original.

Figura 1. Diagrama de flujo del algoritmo ICA



Fuente: ATASHPAZ, Esmail y LUCAS, Caro.

Figura 2. Población inicial de imperialistas y sus colonias



Fuente: ATASHPAZ, Esmail y LUCAS, Caro

3. Intercambio de posiciones entre el imperialista y una colonia.

Al efectuarse el movimiento de una colonia hacia su imperialista, ésta puede encontrar una posición con menor costo que la del imperialista, en ese caso, el imperialista se mueve a la posición de la colonia y viceversa.

4. Poder total de un imperio.

El poder de un imperio se ve afectado por el poder del país imperialista y de las colonias, sin importar que sea insignificante el poder de cada una de ellas. El costo total se define mediante la siguiente ecuación:

$$T.C_n = \text{costo}(\text{imperialista}_n) + \xi(\text{Costo}(\text{colonias del imperio})) \quad (8)$$

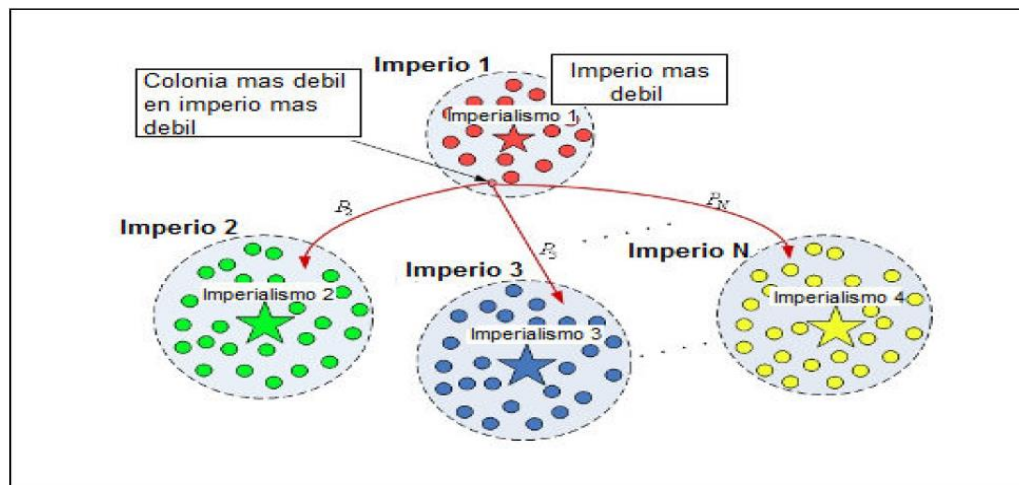
Donde $T.C_n$ es el costo total del n-ésimo imperio y ξ es un número positivo menor a 1, si éste valor es pequeño significa que el poder total del imperio dependerá exclusivamente del país imperialista, mientras que si éste valor es cercano a 1, aumentará la influencia de las colonias sobre el costo total del imperio.

5. Competencia imperialista.

A raíz de que los imperios tratarán de tomar posesión de las colonias de otros imperios y controlarlas, el desarrollo de la competencia entre los imperios tendrá como resultado la disminución del poder de los imperios más débiles y el incremento del poder de los más fuertes. En la figura 3 es posible observar el proceso de competencia imperialista.

El proceso inicia al seleccionar la colonia más débil del imperio más débil y hacer una competencia entre todos los imperios para poseer dicha colonia, el resultado de la competencia dependerá de la probabilidad que tenga un imperio de tomar posesión de una colonia la cual es proporcional al poder que tenga.

Figura 3. Competencia entre los imperialistas



Fuente: ATASHPAZ, Esmaeil y LUCAS, Caro. Op

6. Destrucción del imperio más débil

El imperio con el menor poder colapsará y sus colonias serán divididas entre los imperios restantes.

2.4. PROGRAMACIÓN DE OPERACIONES

La programación de operaciones denominada también como función de *scheduling*, se encarga de la asignación de recursos y actividades con el objetivo de optimizar una o más medidas de desempeño.

De acuerdo con lo expuesto por Heizer y Render, la importancia estratégica del *scheduling* radica en tres aspectos principales:

3. Movimiento más rápido de bienes y servicios a través de la instalación, lo cual significa mayor aprovechamiento de la capacidad instalada y por consiguiente, una reducción en los costos.
4. Capacidad adicional, facturación más rápida y relativa flexibilidad, lo que representa un mejor nivel de servicio al cliente.
5. Cumplimiento de compromisos y entregas fiables.

2.4.1. Clases de Schedule. Al hablar de problemas de *scheduling*, es necesario establecer la diferencia entre los términos *schedule* y secuencia. Una secuencia corresponde a una permutación del orden en que los trabajos se van a procesar en una máquina dada. Por otra parte, un *schedule* hace referencia a la asignación de puestos de trabajo en un entorno más complicado de ajuste de máquinas, permitiendo por ejemplo que un trabajo tenga derechos preferentes sobre otros asignados previamente a una máquina.

- *Schedule* sin retraso, es aquel en el que ninguna máquina se mantiene inactiva mientras una operación está esperando para ser procesada.
- *Schedule* activo, un *schedule* factible se denomina activo si no es posible construir otra secuencia realizando cambios en el orden de procesamiento en las máquinas con al menos una operación terminando antes de tiempo y ninguna operación terminando con retrasos.
- *Schedule* semi-activo, un *schedule* es semi-activo si una operación no puede

ser completada antes del tiempo programado sin que se cambie el orden de procesamiento en cualquiera de las máquinas.

2.4.2. Tipos de problema de *scheduling*. Los problemas de *scheduling* pueden ser clasificados de acuerdo a la configuración del sistema productivo, es decir la forma como las máquinas los operarios y centros de trabajo se organizan para procesar los productos, entre las diferentes configuraciones es posible encontrar³²

- *Project scheduling.*

En talleres organizados por proyectos, la programación de las operaciones se ocupa principalmente de la secuencia de las actividades sujetas a restricciones de precedencia y asignación de recursos. Para organizar el conjunto de actividades se utilizan técnicas de programación conocidas como PERT (Programa de Evaluación y Revisión técnica) que se utiliza en proyectos con actividades probabilísticas) y CPM (*Critical Path Method*), implementado en proyectos con actividades deterministas.

- *Single machine scheduling.*

Se refiere a la secuenciación de varios trabajos en una sola máquina. Algunas de las características de los trabajos a programar son tiempos de procesamiento.

- *Parallel machine scheduling.*

El problema de programación de máquinas paralelas es una generalización del problema de programación de una sola máquina, en él se considera que cada

³² YAGMAHAN, Betul y YENISEY, Mehmet Mutlu. Scheduling practice and recent developments in flow shop and job shop scheduling. En Computational intelligence in flow shop and job shop scheduling. Springer Berlin Heidelberg, 2009. p. 261-300.

trabajo puede ser procesado en cualquiera de las máquinas. En este problema se pueden identificar dos tipos de decisiones, la primera es determinar a qué máquinas asignar los trabajos y la segunda es determinar la secuencia de procesamiento de los trabajos asignados a cada máquina. Las máquinas pueden clasificarse en:

- a. Máquinas idénticas: todas las máquinas tienen las mismas especificaciones. Por consiguiente no existe diferencia en el procesamiento de los trabajos entre las diferentes máquinas.
 - b. Máquinas uniformes: las máquinas tienen diferentes velocidades de procesamiento (s_j), a su vez los trabajos tienen un requerimiento de procesamiento (p_i). El tiempo de procesamiento de una máquina requiere p_i/s_j unidades de tiempo.
 - c. Máquinas no relacionadas: cada trabajo tiene diferentes tiempos de procesamiento en diferentes máquinas.
- *Open shop scheduling.*

Sistema productivo en el cual hay m máquinas y cada trabajo debe ser procesado una vez en cada una de ellas, sin embargo en esta configuración el orden de procesamiento no es importante.

- *Flow shop scheduling.*

Consiste en un conjunto de máquinas que realizan diferentes tareas para el procesamiento de un trabajo, se caracteriza porque todos los trabajos tienen el mismo orden de procesamiento a través de las máquinas que están dispuestas de forma lineal, es decir, el procesamiento de un trabajo lo componen una lista ordenada de tareas en las que la i ésima tarea de cada puesto de trabajo está determinada por la misma máquina requerida y el tiempo de procesamiento en

él³³. El problema de *flow shop* parte de los siguientes supuestos:

- a. Un conjunto de n trabajos está disponible para su procesamiento en el momento cero.
- b. Los tiempos de alistamiento para las diferentes operaciones son dependientes de la secuencia.
- c. Las descripciones de los trabajos son conocidas en avance.
- d. Diferentes máquinas están continuamente disponibles.

2.5. FLOW SHOP HÍBRIDO

Es una generalización del problema de *flow shop* en la cual un conjunto de trabajos tiene que ser procesado en una serie de m etapas, cumpliendo con las siguientes condiciones:

- El número de etapas de procesamiento m es al menos 2.
- Cada etapa k tiene $m(k)$ máquinas en paralelo y en al menos una de las etapas $m(k) > 1$ máquinas.
- Todos los trabajos son procesados siguiendo el mismo orden de producción.
- Un trabajo puede omitir cualquier número de etapas siempre y cuando se procese en al menos una de las etapas globales.

El problema también se conoce como *flow shop* compuesto, *flow shop* flexible, o *flow shop* con múltiples procesadores, sin embargo, la denominación de *flow shop* híbrido es la más apropiada dado que el problema es un híbrido de dos sistemas productivos: una estación con múltiples máquinas en paralelo y el *flow shop* simple.

³³ BŁAŻEWICZ, Jacek, et al. Handbook on scheduling: from theory to applications. Springer Science & Business Media, 2007.

Notación del problema. Con base en la notación propuesta por Graham et al. (1979)³⁴, el problema de *flow shop* híbrido puede ser clasificado en función de los parámetros α, β, γ los cuales se describen a continuación.

α : Hace referencia al tipo y tamaño del taller, algunas de las clasificaciones son por ejemplo:

Fm: *flow shop* simple con m máquinas paralelas

F(k1,k2,...,kn): *flow shop* híbrido con k_i máquinas paralelas idénticas en la etapa i .

El parámetro α se compone de cuatro elementos, el primero de ellos α_1 indica la configuración general del taller, α_2 describe el número de etapas en el taller, α_3 y α_4 describen las propiedades de las máquinas en las diferentes etapas.

β : Enumera las características del taller, derivadas del problema de *flow shop* simple. Las más comunes son:

- Fechas de liberación (r_j): momento en el cual los trabajos pueden comenzar a procesarse.
- Precedencia (*prec*): ciertos trabajos deben ser completados antes de que otros puedan comenzar su procesamiento. Si un trabajo tiene por lo menos un predecesor y al menos un sucesor, se denomina como *chair* la restricción, si un trabajo tiene al menos un sucesor se denomina como *intree* y si un trabajo tiene al menos un predecesor se denominan *outtree*.

³⁴ BŁAŻEWICZ. Op. Cit. p., 5

- Preemption (*pmtn*): hace referencia a si el procesamiento de un trabajo en una máquina puede ser interrumpido e incluso posteriormente terminado en otra máquina.
- *No espera* (*no – wait*): los trabajos no pueden ser almacenados en zonas intermedias entre dos etapas de procesamiento sucesivas.

γ : Describe el criterio a ser minimizado o maximizado, en problemas de *flow shop* algunos de los objetivos comúnmente utilizados son:

- C_{max} : comúnmente conocido como *makespan*, corresponde al plazo máximo de ejecución de un trabajo.
- L_{max} (*maximum Lateness*): corresponde a la diferencia entre en tiempo de finalización de un trabajo y su fecha de vencimiento.
- Tardanza ponderada (*weighted Tardiness*): tiempo que un trabajo se excede después de la fecha de entrega planeada.

3. REVISIÓN DE LA LITERATURA

El problema de *flow shop* híbrido con máquinas paralelas idénticas ha sido ampliamente estudiado en la literatura y los primeros estudios se remontan al año 1954 cuando Johnson³⁵ lo aborda considerando dos y tres etapas de procesamiento. Sin embargo, el *flow shop* híbrido con máquinas paralelas no relacionadas no reporta muchos estudios en el medio académico. Entre los autores que estudian este problema se encuentran Toshifumi Uetake, Hitoshi Tsubone y Masaaki Ohba³⁶ en 1995 cuando lo consideran con dos etapas de producción. Desde ese entonces otros autores se han interesado en el problema, los cuales han realizado revisiones de la literatura, entre ellos Wepeng Wang³⁷ quien describe los métodos de solución comúnmente utilizados, Morais et al.,³⁸ concluyen que sólo el 3% de las investigaciones realizadas contemplan las máquinas no relacionadas. Pedro Gómez³⁹ plantea un marco general para el diseño e implementación de un algoritmo genético basado en un sistema multiagente para la programación de la producción en sistemas *flow shop* híbrido,

³⁵ JOHNSON, S.M. Optimal two-and three-stage production schedules with setup times included. En: Naval Research Logistics Quarterly. 1954, no 1. p. 61-68.

³⁶ UETAKE, Toshifumi, et al. A production scheduling system in a hybrid flow shop. En: International journal of production economics, 1995, vol. 41, no 1, p. 395-398.

³⁷ WANG, Wenpeng. Review on Hybrid Flow Shop Scheduling. En: Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on. IEEE, 2011. p. 7-10

³⁸ DE FATIMA MORAIS, Marcia, et al. Multicriteria hybrid flow shop scheduling problem: literature review, analysis, and future research. En Independent Journal of Management & Production, 2014, vol. 5, no 4, p. 1004-1031.

³⁹ GÓMEZ, Pedro. Un nuevo Algoritmo Genético Basado en un Sistema Multiagente para la Programación de la Producción en un Taller de Flujo Híbrido. En XI Congreso de Ingeniería de Organización. 2007. p. 1675-1685

y Linn y Zahng⁴⁰ clasifican el problema de acuerdo al número de estaciones que lo componen, es decir, dos, tres y múltiples etapas.

De acuerdo con lo expuesto por Linn y Zahng⁴¹, el *flow shop* híbrido de dos etapas presenta las siguientes sub-clasificaciones: una sola máquina en la primera etapa y m máquinas no relacionadas en la segunda, m máquinas no relacionadas en la primer etapa y una máquina en la segunda y, m máquinas no relacionadas en ambas etapas. Y el *flow shop* híbrido de múltiples etapas presenta dos variantes el *flow shop* híbrido flexible y el *flow shop* híbrido cíclico. Encontramos la primera subclasificación del *flow shop* híbrido de dos etapas en los estudios de Uetake et al.⁴², y posteriormente en los de Almeder y Hartl⁴³ quienes lo consideran a través de un caso de aplicación en la industria metalúrgica y, Figielska⁴⁴ que toma en cuenta la existencia de un *buffer* de capacidad limitada y reglas de preferencia de trabajos.

Figielska, en sus investigaciones de los años 2008⁴⁵ y 2009⁴⁶, considera restricciones como preferencias de trabajos, *buffer* entre etapas y recursos

⁴⁰LINN, Richard y ZHANG, Wei. Hybrid flow shop scheduling: a survey. En: Computers & industrial engineering, 1999, vol. 37, no 1, p. 57-61

⁴¹ Ibid., p. 59.

⁴² UETAKE, Toshifumi et al. Op. Cit., p. 395-398.

⁴³ ALMEDER, Christian y HARTL, Richard F. A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer. En: International Journal of Production Economics, 2013, vol. 145, no 1, p. 88-95.

⁴⁴ FIGIELSKA, Ewa. A heuristic for scheduling in a two-stage hybrid flowshop with renewable resources shared among the stages. En: European Journal of Operational Research, 2014, vol. 236, no 2, p. 433-444.

⁴⁵ FIGIELSKA, Ewa. A new heuristic for scheduling the two-stage flowshop with additional resources. En: Computers & Industrial Engineering, 2008, vol. 54, no 4, p. 750-763

⁴⁶ FIGIELSKA, Ewa. A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flowshop with additional resources. En: Computers & Industrial Engineering, 2009, vol. 56, no 1, p. 142-151

limitados y Low et al.⁴⁷ en el año 2008 tienen en cuenta restricciones de precedencia y de elegibilidad de máquinas, presentan la segunda variante del *flow shop* de dos etapas. Asimismo la clasificación de m máquinas no relacionadas en ambas etapas es investigada por Figielska⁴⁸, quien tiene en cuenta así como en sus trabajos anteriores la existencia de *buffers* de capacidad ilimitada y el requerimiento de recursos para el procesamiento de las tareas. Y Rabiee et al.⁴⁹, que presentan la restricción de *no-wait* en la cual no se permiten los tiempos de espera para la transferencias de los trabajos entre etapas, además de considerar los tiempos de alistamiento dependientes de la secuencia.

Por otra parte y presentando la segunda clasificación del problema, es decir, *flow shop* híbrido de tres etapas, se encuentran Yaurima et al., quienes en 2008⁵⁰ realizan estudios en industrias de fabricación de televisores con circuito impreso a bordo, y luego en 2009⁵¹ incluyen las zonas de almacenamiento intermedias o *buffers* con capacidad limitada, luego están Li et al.⁵², que en 2015 abordan el

⁴⁷ LOW, Chinyao et al. A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines. En: Computers & Operations Research, 2008, vol. 35, no 3, p. 845-853

⁴⁸ FIGIELSKA, Ewa. Heuristic algorithms for preemptive scheduling in a two-stage hybrid flowshop with additional renewable resources at each stage. En: Computers & Industrial Engineering, 2010, vol. 59, no 4, p. 509-51

⁴⁹ RABIEE, M., et al. An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines. En: The International Journal of Advanced Manufacturing Technology, 2014, vol. 71, no 5-8, p. 1229-1245

⁵⁰ YAUURIMA, Victor, et al. Hybrid flowshop with unrelated machines, sequence dependent setup time and availability constraints: An enhanced crossover operator for a genetic algorithm. Parallel processing and applied mathematics. En: Springer Berlin Heidelberg, 2008. p. 608-617.

⁵¹ YAUURIMA, Victor et al. Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. En: Computers & Industrial Engineering, 2009, vol. 56, no 4, p. 1452-1463.

⁵² LI, Zhan-tao, et al. Approximation algorithms for the three-stage flexible flow shop problem with mid group constraint. En: Expert Systems with Applications, 2015, vol. 42, no 7, p. 3571-3584.

problema considerando restricciones de grupo en la segunda etapa y proponen 10 algoritmos: nueve de ellos basados en el algoritmo *Shop Partitioning (SP)* o enfoque de descomposición orientado a las etapas, y uno basado en el algoritmo *Three Stage (TS)* es decir, el enfoque holístico que considera el problema de programación completo de forma integrada.

Al hablar del problema de *flow shop* híbrido con más de tres etapas, es posible encontrar los trabajos desarrollados por autores como Nagano y Mocellin quienes proporcionan dos métodos heurísticos para la solución del problema presentado como dos *flow shop* tradicionales funcionando en paralelo, posteriormente Chinyao Low⁵³, propone una heurística basada en el algoritmo de recocido simulado considerando además la restricción de tiempos de preparación independientes de la secuencia y Jitti et al., quienes realizan varias investigaciones en los años 2006⁵⁴, 2008⁵⁵, y 2009⁵⁶ considerando en cada una de ellas las mismas restricciones de tiempos de alistamiento independientes de la secuencia y de la máquina, y utilizando diferentes métodos de solución.

Son varios los investigadores que se enfocaron en esta nueva distribución del *flow shop* teniendo en cuenta diversas restricciones y métodos de solución. Unas de las restricciones mayormente consideradas es la de tiempos de alistamiento

⁵³ LOW, Chinyao. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. En: Computers & Operations Research, 2005, vol. 32, no 8, p. 2013-2025

⁵⁴ JUNGWATTANAKIT, Jitti, et al. Constructive and simulated annealing heuristics for hybrid flow shops with unrelated parallel machines. En: Proceedings of the 3rd OR-CRN Operations Research Conference, Bangkok, Thailand. 2006. p. 110-121.

⁵⁵ JUNGWATTANAKIT, Jitti, et al. Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: The International Journal of Advanced Manufacturing Technology, 2008, vol. 37, no 3-4, p. 354-370

⁵⁶ JUNGWATTANAKIT, Jitti, et al. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: Computers & Operations Research, 2009, vol. 36, no 2, p. 358-378.

dependientes de la secuencia contemplada por Ruiz y Maroto⁵⁷ en un caso de aplicación de fabricación de baldosas cerámicas; Parra y Mejía⁵⁸ en un caso en la industria ladrillera, Jenabi et al.⁵⁹ en un problema de lote económico de programación, Defersha y Chen⁶⁰ que establecen que pueden ser anticipatorios o no anticipatorios; y Lugo y Teixeira⁶¹ quienes además tienen en cuenta la restricción de elegibilidad de máquinas.

Igualmente los métodos de solución mayormente utilizados en esta clasificación, son los algoritmos basados en población, en este aspecto Xu et al.⁶², proponen la modificación del algoritmo memético planteado por Eusiff y Lansey utilizando una estrategia metalamarkiana⁶³ para establecer la búsqueda a utilizar en cada generación para encontrar solución al problema. Así mismo Gómez Gasquet et

⁵⁷ RUIZ, Ruben y MAROTO, Concepción. A genetic algorithm for hybrid flow shops with sequence dependent setup times and machine eligibility. En: European Journal of Operational Research, 2006, vol. 169, no 3, p. 781-800.

⁵⁸ PARRA Javier y MEJÍA, Gonzalo. Desarrollo de un modelo para la secuenciación de trabajos en la mediana industria ladrillera de la localidad XIX. 2006

⁵⁹ JENABI, Masoud, et al. Two hybrid meta-heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines. En: Applied Mathematics and Computation, 2007, vol. 186, no 1, p. 230-245.

⁶⁰ DEFERSHA, Fantahun y CHEN, Mingyuan. Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem. En: The International Journal of Advanced Manufacturing Technology, 2012, vol. 62, no 1-4, p. 249-265.

⁶¹ LUGO, Pedro, et al. Um modelo de programação inteira mista para a programação da produção em flowshop híbrido com buffers limitados. En: Simpósio Brasileiro de Pesquisa Operacional (45: 16-19 septiembre, Natal, Brasil). Publicaciones. Universidad Federal de Sao Carlos. 2013, p. 154-165.

⁶² XU, Ye, et al.. An effective shuffled frog leaping algorithm for solving hybrid flow-shop scheduling problem. En: Advanced Intelligent Computing. Springer Berlin Heidelberg, 2012. p. 560-567.

⁶³ Según Ong et al, 2004, corresponde a una simple coordinación aleatoria de memes sin adaptación

al.⁶⁴, utilizan un algoritmo genético de programación multiagente *MAGSA*, también Lugo y Teixeira⁶⁵, plantean un algoritmo genético modificando el operador de mutación por un operador de búsqueda local. Bajo este enfoque, Rahman et al.⁶⁶ implementan, un algoritmo de evolución diferencial con la heurística cuello de botella. También se desarrollan algoritmos híbridos como el planteado por Li y Pan⁶⁷ entre la búsqueda tabú y el algoritmo de colonia artificial de abejas. Luego, Costa et al.⁶⁸ estudian la variante de que algunas máquinas pueden procesar de forma simultánea algunos grupos de trabajos y plantean un algoritmo genético para su solución; también Jun y Park⁶⁹ presentan un caso de aplicación en la industria de fabricación de transformadores y plantean un algoritmo híbrido entre el algoritmo genético y las reglas *NEH* (desarrollada por Newaz, Enscore y Ham, de quienes toma su nombre) y *EDD* (fecha de entrega más temprana).

Algunas de las variantes del problema se debe a consideraciones de algunos

⁶⁴ GÓMEZ, Pedro et al. An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimise makespan. *En: Expert Systems with Applications*, 2012, vol. 39, no 9, p. 8095-8107

⁶⁵ LUGO, Pedro, et al. Algoritmo genético com busca local para a programação da produção em sistemas flow-shop híbridos. *En: Congreso Latino-Iberoamericano de Investigación Operativa* (44:24-28, septiembre, Rio de Janeiro, Brasil). Publicaciones. Universidad Federal de Sao Carlos. 2012, p. 19-29.

⁶⁶ RAHMAN, Riza et al. Hybrid Differential Evolution and Bottleneck Heuristic Algorithm to Solve Bi-Objective Hybrid Flow Shop Scheduling Unrelated Parallel Machines Problem.

⁶⁷ LI, Jun-qing y PAN, Quan-ke. Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. *En: Information Sciences*, 2014.

⁶⁸ COSTA, Antonio et al. A novel genetic algorithm for the hybrid flow shop scheduling with parallel batching and eligibility constraints. *En: The International Journal of Advanced Manufacturing Technology*, 2014, vol. 75, no 5-8, p. 833-847

⁶⁹ JUN, Sungbum y PARK, Jinwoo. A hybrid genetic algorithm for the hybrid flow shop scheduling problem with nighttime work and simultaneous work constraints: A case study from the transformer industry. *En: Expert Systems with Applications*, 2015, vol. 42, no 15, p. 6196-6204.

autores, por ejemplo, Chen y Chen⁷⁰ tienen en cuenta la existencia de una etapa cuello de botella, Dai et al.⁷¹, plantean un modelo de eficiencia energética buscando optimizar la utilización de las máquinas. Otros autores como Elmi y Topaloglu⁷² estudian la variante denominada “*Blocking hybrid flow shop robotic cell scheduling*”.

Para concluir, el *flow shop* híbrido con múltiples etapas puede ser clasificado en *flow shop* híbrido flexible, sistema productivo que se caracteriza porque los trabajos pueden omitir algunas de las etapas de procesamiento, tal variación ha sido estudiada por Ruiz et al.⁷³, Zandieh et al.⁷⁴, Borzogirad y Logendran⁷⁵, Attar et al.⁷⁶, Siqueira et al.⁷⁷, Li et al.⁷⁸, y Arango et al.⁷⁹ y *flow shop* híbrido reentrante

⁷⁰ CHEN, Chun-Lung y CHEN, Chuen-Lung. A bottleneck-based heuristic for minimizing makespan in a flexible flow line with unrelated parallel machines. En: Computers & Operations Research, 2009, vol. 36, no 11, p. 3073-3081

⁷¹ DAI, Min, et al. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. En: Robotics and Computer-Integrated Manufacturing, 2013, vol. 29, no 5, p. 418-429

⁷² ELMI, Atabak y TOPALOGLU, Seyda. A scheduling problem in blocking hybrid flow shop robotic cells with multiple robots. En: & Operations Research, 2013, vol. 40, no 10, p. 2543-2555.

⁷³ RUIZ, Rubén et al. Modeling realistic hybrid flexible flowshop scheduling problems. En: Computers & Operations Research, 2008, vol. 35, no 4, p. 1151-1175

⁷⁴ ZANDIEH, M., et al. Mansour. A robust genetic algorithm for scheduling realistic hybrid flexible flow line problems. En: Journal of Intelligent Manufacturing, 2010, vol. 21, no 6, p. 731-743

⁷⁵ BOZORGIRAD, Mir Abbas y LOGENDRAN, Rasaratnam. Hybrid flowshop scheduling problem with a bi-criteria objective and group technology assumption. En: IIE Annual Conference. Proceedings. Institute of Industrial Engineers-Publisher, 2013. p. 3004

⁷⁶ ATTAR, S., et al. Hybrid flexible flowshop scheduling problem with unrelated parallel machines and limited waiting times. En: The International Journal of Advanced Manufacturing Technology, 2013, vol. 68, no 5-8, p. 1583-1599

⁷⁷ DE SIQUEIRA, Eduardo, et al. um algoritmo iterated greedy search aplicado à minimização do makespan no problema flowline híbrido e flexível

o *flow shop* híbrido cíclico considerada por Kim y Lee⁸⁰ que evalúan la tardanza total y el *makespan*; así mismo Soltani y Karimi⁸¹ que consideran el problema sujeto a restricciones de elegibilidad de máquinas y proponen un algoritmo de recocido simulado y un genético para la solución.

⁷⁸ LI, Zhan-tao, et al. Scheduling rules for two-stage flexible flow shop scheduling problem subject to tail group constraint. En: International Journal of Production Economics, 2013, vol. 146, no 2, p. 667-678

⁷⁹ ARANGO-MARÍN, Jaime Antero, et al. Applying TOC Heuristics to Job Scheduling in a Hybrid Flexible Flow Shop. En: Dyna, 2014, vol. 81, no 186, p. 113-119

⁸⁰ KIM, H. W y LEE, D. H. Heuristic algorithms for re-entrant hybrid flow shop scheduling with unrelated parallel machines. Proceedings of the Institution of Mechanical Engineers, En: Part B: Journal of Engineering Manufacture, 2009, vol. 223, no 4, p. 433-442.

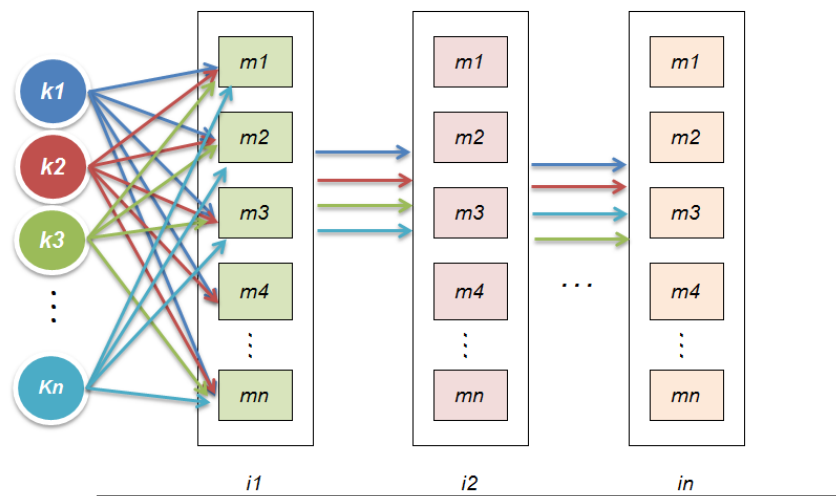
⁸¹ SOLTANI, S. Abolfazl y KARIMI, Behrooz. Cyclic hybrid flow shop scheduling problem with limited buffers and machine eligibility constraints. En: The International Journal of Advanced Manufacturing Technology, 2014, p. 1-17

4. DISEÑO DEL MODELO MATEMÁTICO

4.1. DESCRIPCIÓN DEL SISTEMA PRODUCTIVO BAJO ESTUDIO

El sistema productivo a analizar corresponde a un *flow shop* híbrido con máquinas paralelas no relacionadas, el cual está dedicado a la fabricación de diferentes productos que deben seguir un flujo unidireccional a lo largo del sistema y por consiguiente deben ser procesados en cualquiera de las máquinas existentes en cada una de las etapas de procesamiento (conocidas también como centros de trabajo) requeridas para obtener el producto final, tal como se muestra en la figura 4.

Figura 4. Sistema productivo bajo estudio



En las diferentes etapas existentes en el *flow shop* híbrido objeto de estudio, se cuenta con mínimo dos máquinas paralelas no relacionadas, cuyas características y tiempos de procesamiento difieren entre sí. Estas máquinas pueden procesar todos los trabajos.

En relación con lo descrito anteriormente, los tiempos de procesamiento de cada trabajo son considerados dependientes de la máquina y a su vez los tiempos de alistamiento se contemplan dependientes de la secuencia, es decir, dependen del último trabajo asignado a la máquina. Los tiempos de procesamiento son conocidos y siguen una distribución uniforme entre 75 y 125, de acuerdo con lo expuesto por Lugo et al.⁸², por su parte los tiempos de alistamiento tienen una distribución uniforme entre 25 y 74.

4.2. MODELO MATEMÁTICO

Con el objetivo de resolver el problema de programación de operaciones del sistema productivo bajo estudio, se desarrolló un modelo de programación lineal entera mixta que busca minimizar el *makespan*, teniendo en cuenta los siguientes supuestos:

- Existen al menos dos máquinas paralelas no relacionadas por etapa.
- Todas las máquinas pueden procesar todos los trabajos.
- Un trabajo sólo puede ser asignado a una máquina en cada una de las etapas de procesamiento.
- Una máquina no puede procesar más de un trabajo a la vez.
- No hay límites de trabajos a ser asignados a una máquina.
- Todas las máquinas están disponibles en el instante $t = 0$.
- Para cada una de las máquinas se considera un trabajo ficticio cero, el cual corresponde a un calentamiento inicial de la máquina y dependerá del primer trabajo asignado.
- Los tiempos de alistamiento son dependientes de la secuencia.

⁸² LUGO, Pedro, et al. Um modelo de programação inteira mista para a programação da produção em flowshop híbrido com buffers limitados. En: Simpósio Brasileiro de Pesquisa Operacional (45: 16-19 septiembre, Natal, Brasil). Publicaciones. Universidad Federal de Sao Carlos. 2013, p. 154-165.

- Una vez terminado el procesamiento de un trabajo en una etapa, este debe ser asignado a cualquiera de las máquinas disponibles en la etapa siguiente, en el caso de no existir máquinas disponibles puede esperar en una zona de almacenamiento.

4.2.1. Modelo de programación lineal entera mixta

A continuación se presentan cada uno de los conjuntos, parámetros y variables consideradas en el modelo matemático

- Conjuntos

j, k, h : subíndices para los trabajos.

i : subíndice para las etapas.

m : subíndice para las máquinas

- Parámetros

$S(k, j, m, i)$: tiempo de alistamiento del trabajo k precedido por el trabajo j en la máquina m de la etapa de procesamiento i .

$P(k, m, i)$: tiempo de procesamiento del trabajo k en la máquina m de la etapa de procesamiento i .

U : es un escalar que representa una penalización la cual es mayor a la suma de los tiempos de procesamiento de todos los trabajos en todas las máquinas.

- Variables

$Y(k, j, m, i)$: es una variable binaria que permite realizar la asignación de los trabajos a las máquinas en cada una de las etapas de procesamiento. La variable tendrá un valor de 1, cuando el trabajo k asignado a la máquina m de la etapa i siendo precedido por el trabajo j , y de 0 en el caso contrario.

$c(k, i)$: tiempo en que se completa el trabajo k en la etapa de procesamiento i . Incluye los tiempos de procesamiento y de alistamiento

C_{max} : es la variable a optimizar, en este caso el *makespan*, que corresponde al instante en el que finaliza el procesamiento del último trabajo.

- Modelo matemático

$$\text{Min } C_{max} \quad (9)$$

Sujeto a:

$$\sum_m \sum_j Y_{k,j,m,i} = 1 \quad \forall k, i; \quad j \neq k; \quad k \neq 0 \quad (10)$$

$$\sum_m \sum_j Y_{k,j,m,i} \leq 1 \quad \forall k, i; \quad j \neq k; \quad k \neq 0 \quad (11)$$

$$\sum_h Y_{j,h,m,i} \geq Y_{k,j,m,i} \quad \forall k, i, m, j; \quad j \neq k; \quad j \neq h; \quad k \neq h; \quad k \neq 0 \quad (12)$$

$$\sum_k Y_{k,j,m,i} \leq 1 \quad \forall j, i, m; \quad j \neq k; \quad k \neq 0; \quad j = 0 \quad (13)$$

$$\sum_m (Y_{k,j,m,i} + Y_{j,k,m,i}) \leq 1 \quad \forall k, j, i; \quad k \neq j; \quad k \neq 0; \quad (14)$$

$$C_{k,i} = 0 \quad \forall i; \quad k = 0 \quad (15)$$

$$C_{k,i} + U(1 - Y_{k,j,m,i}) \geq C_{j,i} + Y_{k,j,m,i}(S_{k,j,m,i} + P_{k,m,i}) \quad \forall k, i, m, j; j \neq k; k \neq 0 \quad (16)$$

$$C_{k,i} - C_{k,i-1} \geq Y_{k,j,m,i}(S_{k,j,m,i} + P_{k,m,i}) \quad \forall k, i, m, j; j \neq k; k \neq 0; i \neq 1 \quad (17)$$

$$C_{max} \geq C_{k,i} \quad \forall k, i \quad i = \text{última estación} \quad (18)$$

La ecuación (9) define la función objetivo del problema, la cual corresponde a la minimización del *makespan*. La ecuación (10) evita la división de un trabajo en lotes de procesamiento, puesto que asegura que éste sólo sea asignado a una máquina en cualquier etapa de procesamiento. La ecuación (11) garantiza que si una tarea es asignada a cualquiera de las máquinas disponibles en cualquier etapa, entonces dicho trabajo deberá tener como máximo un sucesor en esa máquina. La ecuación (12) afirma que un trabajo debe tener al menos un predecesor en la máquina a la cual fue asignado. La ecuación (13) garantiza que el trabajo ficticio cero debe preceder como máximo a un trabajo en cualquier máquina de cualquier etapa de procesamiento. La ecuación (14) evita la ocurrencia de precedencias cruzadas, es decir, si al trabajo k lo precede el trabajo j, entonces al trabajo j no lo puede preceder el trabajo k. La ecuación (15) asegura que el tiempo de procesamiento del trabajo ficticio sea cero.

Las ecuaciones (16) y (17) permiten calcular el instante en que un trabajo culmina su procesamiento en cualquiera de las etapas. La ecuación (16) indica que el instante en que un trabajo termina su procesamiento en una etapa se ve afectado por el procesamiento del trabajo que lo precede, esto aplica solo para la primera etapa de procesamiento.

En el caso de las etapas subsiguientes, existen dos escenarios que pueden afectar el instante de finalización, el primero de ellos, en el cual el instante de finalización de la tarea que precede al trabajo k es mayor al instante en que finalizó el procesamiento del trabajo k en la etapa inmediatamente anterior, entonces el instante de finalización del precedente será el que se tiene en cuenta

para el cálculo del instante de finalización del trabajo k en dicha etapa lo cual es posible calcular mediante la ecuación (16). El segundo escenario es aquel en el cual el instante de finalización de la tarea precedente al trabajo k en la máquina m de la etapa i es menor al instante de finalización del trabajo k en la etapa anterior, por consiguiente el instante a considerar para el cálculo del instante de finalización del trabajo k en la etapa i corresponderá al de la etapa $i-1$, como se expresa en la ecuación (17). Estos dos últimos escenarios es posible observarlos en las figuras 5 y 6 respectivamente.

Figura 5. Caso $C_{j,i} \geq C_{k,i-1}$

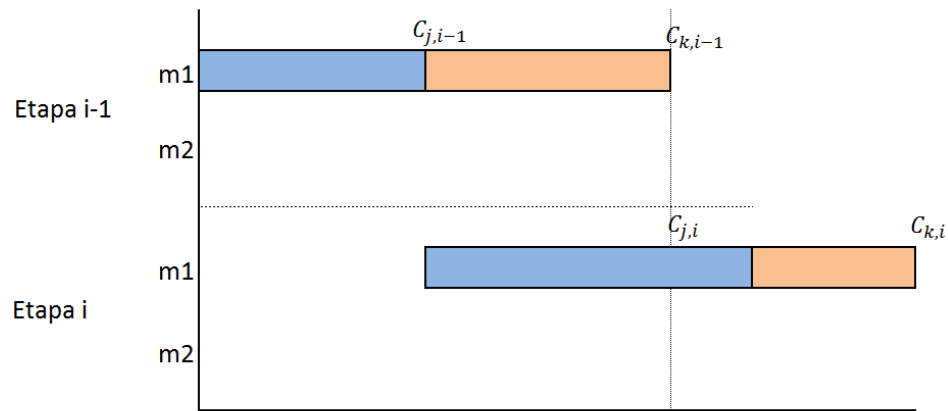
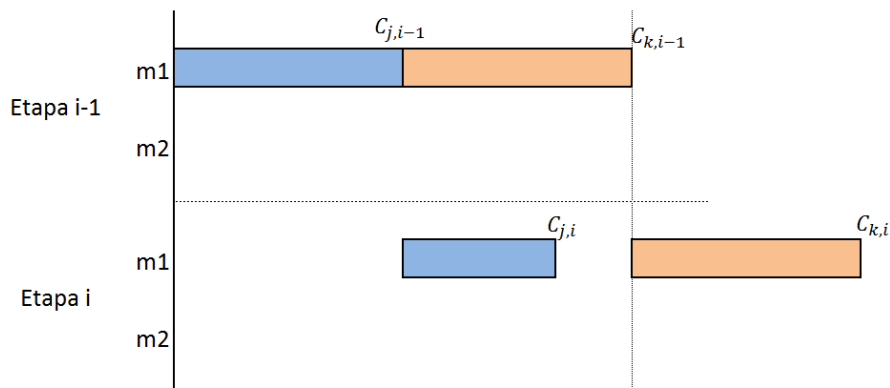


Figura 6. Caso 2 $C_{j,i} \leq C_{k,i-1}$



En el anexo A, se presenta la formulación del modelo descrito anteriormente en el lenguaje GAMS, para una instancia de 5 trabajos, 3 etapas de procesamiento y dos máquinas por etapa.

4.3. ANÁLISIS DE RESULTADOS

Una vez planteado el modelo de programación lineal entera mixta y escrito en lenguaje GAMS, fue ejecutado utilizando el solver *CPLEX* 12, en un computador con procesador Intel Core i5 3.2 GHz con 8 Gigabytes de memoria RAM y un sistema operativo de 64 bytes, se utilizaron 12 instancias las cuales se encuentran en el anexo B; el proceso de validación del modelo se describe en el anexo C, que corresponde a una guía para implementar en forma adecuada modelos de programación en el software Gams.

Dado que el tiempo de ejecución de una de las instancias de mayor complejidad (7 trabajos, 5 etapas de procesamiento y 2 máquinas por etapa) fue de 584847,52 segundos y el gap entre la solución obtenida y la óptima fue de 55.91%, se fijó 10800 segundos como tiempo máximo de ejecución del modelo, teniendo en cuenta que dada la complejidad del problema, en estudios previos ya se había fijado un límite de tiempo, como es el caso del modelo desarrollado por Lugo et al, quienes establecieron un tiempo límite de 3600 segundos.

Los resultados obtenidos son presentados en la tabla 1, donde se muestra para cada una de las instancias la calidad de la solución obtenida en el tiempo límite (% óptimo) y la diferencia existente entre la solución obtenida y la solución óptima (gap).

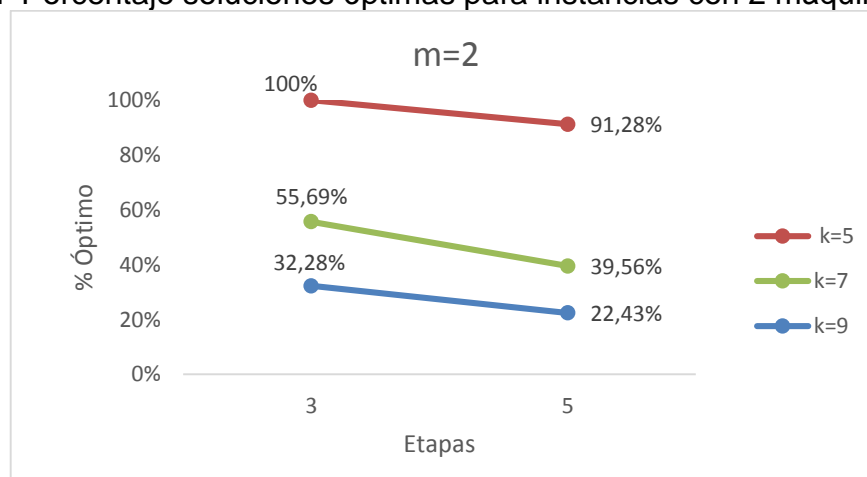
Tabla 1. Porcentaje soluciones óptimas y gap

		I			
		3		5	
k	m	2	3	2	3
5	gap	0,00%	0,00%	8,72%	39,15%
7	gap	44,31%	35,70%	60,44%	44,31%
9	gap	67,72%	72,74%	77,57%	82,15%

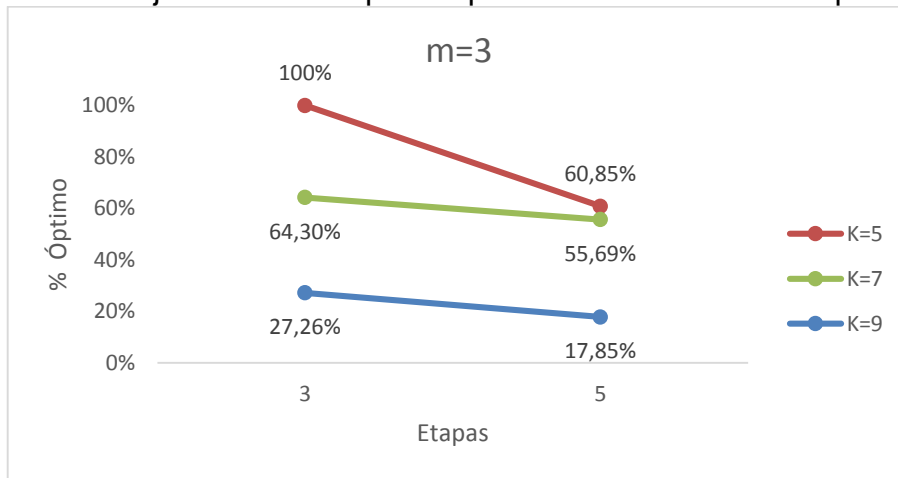
Como se observa en la tabla anterior, las instancias k5m2i3 (5 trabajos, 2 máquinas y 3 etapas de procesamiento) y k5m3i3 (5 trabajos, 3 máquinas y 3 etapas de procesamiento), son las únicas cuya solución obtenida dentro del tiempo límite es la óptima. A medida que aumenta el número de trabajos a ser procesados, las máquinas disponibles por etapa y las etapas de procesamiento, la complejidad del problema aumenta, lo que se ve representado en la calidad de las soluciones obtenidas, la cual disminuye al aumentar cualquiera de los factores nombrados previamente. Este comportamiento evidencia la dificultad del problema para encontrar una solución óptima en un tiempo computacionalmente razonable.

En las gráficas 1 y 2, es posible observar dicho comportamiento cuando se tienen dos y tres máquinas disponibles en cada etapa.

Gráfica 1. Porcentaje soluciones óptimas para instancias con 2 máquinas.



Gráfica 2. Porcentaje soluciones óptimas para instancias con 3 máquinas.



5. DISEÑO DEL ALGORITMO

El algoritmo híbrido está basado en la modificación realizada por Rabiee et al.⁸³ al algoritmo competitivo imperialista planteado por Atashpaz-Gargari y Lucas⁸⁴; se propone combinar el algoritmo mencionado anteriormente y los procesos de selección, cruce y mutación del algoritmo genético.

5.1. DESCRIPCIÓN DETALLADA DEL ALGORITMO

5.1.1. Generación de la población inicial. La población inicial de la metaheurística se obtiene al generar x cantidad de países de forma aleatoria. Un país es un vector de $1 \times [(k + m)i - 1]$ y representa la secuencia de procesamiento de trabajos en cada una de las máquinas y etapas del proceso productivo, k representa el número de trabajos a programar y m el número de máquinas existentes en cada una de las etapas de procesamiento i .

Las posiciones correspondientes a las etapas se calculan mediante la expresión $(k + m)(i - 1)$, y la asignación de máquinas y trabajos a cada una de las etapas se hace mediante una distribución uniforme entre 1 y $(k + m - 1)$.

A continuación en la figura 7, se presenta un ejemplo de generación de un país para un total de 5 trabajos, 3 etapas y 2 máquinas paralelas no relacionadas por estación.

$$k = 5, m = 2, i = 3$$

⁸³ RABIEE, M., et al. An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines. The International Journal of Advanced Manufacturing Technology, 2014, vol. 71, no 5-8, p. 1229-1245

⁸⁴ ATASHPAZ-GARGARI, Esmail y LUCAS, op cit. p. 4661-4667.

Vector: $1 \times [(k + m)i - 1]$

$$1 \times [(5 + 2)3 - 1] = 1 \times 20$$

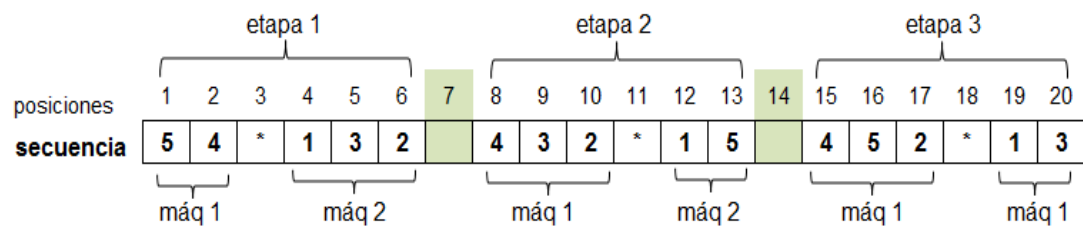
Posiciones de las etapas: $(k + m)(i - 1)$

Posición etapa 1: $(5 + 2)(1 - 1) = 0$

Posición etapa 2: $(5 + 2)(2 - 1) = 7$

Posición etapa 3: $(5 + 2)(3 - 1) = 14$

Figura 7. Ejemplo de generación de un país para k5m2i3



- Cálculo del costo

Después de generar los países que conformaran la población inicial, se procede a calcular el costo de cada uno de ellos, el cual representa el valor de la variable a optimizar. En esta investigación la variable es el *makespan* o tiempo máximo de finalización. Continuando con el ejemplo anterior, se calcula el costo o valor del *makespan* de la secuencia mostrada en la figura 7.

Tabla 2. Tiempos (u.t) de procesamiento para k5m2i3

	<i>i1</i>		<i>i2</i>		<i>i3</i>	
	<i>m1</i>	<i>m2</i>	<i>m1</i>	<i>m2</i>	<i>m1</i>	<i>m2</i>
k1	16	11	15	11	19	18
k2	16	15	12	10	11	19
k3	19	17	20	19	15	14
k4	15	19	19	13	17	15
k5	11	15	12	10	10	15

Tabla 3. Tiempo (u.t) de alistamiento para k5m2i3

		<i>i1</i>					<i>i2</i>					<i>i3</i>																			
		<i>m1</i>		<i>m2</i>			<i>m1</i>		<i>m2</i>			<i>m1</i>		<i>m2</i>																	
<i>k</i>		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5					
<i>j</i>	1		9	7	3	6		4	3	7	6		3	8	2	8		1	3	7	1		7	5	8	6		4	2	4	2
	2	5		2	5	2	2		6	7	7	8		1	8	7	4		3	7	4	4		7	6	7	8		2	1	5
	3	4	1		6	5	2	9		2	7	7	2		5	8	6	6		2	1	7	2		7	5	3	6		5	1
	4	5	2	5		4	6	2	8		8	1	4	4		2	5	5	3		6	9	5	8		7	3	4	7		2
	5	7	1	6	1		4	8	3	3		8	7	8	7		8	8	2	8		4	6	2	8		6	7	3	4	
	0	9	5	9	2	8	5	6	3	5	3	5	5	8	7	7	6	3	3	6	4	6	1	1	3	1	3	7	4	4	6

El *makespan* o en su defecto el costo total de este país o secuencia es de 116, tal como se puede observar en la tabla 4 y en la gráfica 3, este valor está dado por el tiempo de completamiento del trabajo 2 en la máquina 1 de la etapa 3.

- Identificación de los imperialistas

Una vez calculado el costo de cada uno de los países, se identifican los imperialistas, es decir aquellas secuencias cuyo valor de *makespan* sean mínimos dentro del conjunto de países que conforman la población inicial. Los países o secuencias restantes se denominan colonias.

Posteriormente se calcula el costo normalizado mediante la ecuación mostrada a continuación:

$$C_n = \max_i c_i - c_n \quad (19)$$

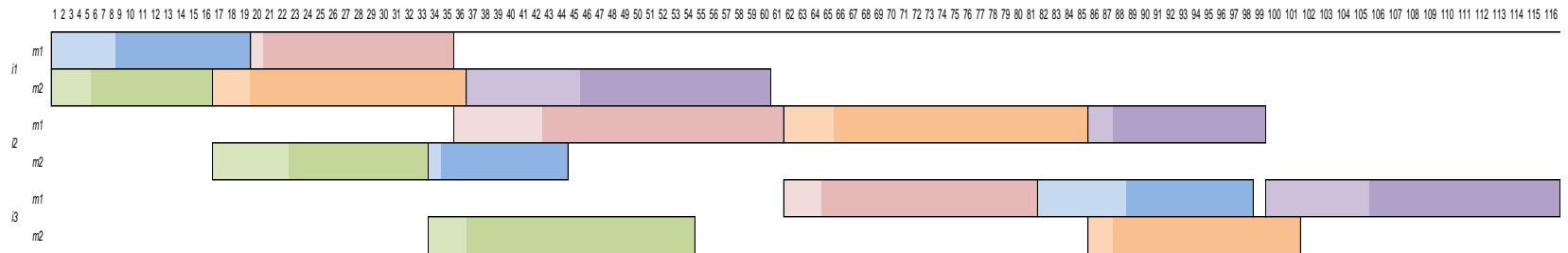
Dónde:

C_n es el costo normalizado,
 c_n el costo del *n*-ésimo imperialista y
 c_i el máximo costo dentro de las colonias.

Tabla 4. Cálculo del costo de un país

	etapa 1					etapa 2					etapa 3																			
	máquina 1		máquina 2			máquina 1			máquina 2		máquina 1			máquina 2																
sec	5	4	1	3	2	4	3	2	1	5	4	5	2	1	3															
	8	11	1	15	5	11	3	17	9	15	7	19	4	20	2	12	6	11	1	10	3	17	7	10	6	11	3	18	2	14
ct	19	35	16	36	60	61	85	99	33	44	81	98	116	54	101															
<i>makespan= 116</i>																														

Gráfica 3. Makespan o costo total del país



El costo normalizado permite calcular el poder normalizado, el cual corresponde al porcentaje de colonias que poseerá el imperialista y con las cuales compartirá sus características, dicho poder es inversamente proporcional al costo normalizado y se calcula mediante la siguiente ecuación:

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (20)$$

Donde C_i es el costo normalizado del i -ésimo imperialista y P_n el poder normalizado.

- Formación de imperios iniciales

La formación de los imperios iniciales se hace con base en el poder normalizado, para ello se calcula el número de colonias a asignar a cada imperialista, las cuales son seleccionadas mediante una distribución uniforme entre 1 y el número de colonias existentes. Las colonias que debe tener cada imperialista se calcula mediante la siguiente expresión:

$$NC_n = \text{round}\{P_n \cdot N_{col}\} \quad (21)$$

Dónde:

N_{col} = número total de colonias

NC_n = número de colonias a asignar a cada imperialista.

Una vez asignadas todas las colonias a los imperialistas se procede al cálculo del poder total del imperio, que depende en gran proporción del poder del país

imperialista, sin embargo las colonias también tienen un efecto, el cual generalmente no es tan significativo.

5.1.2. Movimiento de las colonias hacia su imperialista. Tras conocer cuántas colonias pertenecen a cada imperialista, se procede a la formación del imperio. Un imperio está conformado por un imperialista y sus respectivas colonias, las cuales son seleccionadas de manera aleatoria dentro del conjunto de países restantes luego de elegir los imperialistas.

- Cálculo del costo total del imperio

El costo total de un imperio está dado por el poder del imperialista más un porcentaje del poder del costo de sus colonias:

$$T_c = \text{costo}(\text{imperialista}) + \xi [\text{prom. costo}(\text{colonias})] \quad (22)$$

Donde T_c es el costo total del imperio y ξ una constante entre 0 y 1.

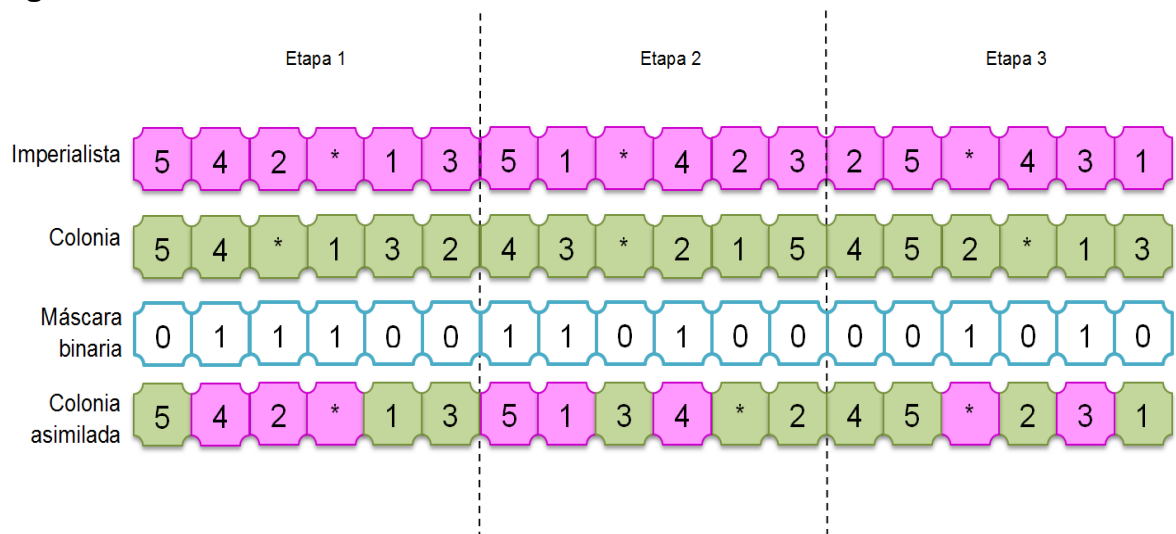
- Asimilación

Los imperialistas son las mejores soluciones y tienen la intención de imponer parte de sus características a las colonias pertenecientes a su dominio; en otras palabras, un imperialista comparte información con sus colonias. Por consiguiente, alguna parte de la estructura de las colonias será similar a la de su respectivo imperialista.

Para este paso, cada una de las colonias se seleccionan para copiar información del imperialista y con este propósito se crea un nuevo vector del mismo tamaño y se generan valores entre 0 y 1 (máscara binaria) en cada una de sus posiciones.

Las posiciones resultantes en 1 son copiadas a la colonia asimilada directamente del imperialista y las posiciones restantes, es decir las que tienen valor de 0, son copiadas de las colonias según como falten, tal como se muestra en la figura 8.

Figura 8. Procedimiento de asimilación



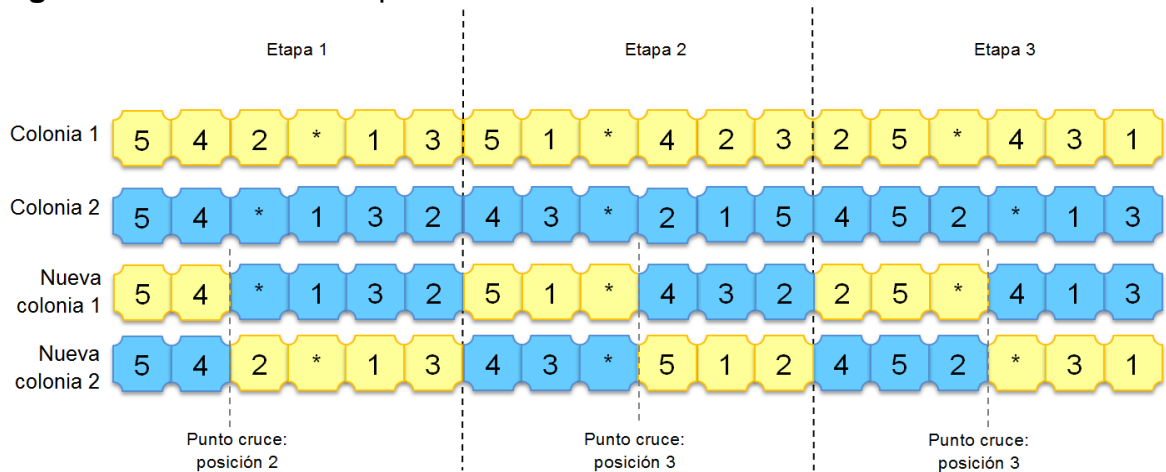
Como se observa, en la etapa 1 los trabajos 4,2 y el “*” que indica la máquina, fueron copiados en la colonia asimilada exactamente en las mismas posiciones que se encontraban en el imperialista, puesto que fueron los valores de 1 que arrojó la máscara binaria, los trabajos faltantes (5,1 y 3), fueron copiados en el orden que aparecen en la colonia. Esto se repite para cada una de las etapas.

Luego de realizar el proceso de asimilación se procede a la actualización del imperio, es decir las colonias asimiladas reemplazan a las antiguas colonias, esto con el fin de mantener el mismo número de colonias.

5.1.3. Información compartida entre colonias. Las colonias comparten información con el fin de mejorar y buscar derrocar al imperialista. El porcentaje de población que intercambia características está dado por Pcrossover (PC), en esta operación se adaptan operadores de cruce (un punto) de algoritmos genéticos.

En la figura 9, se puede observar que el punto de cruce para las etapas 1, 2 y 3 fueron 2,3 y 3 respectivamente, este punto se escogió de manera aleatoria. En la nueva colonia 1, en la etapa 1, se copió información de la colonia 1 hasta el punto de cruce, los valores faltantes se copiaron de la colonia 2, este procedimiento se hizo para cada una de las etapas.

Figura 9. Información compartida entre colonias



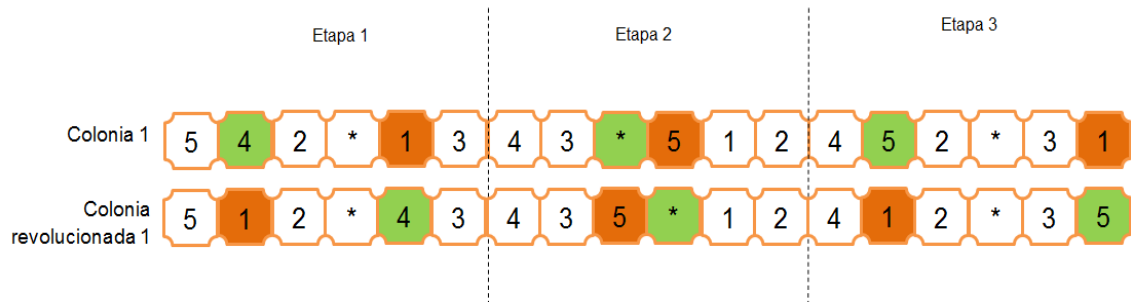
- Actualización del imperio

Luego de realizar los respectivos cruces, se procede al cálculo del costo de las colonias que compartieron información y de las nuevas colonias generadas, esto con el fin de escoger las de menor costo y dejarlas en el lugar de las colonias iniciales. En el caso del ejemplo de la figura 9, dos colonias compartieron información y dos nuevas colonias fueron generadas, se calcula el costo de las cuatro colonias y se escogen solamente las dos con menos costo, que vuelven a ingresar al imperio; las colonias restantes se eliminan con el fin de no aumentar el tamaño de la población de colonias.

5.1.4. Revolución. Varias de las colonias pretenden ser las mejores en su imperio, por esto deciden revolucionarse, es decir mejorar algunas de sus

características. Para esto se utiliza el operador de mutación de salto de un algoritmo genético. Este paso es esquematizado en la figura 10.

Figura 10. Revolución



En la etapa 1, las posiciones 2 y 5 fueron elegidas aleatoriamente para aplicar el operador de salto, es decir, los valores de 4 y 1 intercambian sus posiciones. En la etapa 2 fueron las posiciones 3 y 4, y en la etapa 3, las posiciones 2 y 6.

- Actualización del imperio.

Después de aplicar el paso de revolución, se calcula el costo de cada una de las colonias pertenecientes al imperio, si se encuentra una colonia con un costo menor que el imperialista, estos intercambian posiciones. Es decir, el imperialista pasa a ser una colonia y la colonia será el nuevo imperialista.

- Cálculo del costo total del imperio

Debido a que las colonias se han asimilado, compartido información y revolucionado, las secuencias de éstas han cambiado, en consecuencia el costo del imperio no es el mismo y debe recalcularse. La ecuación para calcular el costo fue descrita en la sección 6.1.2.

5.1.5. Guerra entre imperios. Los imperios más fuertes quitarán colonias al imperio más débil, con el fin de eliminarlo. El imperio débil es aquel que presenta el costo más elevado. En el caso de que se presenten dos imperios débiles, es decir, dos imperios con el mismo costo (el más alto), el imperio a eliminar es aquel que tenga un peor imperialista; y si persiste la igualdad de costos, la decisión se toma al azar.

El número de colonias que cada imperio quitará al más débil, depende del costo, entre menos sea el costo, más colonias se llevará. Esto se calcula mediante las siguientes ecuaciones:

$$difCost_i = T_{cid} - T_{ci} \quad (23)$$

T_{cid} , es el costo del imperio débil y T_{ci} es el costo de cada imperio.

$$\#C = \frac{difCost_i}{\sum_1^N difCost} * Col_{id} \quad (24)$$

Dónde:

$difCost$, es la diferencia entre el costo del imperio débil y otro imperio.

N es el número de imperialistas con costo menor al del imperialista débil.

Col_{id} es el número de países que tiene el imperio débil (incluyendo las colonias y el imperialista).

- Eliminación del imperio débil.

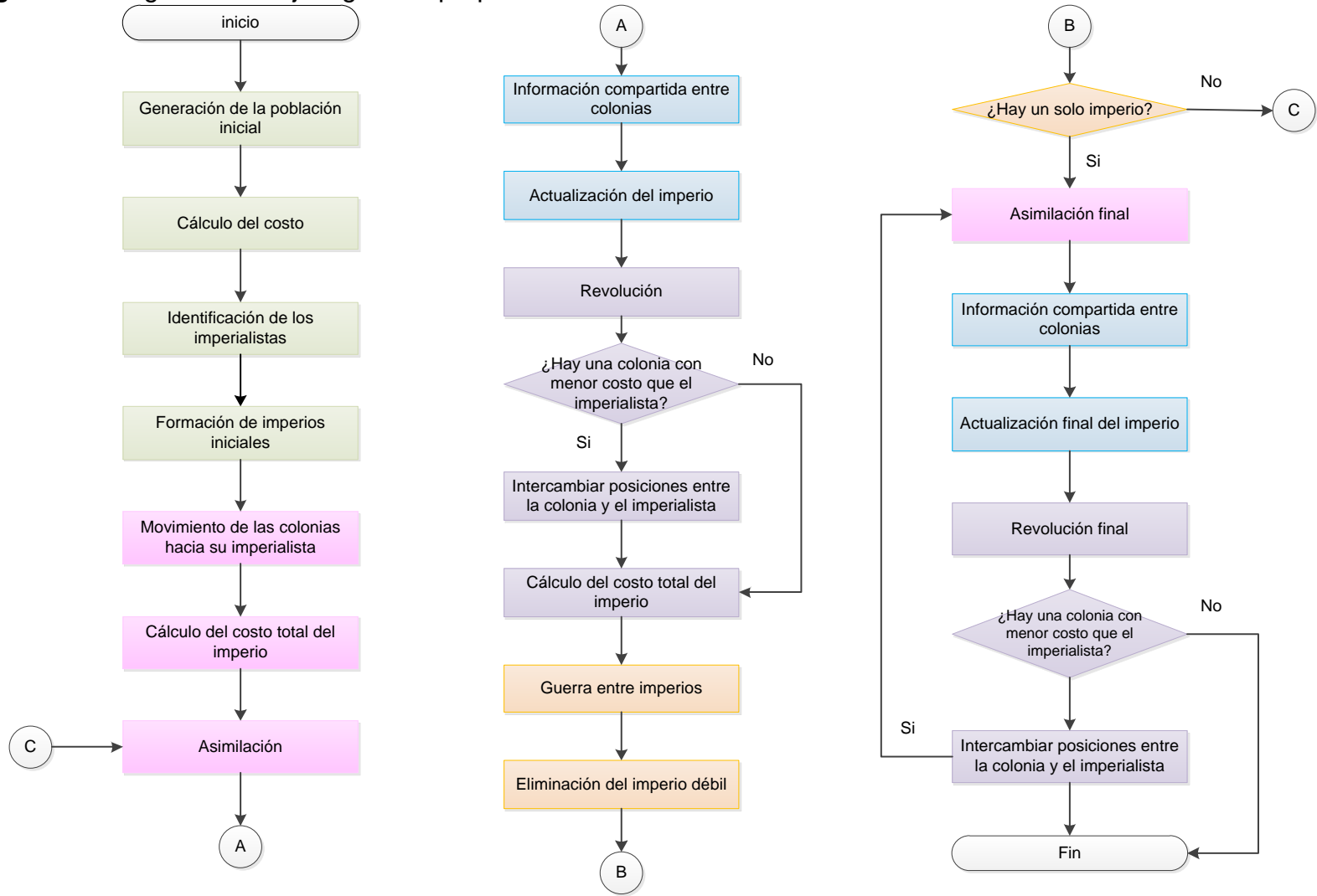
Luego de hallar el número de colonias que cada imperio se llevará, se efectúa el despojo de colonias. Cada imperio tomará al azar del imperio débil el número de colonias que le pertenecen de acuerdo a su costo.

A partir de aquí se realizan iteraciones hasta lograr tener un solo imperio que abarque todas las colonias. Cada iteración consiste en repetir desde el paso de asimilación hasta eliminación del imperio débil.

5.1.6. Condición de parada. Una vez se tiene un solo imperio que controle todas las colonias, se realizan los pasos de asimilación, información compartida entre colonias y revolución, y se guarda esa respuesta.

A continuación se realizan 100 iteraciones con el fin de buscar una mejor solución. La secuencia a seguir del algoritmo se puede ver en la figura 11.

Figura 11. Diagrama de flujo algoritmo propuesto



6. DISEÑO EXPERIMENTAL PARA LA DETERMINACIÓN DE LOS FACTORES INCIDENTES EN EL DESEMPEÑO DEL ALGORITMO.

Para el análisis de los factores influyentes en el *makespan* dado por el algoritmo, se eligió un diseño factorial fraccionado 2^{k-1} , este diseño es una fracción a la mitad del diseño factorial entero⁸⁵. En la realización de este diseño se consideraron 5 factores k y dos niveles para cada factor.

Las instancias evaluadas en el diseño se encuentran en el anexo B y los experimentos numéricos fueron desarrollados en el lenguaje de programación Matlab versión R2012a en un computador con procesador Intel core i5, 3.2GHz, 8GB de memoria RAM y un sistema operativo de 64 bytes. El diseño experimental fue analizado en Minitab 17, los datos utilizados en este análisis se encuentran en el Anexo E.

6.1. FACTORES PRINCIPALES.

El desempeño del algoritmo competitivo imperialista depende de los siguientes parámetros:

- Cantidad de países
- porcentaje de imperialistas
- Porcentaje de cruce o información compartida entre colonias
- Porcentaje de revolución de las colonias
- Porcentaje de ξ o peso de las colonias en el costo del imperio.

⁸⁵ MONTGOMERY, Douglas C. Diseño y análisis de experimentos. Limusa Wiley. 2 ed., 2004.

Los niveles considerados para cada factor son los mismos utilizados en la validación del algoritmo, y son descritos a continuación:

- Cantidad de países (CP): bajo=100 alto=250
- Porcentaje de imperialistas (%Imp): bajo=0,1 alto=0,25
- Porcentaje de cruce (%Cr): bajo=0,4 alto=0,6
- Porcentaje de revolución (%Rev): bajo=0,2 alto=0,4
- Porcentaje de ξ (% ξ): bajo=0,3 alto=0,5

En la tabla 5 se lista el diseño factorial $2^{5-1} = 2^4$, que muestra 16 de las 32 posibles combinaciones de los tratamientos del diseño factorial completo 2^5 , cada una las cuales se ejecutó 8 veces.

Tabla 5. Combinaciones de los tratamientos del diseño factorial 2^{5-1}

N°	A CP	B % Imp	C %Cr	D %Rev	E % ξ	Combinación de los tratamientos
1	-	-	-	-	+	e
2	+	-	-	-	-	a
3	-	+	-	-	-	b
4	+	+	-	-	+	abe
5	-	-	+	-	-	c
6	+	-	+	-	+	ace
7	-	+	+	-	+	bce
8	+	+	+	-	-	abc
9	-	-	-	+	-	d
10	+	-	-	+	+	ade
11	-	+	-	+	+	bde
12	+	+	-	+	-	abd
13	-	-	+	+	+	cde
14	+	-	+	+	-	acd
15	-	+	+	+	-	bcd
16	+	+	+	+	+	abcde

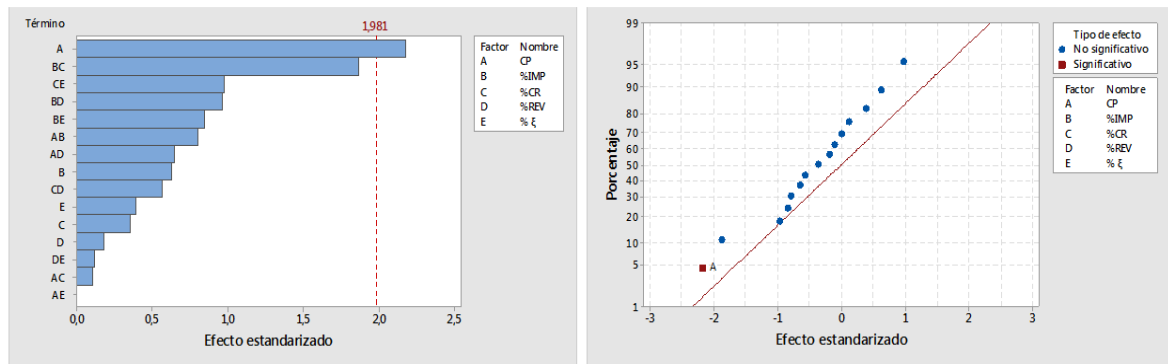
Fuente: Montgomery, Douglas

6.2. ANÁLISIS DE LOS EFECTOS PRINCIPALES DE LOS PARÁMETROS DEL ALGORITMO.

A continuación se presentan cada uno de los análisis de varianzas para cada instancia, determinando cuáles son los factores incidentes en el desempeño de la metaheurística teniendo en cuenta para ello el diagrama de Pareto y la gráfica normal de efectos estandarizados.

6.2.1. Análisis de varianza para la instancia k5m2i3

Gráfica 4. Diagrama Pareto y gráfica normal de efectos estandarizados para *makespan* de k5m2i3.



Gráfica 5. Plot error gráfica normalidad e independencia k5m2i3

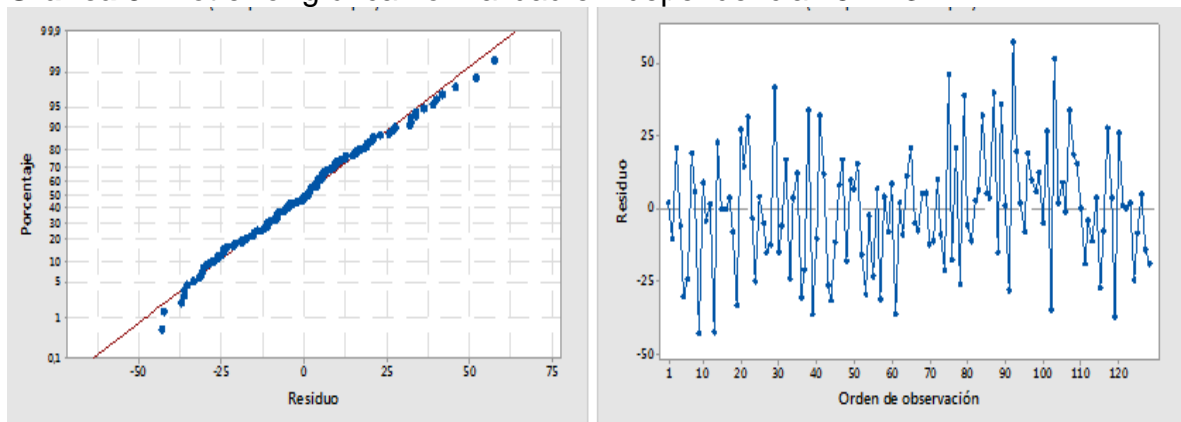


Tabla 6. ANOVA para k5m2i3.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	6246,6	416,44	0,86	0,612
Lineal	5	2635,8	527,16	1,09	0,372
CP	1	2295	2295,03	4,73	0,032
%IMP	1	190,1	190,13	0,39	0,533
%CR	1	60,5	60,5	0,12	0,725
%REV	1	15,1	15,12	0,03	0,86
% ξ	1	75	75,03	0,15	0,695
Interacciones de 2 términos	10	3610,8	361,08	0,74	0,682
CP*%IMP	1	306,3	306,28	0,63	0,429
CP*%CR	1	5,3	5,28	0,01	0,917
CP*%REV	1	205	205,03	0,42	0,517
CP*% ξ	1	0	0	0	1
%IMP*%CR	1	1682	1682	3,47	0,065
%IMP*%REV	1	450	450	0,93	0,338
%IMP*% ξ	1	344,5	344,53	0,71	0,401
%CR*%REV	1	153,1	153,12	0,32	0,575
%CR*% ξ	1	457,5	457,53	0,94	0,334
%REV*% ξ	1	7	7,03	0,01	0,904
Error	112	54344,3	485,22		
Total	127	60590,9			

A partir de la tabla 6 se concluye que únicamente el factor de cantidad de países es significativo en la variable *makespan*, esto debido a que es el único factor que presenta un valor P por debajo del 5%. Asimismo el factor que menos influye en la variable respuesta es el porcentaje de información compartida entre las colonias pues presenta un valor P de 8,6%.

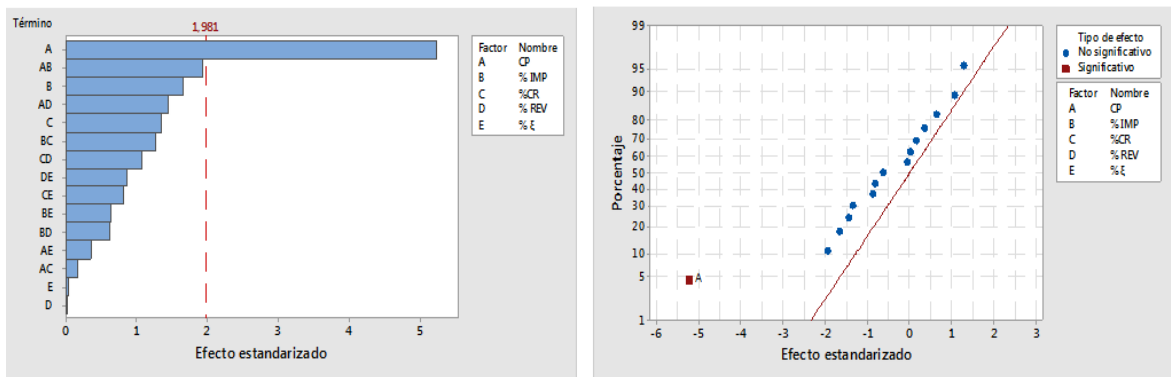
Al observar la gráfica 5 se comprueban los supuestos de normalidad e independencia, es decir los datos provienen de una población con distribución normal, y son independientes entre si, debido a que no se observa ninguna tendencia.

6.2.2. Análisis de varianza para la instancia k5m2i5

De acuerdo con las gráfica 6 y con la tabla 7, se observa que el único factor cuyo efecto es significativo en la variable *makespan* es la cantidad de países en su nivel alto, ya que su valor p es inferior a 0,05. Los factores que menos efecto tienen en la variable respuesta son el porcentaje de revolución y la influencia de las colonias dentro del poder del imperio.

Según lo observado en la gráfica 7, los datos siguen una distribución normal y son independientes entre si.

Gráfica 6. Diagrama Pareto y gráfica normal de efectos estandarizados para *makespan* de k5m2i5.



Gráfica 7. Plot error gráfica normalidad e independencia k5m2i5.

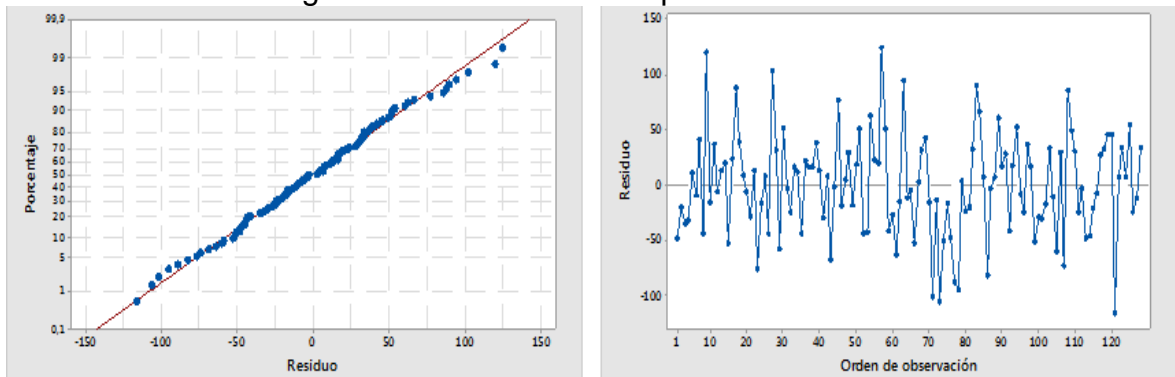
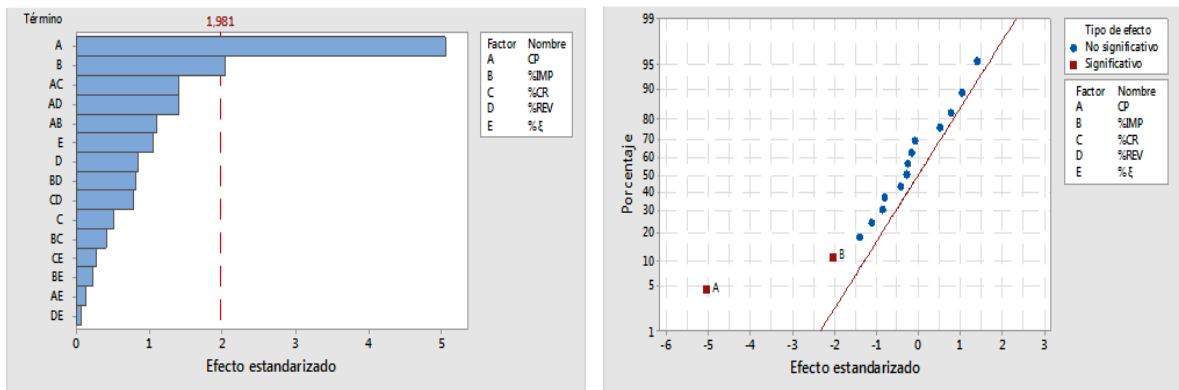


Tabla 7. ANOVA para k5m2i5.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	103290,00	6886,00	2,86	0,00
Lineal	5	76864,00	15372,80	6,38	0,00
CP	1	65885,00	65884,50	27,35	0,00
%IMP	1	6555,00	6555,10	2,72	0,10
%CR	1	4418,00	4418,00	1,83	0,18
%REV	1	2,00	2,00	0,00	0,98
% ξ	1	5,00	4,50	0,00	0,97
Interacciones de 2 términos	10	26426,00	2642,60	1,10	0,37
CP*%IMP	1	9113,00	9112,50	3,78	0,05
CP*%CR	1	66,00	66,10	0,03	0,87
CP*%REV	1	5000,00	5000,00	2,08	0,15
CP*% ξ	1	312,00	312,50	0,13	0,72
%IMP*%CR	1	3916,00	3916,10	1,63	0,21
%IMP*%REV	1	925,00	924,50	0,38	0,54
%IMP*% ξ	1	990,00	990,10	0,41	0,52
%CR*%REV	1	2738,00	2738,00	1,14	0,29
%CR*% ξ	1	1596,00	1596,10	0,66	0,42
%REV*% ξ	1	1770,00	1770,10	0,73	0,39
Error	112	269779,00	2408,70		
Total	127	373070,00			

6.2.3. Análisis de varianza para la instancia k5m3i3

Gráfica 8. Diagrama de Pareto y gráfica normal de efectos estandarizados para *makespan* de k5m3i3.



Gráfica 9. Plot error gráfica normalidad e independencia k5m3i3.

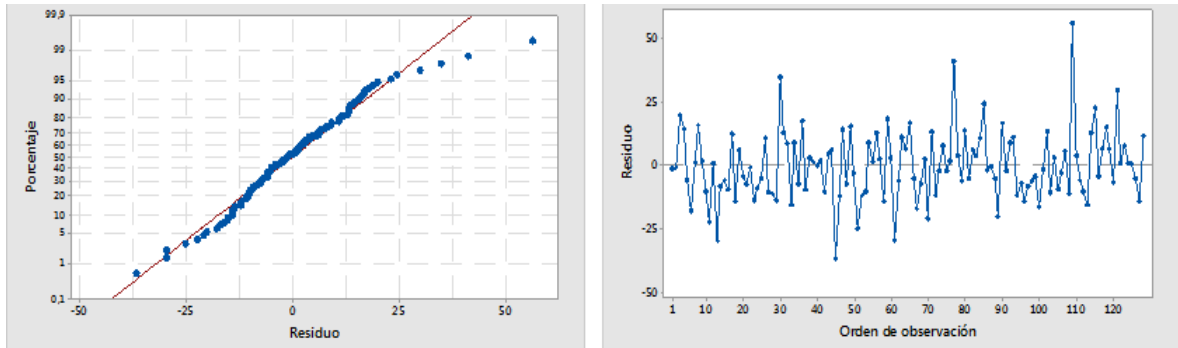


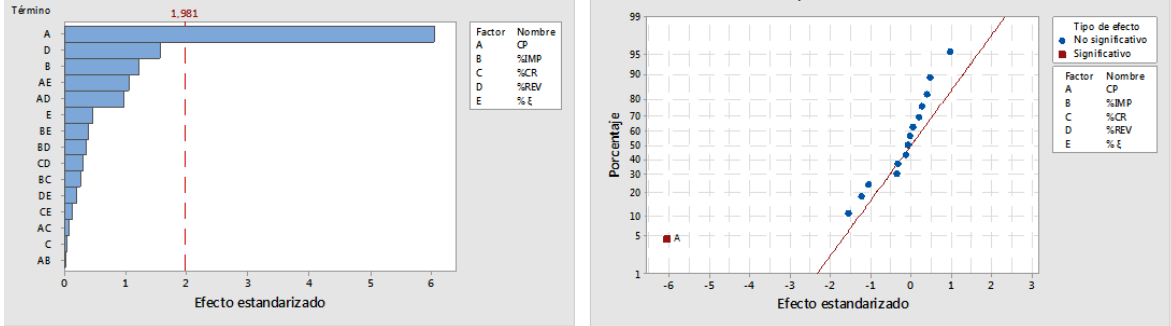
Tabla 8. ANOVA para k5m3i3.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	8179,6	545,31	2,56	0,003
Lineal	5	6753,9	1350,77	6,35	0
CP	1	5421	5421,01	25,48	0
%IMP	1	887,3	887,26	4,17	0,043
%CR	1	56,4	56,45	0,27	0,608
%REV	1	155,3	155,32	0,73	0,395
% ξ	1	233,8	233,82	1,1	0,297
Interacciones de 2 términos	10	1425,8	142,58	0,67	0,75
CP*%IMP	1	255,9	255,95	1,2	0,275
CP*%CR	1	416,9	416,88	1,96	0,164
CP*%REV	1	416,9	416,88	1,96	0,164
CP*% ξ	1	4,1	4,13	0,02	0,889
%IMP*%CR	1	35,1	35,07	0,16	0,686
%IMP*%REV	1	138,2	138,2	0,65	0,422
%IMP*% ξ	1	11,9	11,88	0,06	0,814
%CR*%REV	1	130	130,01	0,61	0,436
%CR*% ξ	1	15,8	15,82	0,07	0,786
%REV*% ξ	1	0,9	0,95	0	0,947
Error	112	23830,1	212,77		
Total	127	32009,7			

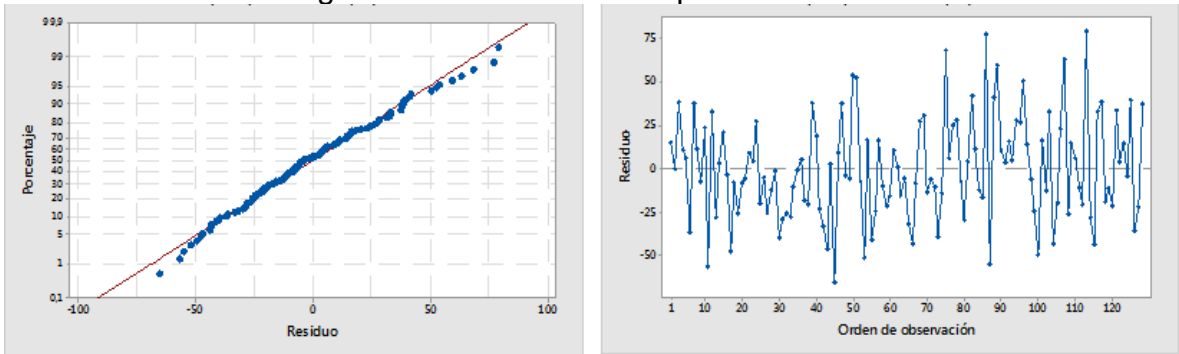
Observando los datos de la tabla 8 se puede concluir que la variable respuesta (*makespan*) se ve afectada por los efectos de la cantidad de países a generar y el porcentaje de imperialistas. Cuando estos presentan sus niveles altos. Los demás factores principales y sus interacciones no influyen en la solución debido a que tienen un valor P por encima de 0.05. Además se concluye que los datos provienen de una población con distribución normal y presentan independencia entre si.

6.2.4. Análisis de varianza para la instancia k5m3i5

Gráfica 10. Diagrama Pareto y gráfica normal de efectos estandarizados para *makespan* de k5m3i5.



Gráfica 11. Plot error gráfica normalidad e independencia k5m3i5.



Con los datos registrados en la tabla 9 se concluye que así como en la mayoría de las instancias los niveles altos de la cantidad de países (A) representan una influencia significativa en la variable respuesta, debido a que su valor P es 0. Los demás factores y sus interacciones no afectan el *makespan* y el factor que menos incide en la respuesta es el porcentaje de cruce o información compartida entre colonias (C).

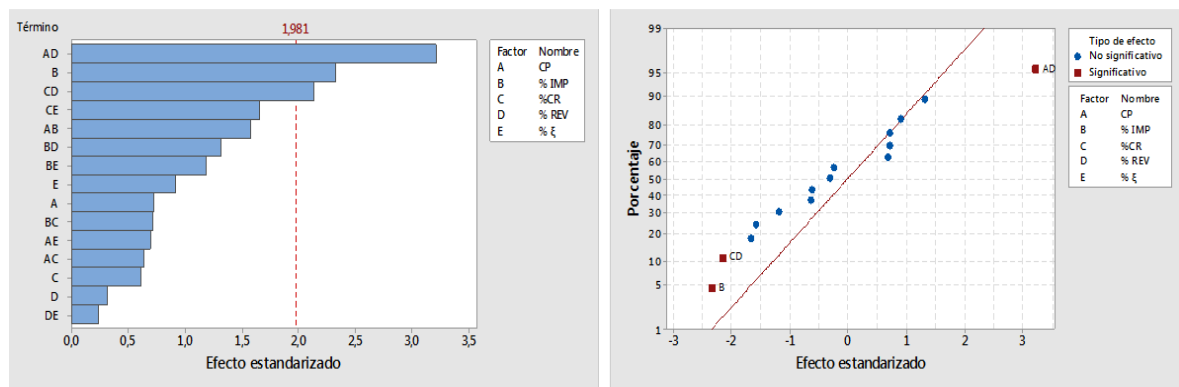
Observando la gráfica 11 se concluye que para esta instancia se cumplen los supuestos de normalidad e independencia del diseño factorial aplicado.

Tabla 9. ANOVA para k5m3i5.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	42943	2862,9	2,88	0,001
Lineal	5	40404	8080,7	8,13	0
CP	1	36281	36281,4	36,51	0
%IMP	1	1478	1478,3	1,49	0,225
%CR	1	2	2,3	0	0,962
%REV	1	2424	2423,8	2,44	0,121
% ξ	1	218	217,9	0,22	0,641
Interacciones de 2 términos	10	2540	254	0,26	0,989
CP*%IMP	1	0	0,4	0	0,984
CP*%CR	1	8	7,5	0,01	0,931
CP*%REV	1	941	940,7	0,95	0,333
CP*% ξ	1	1087	1086,9	1,09	0,298
%IMP*%CR	1	68	67,6	0,07	0,795
%IMP*%REV	1	130	130	0,13	0,718
%IMP*% ξ	1	151	150,9	0,15	0,697
%CR*%REV	1	100	99,8	0,1	0,752
%CR*% ξ	1	19	18,8	0,02	0,891
%REV*% ξ	1	37	37,2	0,04	0,847
Error	112	111307	993,8		
Total	127	154250			

6.2.5. Análisis de varianza para la instancia k7m2i3

Gráfica 12. Diagrama Pareto y gráfica normal de efectos estandarizados para *makespan* de k7m2i3.



Gráfica 13. Plot error gráfica normalidad e independencia k7m2i3.

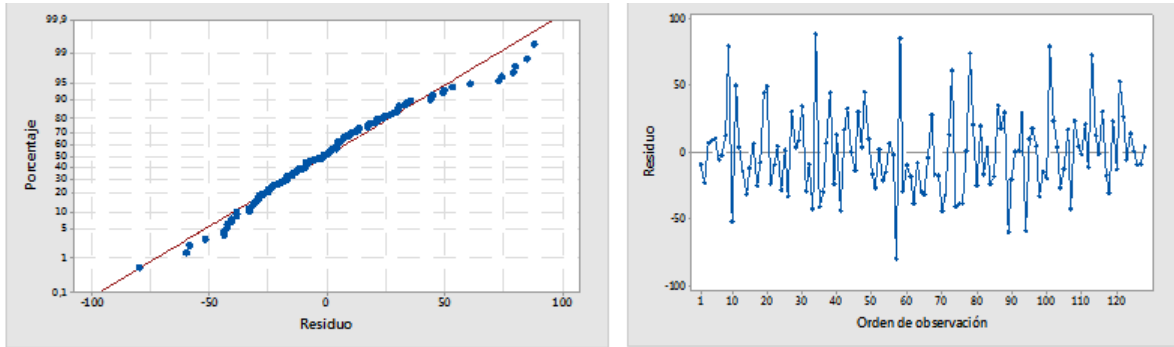


Tabla 10. ANOVA para k7m2i3.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	39093,00	2606,20	2,13	0,01
Lineal	5	8866,00	1773,10	1,45	0,21
CP	1	648,00	648,00	0,53	0,47
%IMP	1	6612,00	6612,50	5,41	0,02
%CR	1	465,00	465,10	0,38	0,54
%REV	1	116,00	116,30	0,10	0,76
% ξ	1	1024,00	1023,80	0,84	0,36
Interacciones de 2 términos	10	30227,00	3022,70	2,47	0,01
CP*%IMP	1	3042,00	3042,00	2,49	0,12
CP*%CR	1	496,00	496,10	0,41	0,53
CP*%REV	1	12601,00	12600,80	10,30	0,00
CP*% ξ	1	587,00	586,50	0,48	0,49
%IMP*%CR	1	630,00	630,10	0,52	0,47
%IMP*%REV	1	2129,00	2128,80	1,74	0,19
%IMP*% ξ	1	1697,00	1696,50	1,39	0,24
%CR*%REV	1	5592,00	5591,50	4,57	0,04
%CR*% ξ	1	3383,00	3382,50	2,77	0,10
%REV*% ξ	1	72,00	72,00	0,06	0,81
Error	112	137010,00	1223,30		
Total	127	176103,00			

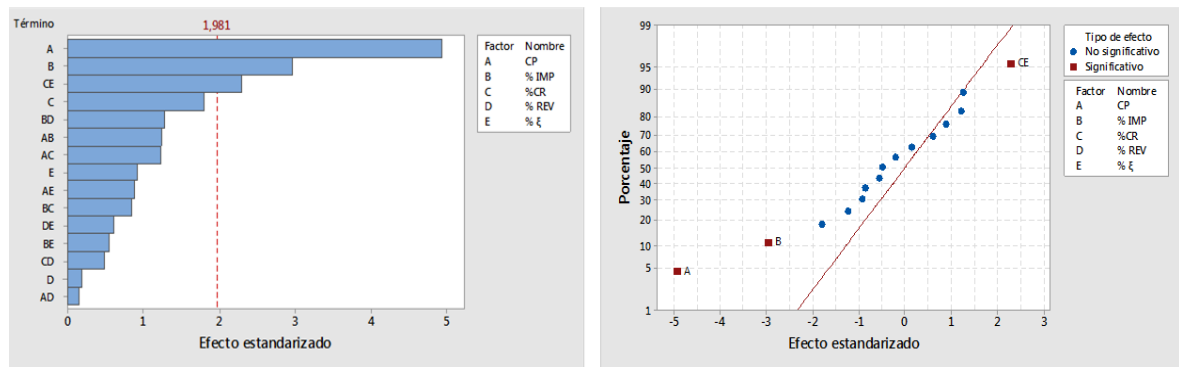
Con base en la información contenida en la tabla 10, es posible concluir que los factores que tienen efectos significativos sobre la variable *makespan* son el porcentaje de países imperialistas, y los efectos confundidos AD+BCE (CP y %Rev) y CD+ABE (%Cr y %Rev), dado que su valor P es inferior a 0,05. Con relación al efecto confundido AD+BCE, se espera que el principal efecto corresponda a las interacciones del porcentaje de imperialistas, influencia de las

colonias en el poder del imperio y el porcentaje de cruce dado que el factor D (Porcentaje de revolución) es el menos significativo. Para el efecto combinado CD+ABE, el efecto principal corresponderá a la combinación de los factores cantidad de países, porcentaje de imperialistas e influencia de las colonias en el poder del imperio.

Según se observa en la gráfica 13, los datos de esta instancia utilizados en el diseño experimental provienen de una población con distribución normal, y son independientes los unos de los otros.

6.2.6. Análisis de varianza para la instancia k7m2i5

Gráfica 14. Diagrama Pareto y gráfica normal de efectos estandarizados para *makespan* de k7m2i5.



Gráfica 15. Plot error gráfica normalidad e independencia k7m2i5.

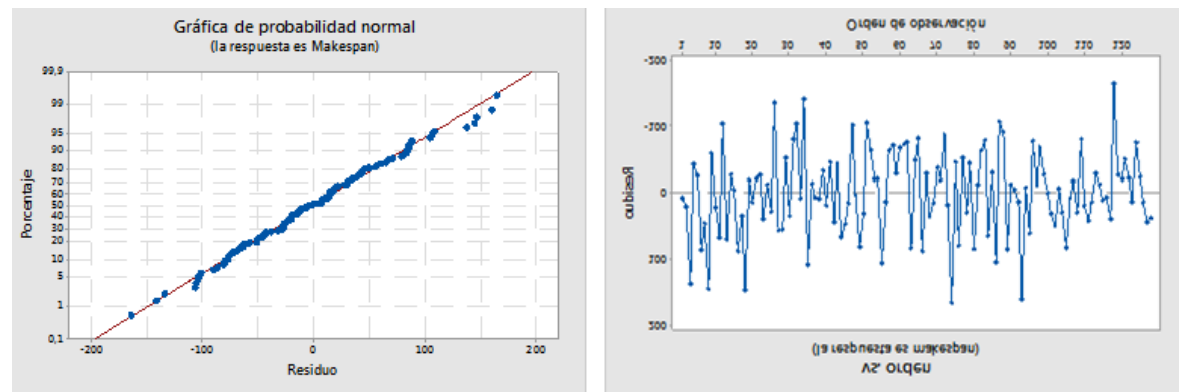


Tabla 11. ANOVA para k7m2i5.

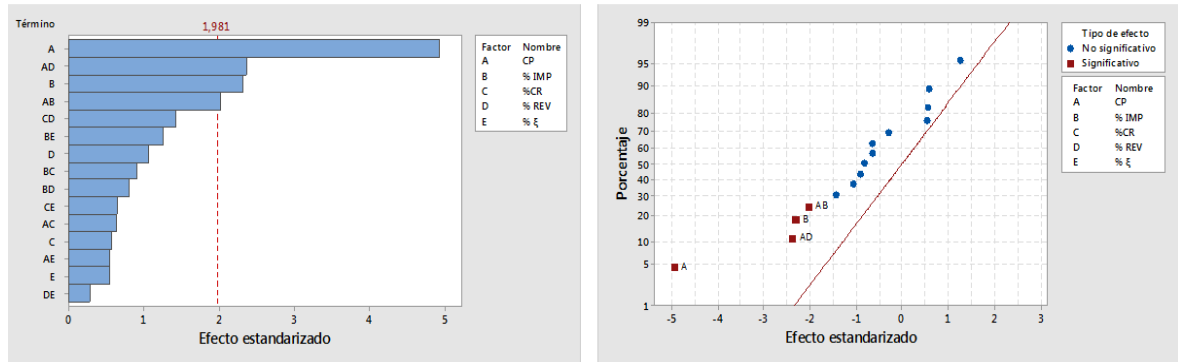
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	228118,00	15208,00	3,30	0,00
Lineal	5	171179,00	34236,00	7,43	0,00
CP	1	111865,00	111865,00	24,28	0,00
%IMP	1	40328,00	40328,00	8,75	0,00
%CR	1	14921,00	14921,00	3,24	0,08
%REV	1	171,00	171,00	0,04	0,85
% ξ	1	3894,00	3894,00	0,85	0,36
Interacciones de 2 términos	10	56939,00	5694,00	1,24	0,28
CP*%IMP	1	7080,00	7080,00	1,54	0,22
CP*%CR	1	6933,00	6933,00	1,50	0,22
CP*%REV	1	98,00	98,00	0,02	0,88
CP*% ξ	1	3591,00	3591,00	0,78	0,38
%IMP*%CR	1	3301,00	3301,00	0,72	0,40
%IMP*%REV	1	7442,00	7442,00	1,62	0,21
%IMP*% ξ	1	1418,00	1418,00	0,31	0,58
%CR*%REV	1	1070,00	1070,00	0,23	0,63
%CR*% ξ	1	24310,00	24310,00	5,28	0,02
%REV*% ξ	1	1697,00	1697,00	0,37	0,55
Error	112	516062,00	4608,00		
Total	127	744180,00			

Para la instancia k7m2i5, es posible observar que los factores que ejercen un efecto significativo sobre la variable *makespan* corresponden a la cantidad de países (A), el porcentaje de imperialistas (B) y el efecto confundido CE+ABD (Cr*% ξ), puesto que el factor D (porcentaje de revolución) es el que menos incide sobre la variable respuesta, el efecto ABD es nulo, por lo tanto la combinación que influye es CE. El *makespan* se minimiza cuando los efectos significativos se encuentran en su nivel alto, por su parte la combinación de las interacciones que minimiza el *makespan* es los niveles altos de C (Porcentaje de cruce) y E (% ξ).

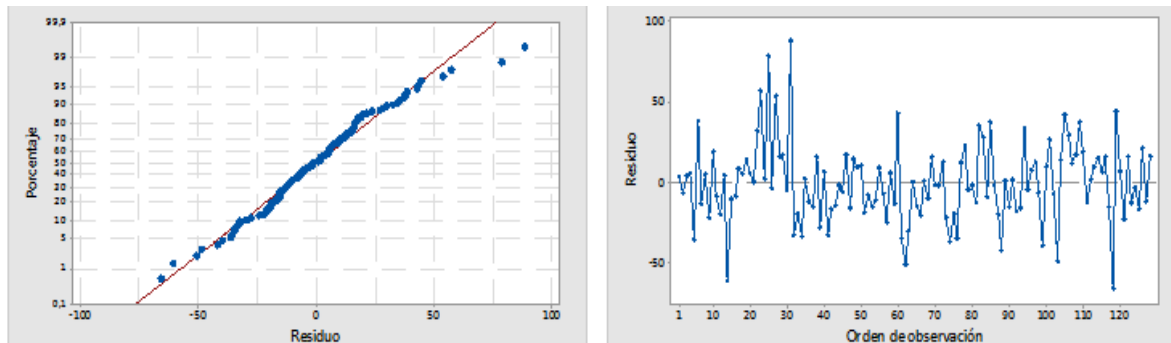
Se comprueba además, que los datos son independientes entre si, y provienen de una población que presenta distribución normal.

6.2.7. Análisis de varianza para la instancia k7m3i3

Gráfica 16. Diagrama Pareto y gráfica normal de efectos estandarizados para *makespan* de k7m3i3.



Gráfica 17. Plot error gráfica normalidad e independencia k7m3i3



Teniendo en cuenta la información descrita en la tabla 12, los efectos que son significativos en la variable *makespan* son la cantidad de países (A), el porcentaje de imperialistas (B) y los efectos confundidos AD+BCE y AB+CDE dado que su valor P es inferior a 0,05.

Al analizar el efecto confundido AD+BCE, es posible concluir que el efecto significativo sea producto de la combinación AD, ya que los efectos E y C son los menos significativos. Por otra parte al analizar el efecto confundido AB+CDE, el

efecto significativo es producto de la combinación AB, teniendo en cuenta la significancia de los factores E y C.

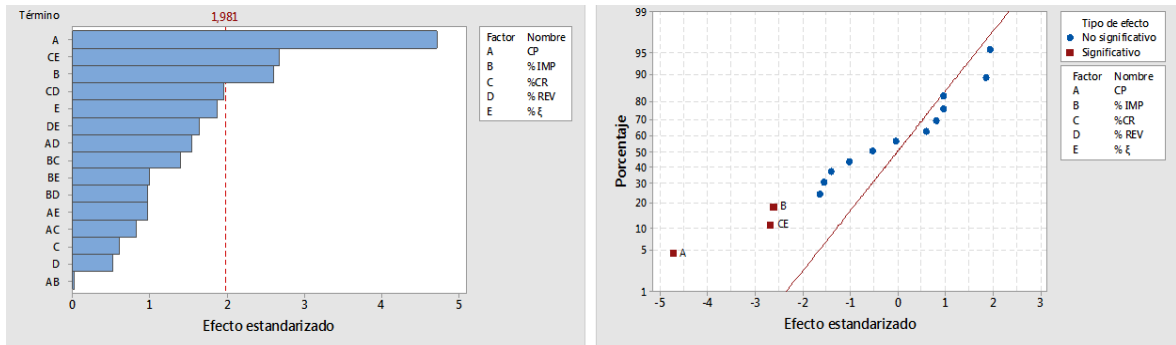
Según los supuestos del diseño representados en la gráfica 17, los datos provienen de una población con distribución normal y son independientes entre si.

Tabla 12. ANOVA para k7m3i3.

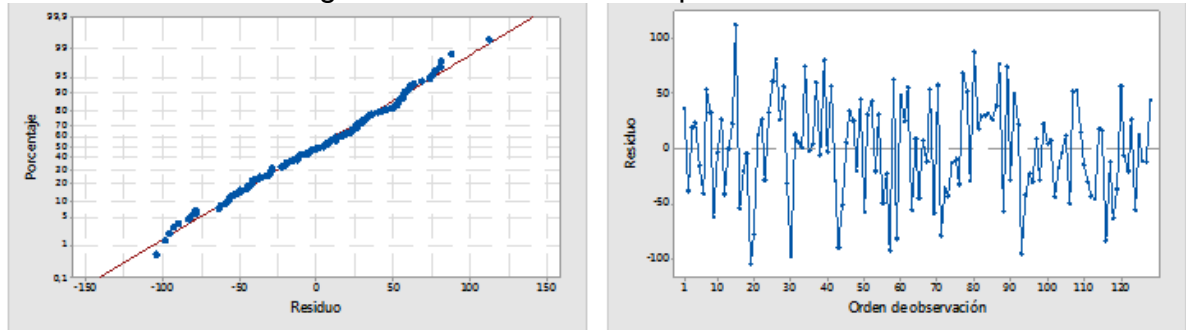
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	32916,00	2194,40	3,16	0,00
Lineal	5	21811,00	4362,30	6,29	0,00
CP	1	16882,00	16882,00	24,33	0,00
%IMP	1	3698,00	3698,00	5,33	0,02
%CR	1	231,00	231,10	0,33	0,57
%REV	1	790,00	790,00	1,14	0,29
% ξ	1	210,00	210,10	0,30	0,58
Interacciones de 2 términos	10	11104,00	1110,40	1,60	0,12
CP*%IMP	1	2831,00	2831,30	4,08	0,05
CP*%CR	1	282,00	282,00	0,41	0,53
CP*%REV	1	3872,00	3872,00	5,58	0,02
CP*% ξ	1	215,00	215,30	0,31	0,58
%IMP*%CR	1	578,00	578,00	0,83	0,36
%IMP*%REV	1	458,00	457,50	0,66	0,42
%IMP*% ξ	1	1105,00	1104,50	1,59	0,21
%CR*%REV	1	1418,00	1417,80	2,04	0,16
%CR*% ξ	1	288,00	288,00	0,42	0,52
%REV*% ξ	1	58,00	57,80	0,08	0,77
Error	112	77710,00	693,80		
Total	127	110625,00			

6.2.8. Análisis de varianza para la instancia k7m3i5

Gráfica 18. Diagrama Pareto y gráfica normal de efectos estandarizados para *makespan* de k7m3i5.



Gráfica 19. Plot error gráfica normalidad e independencia k7m3i5.



De la tabla 13, se puede concluir que los efectos principales que influyen en el *Makespan* son la cantidad de países (A) y el porcentaje de imperialistas (B) junto con el efecto confundido CE+ABD, ya que su valor P es inferior a 0,05.

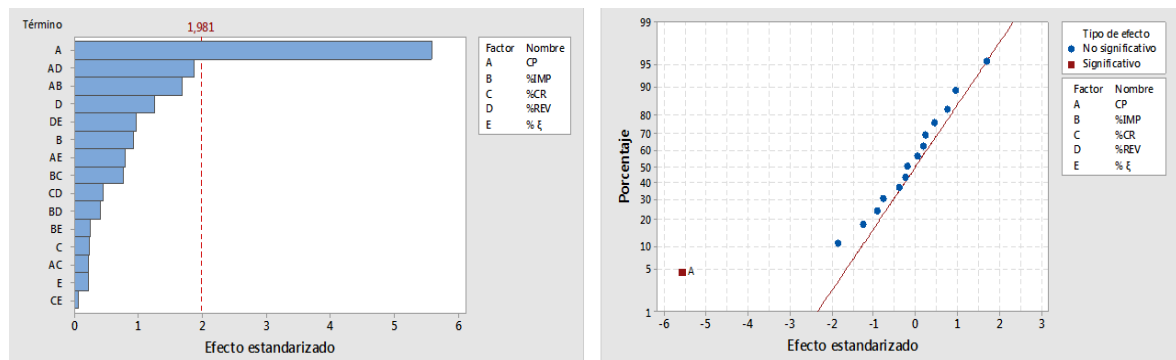
En la gráfica 19, se encuentran representados dos de los supuestos del diseño experimental, los cuales indican que los datos provienen de una población con distribución normal y son independientes.

Tabla 13. ANOVA para k7m3i5.

Fuente	Grados de libertad	Suma de cuadrados	Cuadrado medio	Valor F	Valor P
Modelo	15	129921,00	8661,40	3,65	0,00
Lineal	5	78817,00	15763,30	6,64	0,00
CP	1	52813,00	52812,50	22,23	0,00
%IMP	1	16155,00	16155,00	6,80	0,01
%CR	1	872,00	871,50	0,37	0,55
%REV	1	657,00	657,00	0,28	0,60
% ξ	1	8320,00	8320,50	3,50	0,06
Interacciones de 2 términos	10	51105,00	5110,50	2,15	0,03
CP*%IMP	1	1,00	1,10	0,00	0,98
CP*%CR	1	1596,00	1596,10	0,67	0,41
CP*%REV	1	5671,00	5671,10	2,39	0,13
CP*% ξ	1	2228,00	2227,80	0,94	0,34
%IMP*%CR	1	4632,00	4632,00	1,95	0,17
%IMP*%REV	1	2261,00	2261,30	0,95	0,33
%IMP*% ξ	1	2381,00	2380,50	1,00	0,32
%CR*%REV	1	9079,00	9078,80	3,82	0,05
%CR*% ξ	1	16928,00	16928,00	7,13	0,01
%REV*% ξ	1	6328,00	6328,10	2,66	0,11
Error	112	266068,00	2375,60		
Total	127	395989,00			

6.2.9. Análisis de varianza para la instancia k9m2i3

Gráfica 20. Diagrama Pareto y gráfica normal de efectos estandarizados para *makespan* de k9m2i3.



Gráfica 21. Plot error gráfica normalidad e independencia k9m2i3.

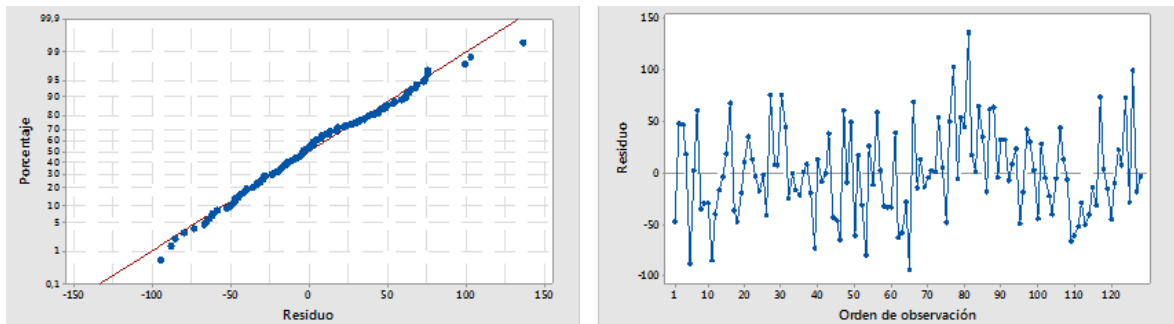


Tabla 14. ANOVA para k9m2i3.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	89391	5959,4	2,82	0,001
Lineal	5	70781	14156,2	6,7	0
CP	1	65567	65567,3	31,04	0
%IMP	1	1733	1733,1	0,82	0,367
%CR	1	107	106,9	0,05	0,822
%REV	1	3291	3290,6	1,56	0,215
% ξ	1	83	82,9	0,04	0,843
Interacciones de 2 términos	10	18610	1861	0,88	0,553
CP*%IMP	1	6009	6008,8	2,84	0,094
CP*%CR	1	83	82,9	0,04	0,843
CP*%REV	1	7275	7275,2	3,44	0,066
CP*% ξ	1	1281	1281,4	0,61	0,438
%IMP*%CR	1	1194	1194,4	0,57	0,454
%IMP*%REV	1	322	321,9	0,15	0,697
%IMP*% ξ	1	118	118,2	0,06	0,813
%CR*%REV	1	410	409,7	0,19	0,661
%CR*% ξ	1	3	3,4	0	0,968
%REV*% ξ	1	1914	1914,3	0,91	0,343
Error	112	236598	2112,5		
Total	127	325989			

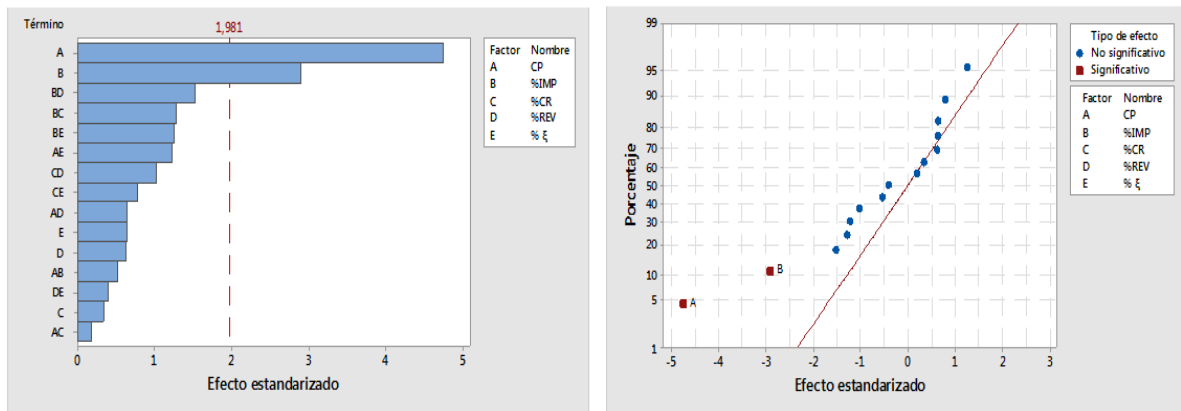
En la tabla 14 se observa que únicamente el efecto generado por la cantidad de países (A) es significativo para la variable *makespan* al tener un valor P menor al

5%. El factor que menos incide en la respuesta es el porcentaje de cruce o información compartida entre colonias (C).

Según se observa en la gráfica 21, los datos utilizados en la realización del diseño experimental provienen de una población con distribución normal y son independientes, debido a que no se observa alguna tendencia.

6.2.10. Análisis de varianza para la instancia k9m2i5

Gráfica 22. Diagrama Pareto y gráfica normal de efectos estandarizados para *makespan* de k9m2i5.



Gráfica 23. Plot error gráfica normalidad e independencia k9m2i5.

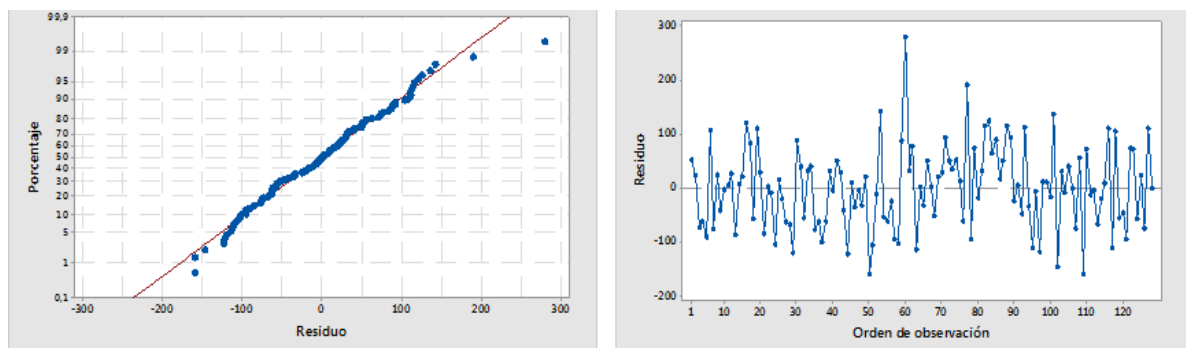


Tabla 15. ANOVA para k9m2i5.

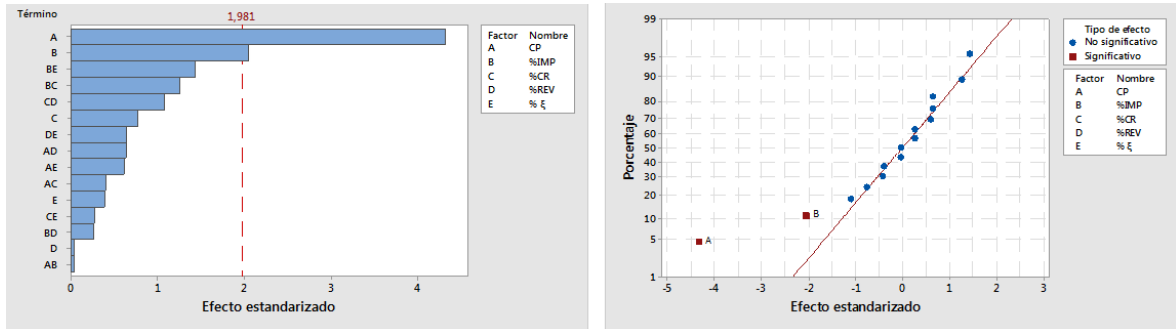
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	273225	18215	2,76	0,001
Lineal	5	210061	42012	6,37	0
CP	1	148581	148581	22,54	0
%IMP	1	55403	55403	8,41	0,005
%CR	1	765	765	0,12	0,734
%REV	1	2583	2583	0,39	0,533
% ξ	1	2729	2729	0,41	0,521
Interacciones de 2 términos	10	63164	6316	0,96	0,484
CP*%IMP	1	1853	1853	0,28	0,597
CP*%CR	1	223	223	0,03	0,854
CP*%REV	1	2766	2766	0,42	0,518
CP*% ξ	1	9888	9888	1,5	0,223
%IMP*%CR	1	10897	10897	1,65	0,201
%IMP*%REV	1	15291	15291	2,32	0,131
%IMP*% ξ	1	10314	10314	1,56	0,214
%CR*%REV	1	6830	6830	1,04	0,311
%CR*% ξ	1	4016	4016	0,61	0,437
%REV*% ξ	1	1087	1087	0,16	0,685
Error	112	738205	6591		
Total	127	1011430			

Para esta instancia existen dos factores cuyos niveles altos afectan el resultado del *makespan*, dichos factores son la cantidad de países (A) junto con el % de imperialistas (B). Ninguna de las interacciones de los demás factores principales incide en la respuesta y el factor menos relevante, es decir, independientemente de los niveles que utilice afecta la respuesta es el porcentaje de cruce (C), con un 7,34% en el valor P.

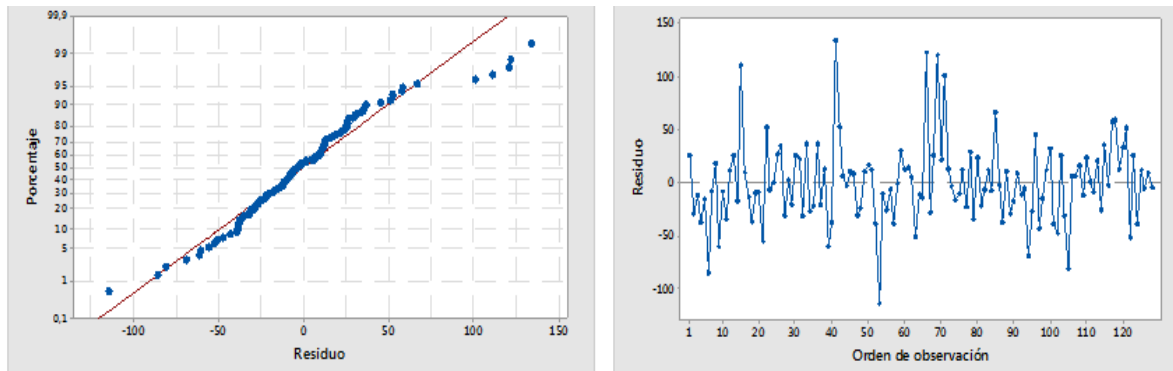
Se comprueba además, que los datos son independientes entre si, y provienen de una población que presenta distribución normal.

6.2.11. Análisis de varianza para la instancia k9m3i3

Gráfica 24. Diagrama de Pareto de efectos estandarizados para *makespan* de k9m3i3.



Gráfica 25. Plot error gráfica normalidad e independencia k9m3i3.



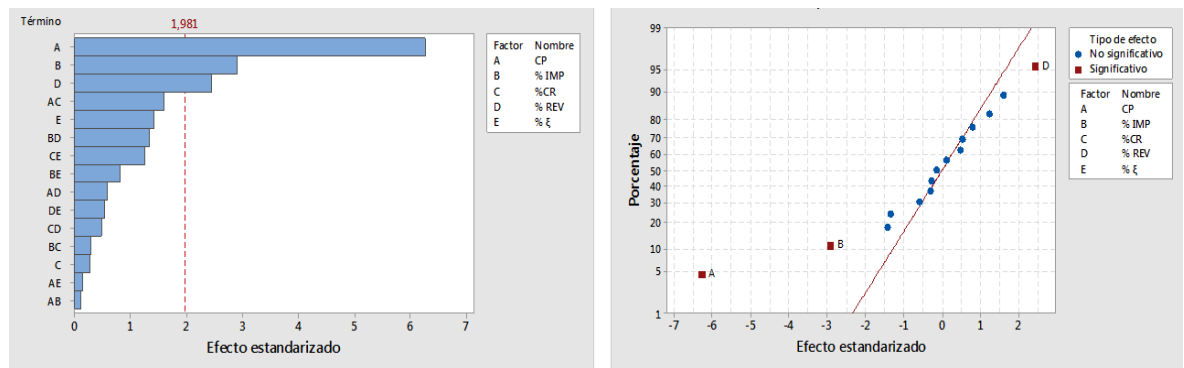
Para esta instancia, así como para k9m2i5 los factores incidentes en la respuesta son la cantidad de países (A) y el porcentaje de imperialistas (C), cuyos valores P son 0, y 0,042 respectivamente. Las mejores soluciones del *makespan* se presentan cuando ambos factores están en sus niveles altos. El factor menos incidente es el% de revolución con un 9,73%. También se observa que los datos son normales e independientes entre si.

Tabla 16. ANOVA para k9m3i3.

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	51296	3419,7	2	0,021
Lineal	5	40480	8096	4,73	0,001
CP	1	32005	32004,5	18,69	0
%IMP	1	7230	7230	4,22	0,042
%CR	1	990	990,1	0,58	0,449
%REV	1	2	2	0	0,973
% ξ	1	253	253,1	0,15	0,701
Interacciones de 2 términos	10	10816	1081,6	0,63	0,784
CP*%IMP	1	2	2	0	0,973
CP*%CR	1	282	282	0,16	0,686
CP*%REV	1	694	693,8	0,41	0,526
CP*% ξ	1	639	639	0,37	0,543
%IMP*%CR	1	2701	2701,1	1,58	0,212
%IMP*%REV	1	120	120,1	0,07	0,792
%IMP*% ξ	1	3528	3528	2,06	0,154
%CR*%REV	1	2032	2032	1,19	0,278
%CR*% ξ	1	124	124	0,07	0,788
%REV*% ξ	1	694	693,8	0,41	0,526
Error	112	191803	1712,5		
Total	127	243099			

6.2.12. Análisis de varianza para la instancia k9m3i5

Gráfica 26. Diagrama Pareto y gráfica normal de efectos estandarizados para *makespan* de k9m3i5.



Gráfica 27. Plot error gráfica normalidad e independencia k9m3i5.

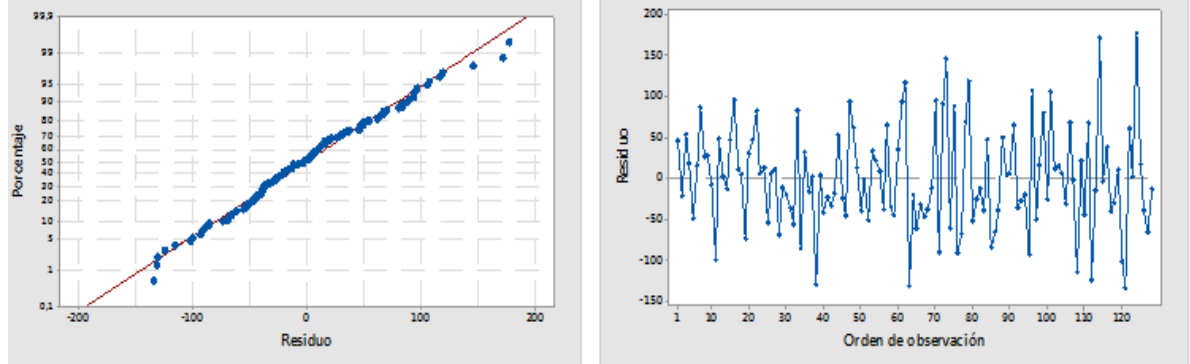


Tabla 17. ANOVA para k9m3i5

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	15	279550,00	18637,00	4,21	0,00
Lineal	5	246486,00	49297,00	11,14	0,00
CP	1	173976,00	173976,00	39,31	0,00
%IMP	1	37094,00	37094,00	8,38	0,01
%CR	1	348,00	348,00	0,08	0,78
%REV	1	26306,00	26306,00	5,94	0,02
% ξ	1	8762,00	8762,00	1,98	0,16
Interacciones de 2 términos	10	33064,00	3306,00	0,75	0,68
CP*%IMP	1	54,00	54,00	0,01	0,91
CP*%CR	1	11231,00	11231,00	2,54	0,11
CP*%REV	1	1478,00	1478,00	0,33	0,56
CP*% ξ	1	86,00	86,00	0,02	0,89
%IMP*%CR	1	368,00	368,00	0,08	0,77
%IMP*%REV	1	7922,00	7922,00	1,79	0,18
%IMP*% ξ	1	2879,00	2879,00	0,65	0,42
%CR*%REV	1	1018,00	1018,00	0,23	0,63
%CR*% ξ	1	6772,00	6772,00	1,53	0,22
%REV*% ξ	1	1256,00	1256,00	0,28	0,60
Error	112	495737,00	4426,00		
Total	127	775287,00			

Para la instancia k9m3i5, es posible concluir que los efectos principales que inciden sobre el *makespan* son la cantidad de países (A), el porcentaje de

imperialistas (B) y el porcentaje de revolución (D), los efectos confundidos no inciden de manera significativa sobre la variable respuesta. Así pues, el valor del *makespan* se minimizará al combinar los niveles altos de los factores A, B y D.

Se observa en la gráfica 27, que los datos utilizados en el diseño experimental provienen de una población con distribución normal y son independientes entre si.

7. VALIDACIÓN DEL ALGORITMO

El desempeño del algoritmo propuesto cuyo código puede detallarse en el anexo D, fue evaluado mediante 12 instancias descritas en el anexo B con el propósito de comparar la eficiencia del algoritmo frente a los resultados obtenidos con el método exacto.

Los parámetros que se tuvieron en cuenta para el proceso de validación son: la cantidad de países, el porcentaje de países imperialistas, el porcentaje de países que comparten información o porcentaje de cruce, la influencia o peso de las colonias en el poder total del imperio ($\% \xi$) y el porcentaje de revolución. El algoritmo fue probado en el lenguaje de programación Matlab versión R2012a.

Para cada una de las instancias el algoritmo se ejecutó 10 veces, probando en cada una de las corridas las 32 posibles combinaciones, esto puede detallarse en el anexo F. El promedio de las soluciones obtenidas para cada una de las combinaciones son comparadas con la solución obtenida mediante el método exacto.

En la tabla 18 se presentan los factores que influyen en el desempeño del algoritmo y sus respectivos niveles, cuyos valores corresponden a los utilizados por Rabiee et al⁸⁶, quien desarrolló un algoritmo híbrido basado en el algoritmo competitivo imperialista aplicado a la solución del problema de *flow shop* híbrido flexible con dos etapas de procesamiento y máquinas paralelas no relacionadas en cada etapa.

⁸⁶ RABIEE, M. et al. Op cit, p.a229-1245

Tabla 18. Factores y niveles

Factor	Nivel bajo	Nivel alto
Cantidad de países	100	250
% Imperialistas	0.10	0.25
% Cruce	0.40	0.60
% Revolución	0.20	0.40
% ξ	0.30	0.50

7.1. ANÁLISIS DE RESULTADOS.

A partir de la tabla 19 hasta la tabla 30 se presentan los resultados para cada una de las instancias evaluadas, para ello se tienen en cuenta los siguientes parámetros:

CP: cantidad de países.

% ξ : influencia de las colonias en el poder total del imperio.

%Cr: porcentaje de cruce.

%Imp: porcentaje de imperialistas.

%Rev: porcentaje de revolución.

SC: solución conocida (método exacto).

Med: media de las diez réplicas establecidas para cada combinación.

SD: desviación estándar.

%dif: porcentaje de diferencia entre SC y Med.

MO: mejor solución obtenida por el algoritmo para cada una de las combinaciones.

Comparando los datos de la tabla 19, los cuales fueron proporcionados por el algoritmo y el método exacto se observa que el método exacto siempre encuentra la mejor solución. El ICA, independientemente de la combinación que se utilice no llega a estos resultados, siendo 678 el valor encontrado más cercano al *makespan*, cuya combinación arroja el mejor resultado promedio distante en un 5,91% de la solución conocida.

Tabla 19. Resultados instancia k5m2i3

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1	100	0,1	0,4	0,2	0,3	659	699	721,5	0,0866	21,4
2	250	0,1	0,4	0,2	0,3	659	688	712,8	0,0755	17,19
3	100	0,25	0,4	0,2	0,3	659	688	736,1	0,1047	41,76
4	250	250	0,4	0,2	0,3	659	688	707,2	0,0682	17,64
5	100	0,1	0,6	0,2	0,3	659	678	700,4	0,0591	17,63
6	250	0,1	0,6	0,2	0,3	659	687	715,7	0,0792	20,09
7	100	0,25	0,6	0,2	0,3	659	688	719,9	0,0846	24,03
8	250	0,25	0,6	0,2	0,3	659	678	707,3	0,0683	19,9
9	100	0,1	0,4	0,4	0,3	659	678	723	0,0885	25,66
10	250	0,1	0,4	0,4	0,3	659	687	720	0,0847	42,73
11	100	0,25	0,4	0,4	0,3	659	690	724,8	0,0908	17,92
12	250	0,25	0,4	0,4	0,3	659	688	708,4	0,0697	15,92
13	100	0,1	0,6	0,4	0,3	659	687	719,5	0,0841	29,82
14	250	0,1	0,6	0,4	0,3	659	685	710,8	0,0729	19,63
15	100	0,25	0,6	0,4	0,3	659	688	721,3	0,0864	32,32
16	250	0,25	0,6	0,4	0,3	659	688	711,4	0,0737	18,98
17	100	0,1	0,4	0,2	0,5	659	690	728,8	0,0958	22,08
18	250	0,1	0,4	0,2	0,5	659	688	713,2	0,076	19,9
19	100	0,25	0,4	0,2	0,5	659	703	730,1	0,0974	22,56
20	250	0,25	0,4	0,2	0,5	659	688	710,3	0,0722	16,74
21	100	0,1	0,6	0,2	0,5	659	690	721,1	0,0861	15,47
22	250	0,1	0,6	0,2	0,5	659	699	717,4	0,0814	21,55
23	100	0,25	0,6	0,2	0,5	659	688	710,2	0,0721	18,78
24	250	0,25	0,6	0,2	0,5	659	688	708,8	0,0703	14,5
25	100	0,1	0,4	0,4	0,5	659	689	715	0,0783	17,4
26	250	0,1	0,4	0,4	0,5	659	689	719,4	0,084	18,54
27	100	0,25	0,4	0,4	0,5	659	688	712,8	0,0755	17,83
28	250	0,25	0,4	0,4	0,5	659	678	709,1	0,0707	19,78
29	100	0,1	0,6	0,4	0,5	659	702	732,1	0,0998	18,64
30	250	0,1	0,6	0,4	0,5	659	688	722	0,0873	26,17
31	100	0,25	0,6	0,4	0,5	659	688	707,8	0,0689	16,82
32	250	0,25	0,6	0,4	0,5	659	688	711,2	0,0734	21,9

Tabla 20. Resultados instancia k5m2i5.

CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
100	0,1	0,4	0,2	0,3	796	1030	1087,7	0,28	36,85
250	0,1	0,4	0,2	0,3	796	958	1024	0,24	50,64
100	0,25	0,4	0,2	0,3	796	1006	1084,6	0,29	44,41
250	0,25	0,4	0,2	0,3	796	974	1039	0,25	41,67
100	0,1	0,6	0,2	0,3	796	1019	1099,2	0,30	48,33
250	0,1	0,6	0,2	0,3	796	976	1038,9	0,25	50,34
100	0,25	0,6	0,2	0,3	796	990	1071,7	0,28	49,28
250	0,25	0,6	0,2	0,3	796	990	1031,9	0,24	28,75
100	0,1	0,4	0,4	0,3	796	1024	1089,5	0,29	57,48
250	0,1	0,4	0,4	0,3	796	961	1047,9	0,26	61,66
100	0,25	0,4	0,4	0,3	796	961	1083,7	0,30	80,38
250	0,25	0,4	0,4	0,3	796	948	1023,2	0,24	46,31
100	0,1	0,6	0,4	0,3	796	970	1105,2	0,32	67,58
250	0,1	0,6	0,4	0,3	796	949	1055,1	0,27	59,95
100	0,25	0,6	0,4	0,3	796	1026	1083,2	0,28	38,48
250	0,25	0,6	0,4	0,3	796	969	1006,1	0,22	37,31
100	0,1	0,4	0,2	0,5	796	1021	1093,9	0,29	37,13
250	0,1	0,4	0,2	0,5	796	984	1057,6	0,27	38,66
100	0,25	0,4	0,2	0,5	796	958	1051,5	0,27	44,87
250	0,25	0,4	0,2	0,5	796	960	1000,7	0,21	26,28
100	0,1	0,6	0,2	0,5	796	994	1108,8	0,31	68,87
250	0,1	0,6	0,2	0,5	796	960	1053,5	0,27	69,94
100	0,25	0,6	0,2	0,5	796	1017	1064	0,26	47,27
250	0,25	0,6	0,2	0,5	796	997	1024	0,23	21,62
100	0,1	0,4	0,4	0,5	796	1041	1115,1	0,31	45,22
250	0,1	0,4	0,4	0,5	796	949	1036	0,25	45,94
100	0,25	0,4	0,4	0,5	796	996	1062,5	0,27	55,58
250	0,25	0,4	0,4	0,5	796	960	1018,4	0,23	45,40
100	0,1	0,6	0,4	0,5	796	974	1112,1	0,32	56,26
250	0,1	0,6	0,4	0,5	796	991	1049,9	0,26	42,75
100	0,25	0,6	0,4	0,5	796	1014	1073,2	0,27	36,11
250	0,25	0,6	0,4	0,5	796	982	1025,6	0,23	38,68

Al comparar los resultados obtenidos mediante el algoritmo propuesto con la solución conocida, la cual fue calculada mediante el método exacto, es posible evidenciar que para la instancia k5m2i5, ninguna de las soluciones obtenidas es mejor a dicha solución, dado que su complejidad computacional es baja, por lo cual el método exacto, aunque no encuentra la solución óptima en el tiempo límite establecido (tal como se observó en la sección 3 del capítulo 4), encuentra una solución de buena calidad.

A su vez, es posible observar que la mejor solución obtenida es de 948 unidades de tiempo, la cual se obtiene al ajustar los parámetros de población inicial, porcentaje de imperialistas y porcentaje de revolución de las colonias en sus niveles altos, con los niveles bajos de los factores porcentaje de cruce e influencia de las colonias en el poder del imperio ($\% \xi$)

Tabla 21. Resultados instancia k5m3i3

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1	100	0,1	0,4	0,2	0,3	545	561	581,6	0,0629	14,99
2	250	0,1	0,4	0,2	0,3	545	545	566,3	0,0376	9,56
3	100	0,25	0,4	0,2	0,3	545	560	576,9	0,0553	12,55
4	250	250	0,4	0,2	0,3	545	550	562,4	0,0309	10,1
5	100	0,1	0,6	0,2	0,3	545	556	590,3	0,0767	35,46
6	250	0,1	0,6	0,2	0,3	545	548	570	0,0439	15,56
7	100	0,25	0,6	0,2	0,3	545	554	576,5	0,0546	14,47
8	250	0,25	0,6	0,2	0,3	545	545	556	0,0198	8,16
9	100	0,1	0,4	0,4	0,3	545	548	570,7	0,0450	16,23
10	250	0,1	0,4	0,4	0,3	545	545	564,6	0,0347	11,15
11	100	0,25	0,4	0,4	0,3	545	548	569	0,0422	13,05
12	250	0,25	0,4	0,4	0,3	545	548	560,9	0,0283	9,02
13	100	0,1	0,6	0,4	0,3	545	557	576,7	0,0550	12,39
14	250	0,1	0,6	0,4	0,3	545	548	564,9	0,0352	12,07
15	100	0,25	0,6	0,4	0,3	545	545	566,9	0,0386	11,24
16	250	0,25	0,6	0,4	0,3	545	565	554,1	0,0164	7,37
17	100	0,1	0,4	0,2	0,5	545	545	581	0,0620	14,82
18	250	0,1	0,4	0,2	0,5	545	545	562,1	0,0304	16,49
19	100	0,25	0,4	0,2	0,5	545	545	567,1	0,0390	18,14
20	250	0,25	0,4	0,2	0,5	545	545	563,1	0,0321	10,98
21	100	0,1	0,6	0,2	0,5	545	560	580,1	0,0605	18,34
22	250	0,1	0,6	0,2	0,5	545	545	569,7	0,0434	12,51
23	100	0,25	0,6	0,2	0,5	545	545	576,1	0,0540	14,77
24	250	0,25	0,6	0,2	0,5	545	545	563,8	0,0333	9,81
25	100	0,1	0,4	0,4	0,5	545	548	571,4	0,0462	13,9
26	250	0,1	0,4	0,4	0,5	545	548	567,8	0,0402	13,15
27	100	0,25	0,4	0,4	0,5	545	545	570,7	0,0450	12,32
28	250	0,25	0,4	0,4	0,5	545	545	561,3	0,0290	11,46
29	100	0,1	0,6	0,4	0,5	545	552	588,2	0,0734	26,56
30	250	0,1	0,6	0,4	0,5	545	548	568,8	0,0418	11,37
31	100	0,25	0,6	0,4	0,5	545	552	573,6	0,0499	13,82
32	250	0,25	0,6	0,4	0,5	545	545	559,8	0,0264	11,46

Para la instancia k5m3i3, el algoritmo competitivo imperialista encuentra las mismas soluciones que el método exacto en 14 de las 32 combinaciones en cada una de las cuales predominan los niveles altos de los parámetros: cantidad de países, porcentaje de imperialistas y peso de las colonias en el costo total del imperio. Asimismo el máximo porcentaje de diferencia entre el promedio de las soluciones y la solución conocida es del 7,67%.

Tabla 22. Resultados instancia k5m3i5

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1	100	0,1	0,4	0,2	0,3	819	710	755,8	0,084	26,33
2	250	0,1	0,4	0,2	0,3	819	701	753,1	0,088	34,78
3	100	0,25	0,4	0,2	0,3	819	695	760,9	0,076	39,16
4	250	250	0,4	0,2	0,3	819	707	735,9	0,113	20,84
5	100	0,1	0,6	0,2	0,3	819	727	787,1	0,041	30,87
6	250	0,1	0,6	0,2	0,3	819	707	741,1	0,105	30,11
7	100	0,25	0,6	0,2	0,3	819	709	758,4	0,080	40,54
8	250	0,25	0,6	0,2	0,3	819	696	734,6	0,115	25,55
9	100	0,1	0,4	0,4	0,3	819	726	778,2	0,052	39,02
10	250	0,1	0,4	0,4	0,3	819	687	732,2	0,119	24,8
11	100	0,25	0,4	0,4	0,3	819	709	776	0,055	37,88
12	250	0,25	0,4	0,4	0,3	819	705	733,9	0,116	21,3
13	100	0,1	0,6	0,4	0,3	819	725	770,3	0,063	30,33
14	250	0,1	0,6	0,4	0,3	819	704	741,6	0,104	23,96
15	100	0,25	0,6	0,4	0,3	819	732	763,9	0,072	25,64
16	250	0,25	0,6	0,4	0,3	819	695	718,37	0,140	23,54
17	100	0,1	0,4	0,2	0,5	819	724	790,4	0,036	40,61
18	250	0,1	0,4	0,2	0,5	819	688	744,7	0,100	28
19	100	0,25	0,4	0,2	0,5	819	707	762,6	0,074	39,47
20	250	0,25	0,4	0,2	0,5	819	685	734,5	0,115	28,04
21	100	0,1	0,6	0,2	0,5	819	705	768,7	0,065	44,62
22	250	0,1	0,6	0,2	0,5	819	716	743,4	0,102	30,39
23	100	0,25	0,6	0,2	0,5	819	715	757,1	0,082	35,22
24	250	0,25	0,6	0,2	0,5	819	702	734,6	0,115	29,31
25	100	0,1	0,4	0,4	0,5	819	747	782,9	0,046	26,8
26	250	0,1	0,4	0,4	0,5	819	720	750	0,092	27,66
27	100	0,25	0,4	0,4	0,5	819	718	766,8	0,068	39,06
28	250	0,25	0,4	0,4	0,5	819	692	723,7	0,132	22,11
29	100	0,1	0,6	0,4	0,5	819	724	778,7	0,052	24,53
30	250	0,1	0,6	0,4	0,5	819	691	742,6	0,103	26,55
31	100	0,25	0,6	0,4	0,5	819	728	753,5	0,087	17,48
32	250	0,25	0,6	0,4	0,5	819	688	731,5	0,120	35,45

Para la instancia k5m3i5, el algoritmo competitivo imperialista encuentra en promedio mejores soluciones que el método exacto en cada una de las 32 combinaciones de los parámetros. Por su parte, el mejor valor encontrado es 685 y ocurre cuando la cantidad de países, el porcentaje de imperialistas y el peso de las colonias están en niveles altos.

Tabla 23. Resultados instancia k7m2i3

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1	100	0.10	0.40	0.20	0.30	796	806	863,8	0,0841	42,438
2	250	0.10	0.40	0.20	0.30	796	803	828	0,0399	15,877
3	100	0.25	0.40	0.20	0.30	796	805	844,7	0,0605	31,496
4	250	0.25	0.40	0.20	0.30	796	779	827,8	0,0408	38,328
5	100	0.10	0.60	0.20	0.30	796	840	884,2	0,105	45,122
6	250	0.10	0.60	0.20	0.30	796	773	831,7	0,0462	26,994
7	100	0.25	0.60	0.20	0.30	796	784	855	0,0753	39,341
8	250	0.25	0.60	0.20	0.30	796	800	837,8	0,0522	23,967
9	100	0.10	0.40	0.40	0.30	796	795	883,6	0,1102	65,247
10	250	0.10	0.40	0.40	0.30	796	790	835,6	0,0501	22,486
11	100	0.25	0.40	0.40	0.30	796	801	833,5	0,0468	36,701
12	250	0.25	0.40	0.40	0.30	796	778	819,8	0,0306	17,504
13	100	0.10	0.60	0.40	0.30	796	816	863,3	0,0825	31,798
14	250	0.10	0.60	0.40	0.30	796	776	836,6	0,0523	45,167
15	100	0.25	0.60	0.40	0.30	796	823	844	0,0583	17,55
16	250	0.25	0.60	0.40	0.30	796	776	815,2	0,0247	22,039
17	100	0.10	0.40	0.20	0.50	796	831	870,4	0,0895	29,837
18	250	0.10	0.40	0.20	0.50	796	795	833	0,0465	24,184
19	100	0.25	0.40	0.20	0.50	796	813	870,7	0,0919	39,833
20	250	0.25	0.40	0.20	0.50	796	773	813,3	0,0224	20,833
21	100	0.10	0.60	0.20	0.50	796	828	871,8	0,0915	39,265
22	250	0.10	0.60	0.20	0.50	796	812	852,5	0,0696	26,593
23	100	0.25	0.60	0.20	0.50	796	813	855,3	0,0729	36,848
24	250	0.25	0.60	0.20	0.50	796	818	836,7	0,0498	11,056
25	100	0.10	0.40	0.40	0.50	796	804	873,4	0,0963	52,646
26	250	0.10	0.40	0.40	0.50	796	805	823,8	0,0345	14,801
27	100	0.25	0.40	0.40	0.50	796	801	826,5	0,0381	16,209
28	250	0.25	0.40	0.40	0.50	796	788	827,6	0,0401	21,956
29	100	0.10	0.60	0.40	0.50	796	827	866,2	0,0849	26,591
30	250	0.10	0.60	0.40	0.50	796	772	843,4	0,0614	45,955
31	100	0.25	0.60	0.40	0.50	796	777	852,1	0,0722	40,984
32	250	0.25	0.60	0.40	0.50	796	783	823,7	0,0354	82,3,7

De acuerdo con los resultados presentados en la tabla 23, con 14 de las 32

combinaciones de parámetros establecidas para la validación se encontraron mejores soluciones que la obtenida mediante el método exacto, debido a que éste encontró soluciones con un gap del 44,31% (ver tabla 1, capítulo 4). Se observa que la mayoría de las mejores soluciones se obtienen con los niveles altos de los parámetros cantidad de países y porcentaje de imperialistas.

Tabla 24. Resultados instancia k7m2i5

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1,00	100,00	0.10	0.40	0.20	0.30	1202,00	1233,00	1371,10	0,14	85,45
2,00	250,00	0.10	0.40	0.20	0.30	1202,00	1280,00	1350,20	0,12	54,74
3,00	100,00	0.25	0.40	0.20	0.30	1202,00	1294,00	1382,50	0,14	58,32
4,00	250,00	0,25	0.40	0.20	0.30	1202,00	1171,00	1277,80	0,06	54,50
5,00	100,00	0.10	0.60	0.20	0.30	1202,00	1341,00	1398,70	0,15	43,33
6,00	250,00	0.10	0.60	0.20	0.30	1202,00	1264,00	1318,70	0,09	56,16
7,00	100,00	0.25	0.60	0.20	0.30	1202,00	1303,00	1348,80	0,11	36,53
8,00	250,00	0.25	0.60	0.20	0.30	1202,00	1169,00	1267,20	0,06	65,80
9,00	100,00	0.10	0.40	0.40	0.30	1202,00	1335,00	1411,00	0,16	60,39
10,00	250,00	0.10	0.40	0.40	0.30	1202,00	1243,00	1304,50	0,08	47,01
11,00	100,00	0.25	0.40	0.40	0.30	1202,00	1271,00	1361,70	0,13	58,82
12,00	250,00	0.25	0.40	0.40	0.30	1202,00	1237,00	1318,30	0,09	54,96
13,00	100,00	0.10	0.60	0.40	0.30	1202,00	1333,00	1451,10	0,19	84,48
14,00	250,00	0.10	0.60	0.40	0.30	1202,00	1224,00	1313,30	0,09	38,72
15,00	100,00	0.25	0.60	0.40	0.30	1202,00	1285,00	1369,50	0,13	67,27
16,00	250,00	0.25	0.60	0.40	0.30	1202,00	1250,00	1300,80	0,08	28,32
17,00	100,00	0.10	0.40	0.20	0.50	1202,00	1288,00	1393,30	0,15	80,58
18,00	250,00	0.10	0.40	0.20	0.50	1202,00	1283,00	1349,00	0,11	58,50
19,00	100,00	0.25	0.40	0.20	0.50	1202,00	1293,00	1363,50	0,12	74,44
20,00	250,00	0.25	0.40	0.20	0.50	1202,00	1193,00	1293,90	0,08	70,13
21,00	100,00	0.10	0.60	0.20	0.50	1202,00	1214,00	1330,20	0,11	76,71
22,00	250,00	0.10	0.60	0.20	0.50	1202,00	1191,00	1345,60	0,12	84,25
23,00	100,00	0.25	0.60	0.20	0.50	1202,00	1217,00	1320,30	0,10	56,44
24,00	250,00	0.25	0.60	0.20	0.50	1202,00	1159,00	1298,10	0,08	92,16
25,00	100,00	0.10	0.40	0.40	0.50	1202,00	1297,00	1380,40	0,14	58,88
26,00	250,00	0.10	0.40	0.40	0.50	1202,00	1246,00	1335,20	0,11	70,38
27,00	100,00	0.25	0.40	0.40	0.50	1202,00	1315,00	1351,50	0,11	36,40
28,00	250,00	0.25	0.40	0.40	0.50	1202,00	1172,00	1263,10	0,05	57,00
29,00	100,00	0.10	0.60	0.40	0.50	1202,00	1278,00	1347,40	0,11	73,19
30,00	250,00	0.10	0.60	0.40	0.50	1202,00	1174,00	1264,60	0,05	62,11
31,00	100,00	0.25	0.60	0.40	0.50	1202,00	1285,00	1358,40	0,12	65,92
32,00	250,00	0.25	0.60	0.40	0.50	1202,00	1212,00	1292,00	0,07	48,58

Con base en la información presentada en la tabla anterior, para la instancia k7m2i5 el algoritmo propuesto obtiene mejores soluciones que la del método exacto, al combinar principalmente los niveles altos de los factores cantidad de países, influencia de las colonias en el poder del imperio y cantidad de imperialistas con el nivel bajo del porcentaje de revolución. La mejor solución obtenida fue 1159 unidades de tiempo con una diferencia de 43 unidades de tiempo respecto a la solución conocida.

Tabla 25. Resultados instancia k7m3i3

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1	100	0.10	0.40	0.20	0.30	640,00	672,00	692,50	0,08	15,52
2	250	0.10	0.40	0.20	0.30	640,00	669,00	705,00	0,09	18,60
3	100	0.25	0.40	0.20	0.30	640,00	653,00	698,20	0,08	33,09
4	250	250	0.40	0.20	0.30	640,00	650,00	676,10	0,05	16,96
5	100	0.10	0.60	0.20	0.30	640,00	696,00	720,50	0,11	22,55
6	250	0.10	0.60	0.20	0.30	640,00	635,00	680,90	0,06	20,58
7	100	0.25	0.60	0.20	0.30	640,00	659,00	692,00	0,08	27,68
8	250	0.25	0.60	0.20	0.30	640,00	635,00	669,60	0,04	19,74
9	100	0.10	0.40	0.40	0.30	640,00	676,00	706,30	0,09	23,88
10	250	0.10	0.40	0.40	0.30	640,00	648,00	688,80	0,07	21,91
11	100	0.25	0.40	0.40	0.30	640,00	668,00	694,20	0,08	18,55
12	250	0.25	0.40	0.40	0.30	640,00	644,00	678,40	0,06	19,55
13	100	0.10	0.60	0.40	0.30	640,00	676,00	706,40	0,09	29,67
14	250	0.10	0.60	0.40	0.30	640,00	635,00	699,50	0,09	30,20
15	100	0.25	0.60	0.40	0.30	640,00	653,00	680,40	0,06	18,07
16	250	0.25	0.60	0.40	0.30	640,00	649,00	682,40	0,06	21,74
17	100	0.10	0.40	0.20	0.50	640,00	649,00	709,70	0,10	38,50
18	250	0.10	0.40	0.20	0.50	640,00	662,00	690,80	0,07	19,92
19	100	0.25	0.40	0.20	0.50	640,00	661,00	696,30	0,08	28,64
20	250	0.25	0.40	0.20	0.50	640,00	625,00	669,20	0,04	23,43
21	100	0.10	0.60	0.20	0.50	640,00	667,00	689,78	0,07	13,41
22	250	0.10	0.60	0.20	0.50	640,00	626,00	669,80	0,04	22,22
23	100	0.25	0.60	0.20	0.50	640,00	637,00	678,00	0,06	22,63
24	250	0.25	0.60	0.20	0.50	640,00	639,00	683,30	0,06	47,84
25	100	0.10	0.40	0.40	0.50	640,00	662,00	705,80	0,09	22,37
26	250	0.10	0.40	0.40	0.50	640,00	650,00	682,20	0,06	18,30
27	100	0.25	0.40	0.40	0.50	640,00	650,00	688,40	0,07	22,53
28	250	0.25	0.40	0.40	0.50	640,00	625,00	669,50	0,04	669,50
29	100	0.10	0.60	0.40	0.50	640,00	671,00	701,80	0,09	21,53
30	250	0.10	0.60	0.40	0.50	640,00	670,00	687,60	0,07	9,67
31	100	0.25	0.60	0.40	0.50	640,00	655,00	680,20	0,06	15,58
32	250	0.25	0.60	0.40	0.50	640,00	657,00	677,10	0,05	20,94

Para la instancia k7m3i3, la metaheurística propuesta encuentra 8 secuencias cuyo *makespan* es inferior al conocido, al igual que en las instancias anteriores, la mayoría de soluciones son obtenidas al combinar los niveles altos de los parámetros cantidad de países y porcentaje de influencia de las colonias en el imperio, con el nivel bajo de el porcentaje de revolución.

Tabla 26. Resultados instancia k7m3i5

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1	100,00	0.10	0.40	0.20	0.30	924,00	1012,00	1136,80	0,19	66,08
2	250,00	0.10	0.40	0.20	0.30	924,00	1010,00	1058,00	0,13	47,81
3	100,00	0.25	0.40	0.20	0.30	924,00	1083,00	1123,80	0,18	25,00
4	250,00	0,25	0.40	0.20	0.30	924,00	985,00	1071,40	0,14	53,40
5	100,00	0.10	0.60	0.20	0.30	924,00	1068,00	1134,90	0,19	37,33
6	250,00	0.10	0.60	0.20	0.30	924,00	997,00	1105,50	0,16	68,46
7	100,00	0.25	0.60	0.20	0.30	924,00	1017,00	1110,70	0,17	48,27
8	250,00	0.25	0.60	0.20	0.30	924,00	984,00	1054,60	0,12	43,88
9	100,00	0.10	0.40	0.40	0.30	924,00	1065,00	1114,70	0,17	38,43
10	250,00	0.10	0.40	0.40	0.30	924,00	1027,00	1116,30	0,17	61,06
11	100,00	0.25	0.40	0.40	0.30	924,00	1070,00	1112,30	0,17	36,81
12	250,00	0.25	0.40	0.40	0.30	924,00	989,00	1058,10	0,13	40,54
13	100,00	0.10	0.60	0.40	0.30	924,00	1093,00	1165,20	0,21	60,73
14	250,00	0.10	0.60	0.40	0.30	924,00	1025,00	1097,70	0,16	43,64
15	100,00	0.25	0.60	0.40	0.30	924,00	1058,00	1118,90	0,17	34,49
16	250,00	0.25	0.60	0.40	0.30	924,00	1009,00	1040,40	0,11	28,39
17	100,00	0.10	0.40	0.20	0.50	924,00	1081,00	1158,20	0,20	53,79
18	250,00	0.10	0.40	0.20	0.50	924,00	995,00	1071,60	0,14	51,94
19	100,00	0.25	0.40	0.20	0.50	924,00	994,00	1121,00	0,18	53,15
20	250,00	0.25	0.40	0.20	0.50	924,00	980,00	1043,40	0,11	54,76
21	100,00	0.10	0.60	0.20	0.50	924,00	1027,00	1129,80	0,18	55,69
22	250,00	0.10	0.60	0.20	0.50	924,00	1059,00	1132,40	0,18	50,66
23	100,00	0.25	0.60	0.20	0.50	924,00	1004,00	1110,60	0,17	78,19
24	250,00	0.25	0.60	0.20	0.50	924,00	978,00	1035,50	0,11	46,70
25	100,00	0.10	0.40	0.40	0.50	924,00	1070,00	1150,10	0,20	53,49
26	250,00	0.10	0.40	0.40	0.50	924,00	1031,00	1082,20	0,15	36,91
27	100,00	0.25	0.40	0.40	0.50	924,00	1041,00	1113,90	0,17	50,33
28	250,00	0.25	0.40	0.40	0.50	924,00	1003,00	1054,50	0,12	39,83
29	100,00	0.10	0.60	0.40	0.50	924,00	1042,00	1132,70	0,18	49,26
30	250,00	0.10	0.60	0.40	0.50	924,00	1023,00	1088,60	0,15	36,75
31	100,00	0.25	0.60	0.40	0.50	924,00	1014,00	1121,30	0,18	49,21
32	250,00	0.25	0.60	0.40	0.50	924,00	1021,00	1083,50	0,15	41,92

Teniendo en cuenta la información presentada en la tabla anterior, para la instancia k7m3i5 ninguna de las soluciones obtenidas es igual o inferior a la solución conocida, el mejor valor obtenido se encuentra al combinar en sus niveles altos 4 de los 5 factores considerados, a su vez la diferencia entre el promedio de las réplicas de la combinación que obtuvo el mejor resultado y la solución conocida, corresponde al 11%, siendo éste el porcentaje más bajo.

Tabla 27. Resultados instancia k9m2i3

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1	100	0,1	0,4	0,2	0,3	1004	992	1082,1	0,072	81,3913
2	250	0,1	0,4	0,2	0,3	1004	968	1020,7	0,016	43,80779
3	100	0,25	0,4	0,2	0,3	1004	962	1039,1	0,034	31,4305
4	250	250	0,4	0,2	0,3	1004	935	1003	0,001	42,8316
5	100	0,1	0,6	0,2	0,3	1004	967	1045	0,039	52,0048
6	250	0,1	0,6	0,2	0,3	1004	966	1042,9	0,037	56,6303
7	100	0,25	0,6	0,2	0,3	1004	944	1040,5	0,035	70,246
8	250	0,25	0,6	0,2	0,3	1004	981	1026	0,021	42,0984
9	100	0,1	0,4	0,4	0,3	1004	997	1051,7	0,045	33,7476
10	250	0,1	0,4	0,4	0,3	1004	923	1035,2	0,030	67,5044
11	100	0,25	0,4	0,4	0,3	1004	990	1042	0,036	39,7939
12	250	0,25	0,4	0,4	0,3	1004	962	1008	0,004	37,3423
13	100	0,1	0,6	0,4	0,3	1004	943	1081,4	0,072	81,0352
14	250	0,1	0,6	0,4	0,3	1004	930	991,7	0,012	50,0667
15	100	0,25	0,6	0,4	0,3	1004	988	1037,1	0,032	46,8863
16	250	0,25	0,6	0,4	0,3	1004	939	985,2	0,019	39,5693
17	100	0,1	0,4	0,2	0,5	1004	973	1090,8	0,080	64,2301
18	250	0,1	0,4	0,2	0,5	1004	928	1017	0,013	55,4336
19	100	0,25	0,4	0,2	0,5	1004	999	1054,2	0,048	50,9941
20	250	0,25	0,4	0,2	0,5	1004	973	1008,8	0,005	29,8991
21	100	0,1	0,6	0,2	0,5	1004	1002	1071,2	0,063	59,5293
22	250	0,1	0,6	0,2	0,5	1004	963	1014,6	0,010	38,6212
23	100	0,25	0,6	0,2	0,5	1004	1020	1049,9	0,044	28,8769
24	250	0,25	0,6	0,2	0,5	1004	948	999	0,005	29,556
25	100	0,1	0,4	0,4	0,5	1004	981	1079,9	0,070	72,6597
26	250	0,1	0,4	0,4	0,5	1004	968	1007,4	0,003	26,6383
27	100	0,25	0,4	0,4	0,5	1004	977	1049	0,043	61,1664
28	250	0,25	0,4	0,4	0,5	1004	955	988,3	0,016	35,6715
29	100	0,1	0,6	0,4	0,5	1004	1011	1077,3	0,068	50,458
30	250	0,1	0,6	0,4	0,5	1004	978	1032,3	0,027	24,0049
31	100	0,25	0,6	0,4	0,5	1004	971	1053,2	0,047	52,3722
32	250	0,25	0,6	0,4	0,5	1004	952	991,3	0,013	28,4373

Cuando se ejecuta el ICA propuesto en este proyecto para la instancia k9m2i3 se observa que el mejor *makespan* se obtiene cuando al menos 2 de los parámetros: cantidad de países, porcentaje de imperialistas y porcentaje de información compartida entre colonias están en sus valores máximos.

Tabla 28. Resultados instancia k9m2i5

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1	100	0,1	0,4	0,2	0,3	1378	1623	1693,5	0,186	45,79
2	250	0,1	0,4	0,2	0,3	1378	1410	1564,2	0,119	87,13
3	100	0,25	0,4	0,2	0,3	1378	1505	1628,7	0,154	97,30
4	250	250	0,4	0,2	0,3	1378	1479	1555,1	0,114	70,63
5	100	0,1	0,6	0,2	0,3	1378	1538	1687	0,183	82,28
6	250	0,1	0,6	0,2	0,3	1378	1417	1582,3	0,129	118,67
7	100	0,25	0,6	0,2	0,3	1378	1573	1662,3	0,171	92,49
8	250	0,25	0,6	0,2	0,3	1378	1458	1557,8	0,115	65,51
9	100	0,1	0,4	0,4	0,3	1378	1571	1658,5	0,169	69,34
10	250	0,1	0,4	0,4	0,3	1378	1447	1522,6	0,095	84,30
11	100	0,25	0,4	0,4	0,3	1378	1553	1636	0,158	76,41
12	250	0,25	0,4	0,4	0,3	1378	1477	1563,5	0,119	104,56
13	100	0,1	0,6	0,4	0,3	1378	1562	1708,9	0,194	95,56
14	250	0,1	0,6	0,4	0,3	1378	1512	1592,6	0,135	70,99
15	100	0,25	0,6	0,4	0,3	1378	1455	1600,4	0,139	85,32
16	250	0,25	0,6	0,4	0,3	1378	1412	1524,8	0,096	67,14
17	100	0,1	0,4	0,2	0,5	1378	1509	1613,5	0,146	69,81
18	250	0,1	0,4	0,2	0,5	1378	1469	1609	0,144	73,91
19	100	0,25	0,4	0,2	0,5	1378	1508	1609,5	0,144	87,36
20	250	0,25	0,4	0,2	0,5	1378	1418	1538	0,104	91,11
21	100	0,1	0,6	0,2	0,5	1378	1510	1688,6	0,184	98,92
22	250	0,1	0,6	0,2	0,5	1378	1444	1566,6	0,120	73,59
23	100	0,25	0,6	0,2	0,5	1378	1464	1632,3	0,156	97,74
24	250	0,25	0,6	0,2	0,5	1378	1460	1496,7	0,079	29,68
25	100	0,1	0,4	0,4	0,5	1378	1620	1755,2	0,215	102,81
26	250	0,1	0,4	0,4	0,5	1378	1475	1561,3	0,117	65,53
27	100	0,25	0,4	0,4	0,5	1378	1564	1692,2	0,186	62,31
28	250	0,25	0,4	0,4	0,5	1378	1420	1561,2	0,117	101,05
29	100	0,1	0,6	0,4	0,5	1378	1516	1713,1	0,196	117,10
30	250	0,1	0,6	0,4	0,5	1378	1526	1620,1	0,149	105,50
31	100	0,25	0,6	0,4	0,5	1378	1556	1659,9	0,170	66,55
32	250	0,25	0,6	0,4	0,5	1378	1382	1498,3	0,080	61,18

Teniendo en cuenta la información presentada en la tabla 28, se concluye que para la instancia k9m2i5 el algoritmo no encuentra un promedio de soluciones cercanas al *makespan* obtenido con el método exacto, observando que el porcentaje de diferencia más pequeño entre estas dos soluciones es de 7,93 y es presentado cuando el nivel bajo de porcentaje de revolución se combina con los niveles altos de los demás factores.

Tabla 29. Resultados instancia k9m3i3

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1	100	0,1	0,4	0,2	0,3	712	806	891,7	0,202	64,22
2	250	0,1	0,4	0,2	0,3	712	809	845,5	0,158	44,79
3	100	0,25	0,4	0,2	0,3	712	816	855,6	0,168	28,38
4	250	250	0,4	0,2	0,3	712	753	811,1	0,122	26,74
5	100	0,1	0,6	0,2	0,3	712	760	897	0,206	66,05
6	250	0,1	0,6	0,2	0,3	712	771	835,7	0,148	46,91
7	100	0,25	0,6	0,2	0,3	712	790	829,9	0,142	24,86
8	250	0,25	0,6	0,2	0,3	712	781	821,7	0,134	26,49
9	100	0,1	0,4	0,4	0,3	712	796	869,1	0,181	43,62
10	250	0,1	0,4	0,4	0,3	712	788	839,3	0,152	52,40
11	100	0,25	0,4	0,4	0,3	712	808	844,9	0,157	40,56
12	250	0,25	0,4	0,4	0,3	712	777	807,1	0,118	17,32
13	100	0,1	0,6	0,4	0,3	712	807	857	0,169	47,12
14	250	0,1	0,6	0,4	0,3	712	752	847,8	0,160	51,13
15	100	0,25	0,6	0,4	0,3	712	790	847,3	0,160	38,80
16	250	0,25	0,6	0,4	0,3	712	796	814,5	0,126	15,34
17	100	0,1	0,4	0,2	0,5	712	813	838,5	0,151	20,64
18	250	0,1	0,4	0,2	0,5	712	792	834,7	0,147	27,90
19	100	0,25	0,4	0,2	0,5	712	797	835,4	0,148	31,01
20	250	0,25	0,4	0,2	0,5	712	772	816,3	0,128	25,96
21	100	0,1	0,6	0,2	0,5	712	804	877,9	0,189	53,80
22	250	0,1	0,6	0,2	0,5	712	772	843,1	0,155	44,80
23	100	0,25	0,6	0,2	0,5	712	819	861,8	0,174	42,64
24	250	0,25	0,6	0,2	0,5	712	770	811	0,122	28,40
25	100	0,1	0,4	0,4	0,5	712	832	880,9	0,192	55,25
26	250	0,1	0,4	0,4	0,5	712	790	828,8	0,141	23,25
27	100	0,25	0,4	0,4	0,5	712	797	850,9	0,163	27,14
28	250	0,25	0,4	0,4	0,5	712	737	833,9	0,146	66,03
29	100	0,1	0,6	0,4	0,5	712	821	838	0,150	13,37
30	250	0,1	0,6	0,4	0,5	712	736	830,9	0,143	42,06
31	100	0,25	0,6	0,4	0,5	712	796	860,1	0,172	39,81
32	250	0,25	0,6	0,4	0,5	712	719	820,8	0,133	44,28

Según la información presentada en la tabla 29, el método exacto arroja mejores soluciones que el ICA en cada una de las combinaciones siendo mejor el *makespan* de éste que el promedio de las soluciones del algoritmo e incluso que las mejores encontradas, observándose que la diferencia más pequeña entre SC y MO es de 7 unidades.

Tabla 30. Resultados instancia k9m3i5

N°	CP	% Imp	%Cr	%Rev	% ξ	SC	MO	Med	%dif	SD
1,00	100,00	0.10	0.40	0.20	0.30	1127,00	1278,00	1413,90	0,20	44,73
2,00	250,00	0.10	0.40	0.20	0.30	1127,00	1275,00	1336,20	0,16	63,82
3,00	100,00	0.25	0.40	0.20	0.30	1127,00	1320,00	1383,10	0,19	52,69
4,00	250,00	250,00	0.40	0.20	0.30	1127,00	1198,00	1280,30	0,12	49,44
5,00	100,00	0.10	0.60	0.20	0.30	1127,00	1269,00	1432,80	0,21	165,05
6,00	250,00	0.10	0.60	0.20	0.30	1127,00	1183,00	1282,30	0,12	72,76
7,00	100,00	0.25	0.60	0.20	0.30	1127,00	1317,00	1392,20	0,19	57,34
8,00	250,00	0.25	0.60	0.20	0.30	1127,00	1175,00	1263,30	0,11	48,76
9,00	100,00	0.10	0.40	0.40	0.30	1127,00	1299,00	1413,70	0,20	62,40
10,00	250,00	0.10	0.40	0.40	0.30	1127,00	1166,00	1306,40	0,14	59,71
11,00	100,00	0.25	0.40	0.40	0.30	1127,00	1303,00	1404,80	0,20	61,29
12,00	250,00	0.25	0.40	0.40	0.30	1127,00	1164,00	1275,00	0,12	80,11
13,00	100,00	0.10	0.60	0.40	0.30	1127,00	1270,00	1363,20	0,17	63,91
14,00	250,00	0.10	0.60	0.40	0.30	1127,00	1238,00	1291,80	0,13	33,15
15,00	100,00	0.25	0.60	0.40	0.30	1127,00	1259,00	1357,60	0,17	84,97
16,00	250,00	0.25	0.60	0.40	0.30	1127,00	1236,00	1292,20	0,13	44,39
17,00	100,00	0.10	0.40	0.20	0.50	1127,00	1286,00	1362,20	0,17	76,02
18,00	250,00	0.10	0.40	0.20	0.50	1127,00	1249,00	1325,00	0,15	42,96
19,00	100,00	0.25	0.40	0.20	0.50	1127,00	1250,00	1365,90	0,17	67,27
20,00	250,00	0.25	0.40	0.20	0.50	1127,00	1217,00	1285,40	0,12	48,42
21,00	100,00	0.10	0.60	0.20	0.50	1127,00	1314,00	1433,70	0,21	151,18
22,00	250,00	0.10	0.60	0.20	0.50	1127,00	1221,00	1300,40	0,13	56,77
23,00	100,00	0.25	0.60	0.20	0.50	1127,00	1285,00	1367,80	0,18	73,97
24,00	250,00	0.25	0.60	0.20	0.50	1127,00	1151,00	1275,10	0,12	69,38
25,00	100,00	0.10	0.40	0.40	0.50	1127,00	1254,00	1372,90	0,18	65,59
26,00	250,00	0.10	0.40	0.40	0.50	1127,00	1239,00	1315,50	0,14	66,56
27,00	100,00	0.25	0.40	0.40	0.50	1127,00	1264,00	1423,30	0,21	129,55
28,00	250,00	0.25	0.40	0.40	0.50	1127,00	1198,00	1278,00	0,12	72,24
29,00	100,00	0.10	0.60	0.40	0.50	1127,00	1257,00	1337,60	0,16	55,23
30,00	250,00	0.10	0.60	0.40	0.50	1127,00	1171,00	1295,40	0,13	64,41
31,00	100,00	0.25	0.60	0.40	0.50	1127,00	1285,00	1358,40	0,17	65,92
32,00	250,00	0.25	0.60	0.40	0.50	1127,00	1170,00	1295,10	0,13	72,13

Para la instancia K9m3i5, la metaheurística propuesta no logra hallar una solución igual o inferior a la mejor solución conocida, uno de los aspectos a considerar es que, al igual que en la instancia anterior, la mejor solución se obtiene al combinar 4 parámetros en sus niveles altos (cantidad de países, porcentaje de imperialistas, porcentaje de revolución e influencia de las colonias en el poder del imperio) con un parámetro en su nivel bajo (porcentaje de cruce).

8. CONCLUSIONES

Al realizar la revisión de la literatura, se encontró que el problema de *flow shop* híbrido con máquinas paralelas no relacionadas, es uno de los que más se aproximan a entornos reales de manufactura y al mismo tiempo ha sido poco abordado en comparación con el *flow shop* híbrido con máquinas idénticas.

Se evidenció la alta complejidad computacional del problema al implementar un método exacto para la búsqueda de la solución, puesto que el consumo de tiempo y la memoria computacional utilizada fueron altos comparados con lo requerido por la metaheurística propuesta; es decir, mientras que el ICA tardó un tiempo máximo alrededor de 70 segundos en dar una solución, el método exacto tardó 584847 segundos (aproximadamente 7 días) en arrojar una solución cuyo gap fue de 51.91%, estos resultados fueron para la instancia k7m2i5.

Al utilizar como método de solución el método exacto, se pudo establecer que existe una relación directa entre la complejidad computacional de las instancias y la calidad de la solución obtenida; puesto que el software Gams encontró mejores soluciones para las instancias pequeñas, llegando incluso a calcular el *makespan* óptimo, en el caso contrario, para las instancias de mayor complejidad el gap entre la solución obtenida y la óptima supera en algunos casos el 50%.

Mediante el desarrollo de la presente investigación, se comprobó que a pesar que el algoritmo competitivo imperialista fue desarrollado recientemente, su estructura permite obtener soluciones de buena calidad, dado que para la mayoría de las instancias el algoritmo igualó y superó al menos una vez la solución encontrada por el método exacto.

Al comparar los resultados obtenidos mediante el algoritmo propuesto con los obtenidos por el método exacto es posible concluir que para las instancias de

menor tamaño este último obtiene soluciones de mejor calidad en poco tiempo, sin embargo al aumentar el tamaño de las instancias, la metaheurística propuesta en algunos casos y para ciertas combinaciones de los parámetros alcanza una mejor solución en un tiempo muy inferior comparado con el tiempo requerido por la solución conocida (método exacto).

Al realizar el diseño experimental fraccionado de medio nivel, fue posible identificar que para la mayoría de las instancias evaluadas los parámetros cantidad de países y el porcentaje de imperialistas combinados en sus niveles altos son los efectos principales que inciden significativamente en la minimización de la variable respuesta que en la presente investigación corresponde al *makespan*.

9. RECOMENDACIONES

Modificar los operadores de asimilación, cruce de información entre colonias y revolución y comparar los resultados obtenidos con los actuales, para de esta forma establecer si existen diferencias significativas entre los resultados obtenidos y a su vez si al cambiar los operadores mencionados se obtienen soluciones de mejor calidad.

Desarrollar una nueva metaheurística aplicada a la solución del problema de *flow shop* híbrido con máquinas paralelas no relacionadas y validarla utilizando las instancias generadas en la presente investigación, para así comparar los resultados y poder establecer la eficiencia del algoritmo propuesto por los autores del proyecto.

Dada la alta complejidad computacional del problema, se sugiere para futuras investigaciones aplicar métodos de relajación matemática al modelo propuesto, que permitan que mediante el método exacto se obtengan soluciones de mejor calidad en menor o igual tiempo.

Incentivar investigaciones futuras en el tema de *flow shop* híbrido con máquinas paralelas no relacionadas, dado que no es un tema comúnmente abordado por los investigadores del área de programación de operaciones a nivel mundial, para ello se sugiere considerar restricciones como capacidad y recursos limitados, tiempos de transferencia entre etapas de procesamiento, zonas de almacenamiento entre etapas de procesamiento, entre otras.

En el presente proyecto se elaboró una guía para la validación de un modelo matemático en Gams para que sea utilizada en futuros trabajos de investigación; en busca de generar herramientas para los estudiantes del grupo Ópalo.

BIBLIOGRAFIA

ALMEDER, Christian y HARTL, Richard F. A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer. En: International Journal of Production Economics, 2013, vol. 145, no 1, p. 88-95.

ARANGO-MARÍN, Jaime Antero, et al. Applying TOC Heuristics to Job Scheduling in a Hybrid Flexible Flow Shop. En: Dyna, 2014, vol. 81, no 186, p. 113-119.

ATASHPAZ-GARGARI, Esmail y LUCAS, Caro. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. En Evolutionary computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007. p. 4661-4667.

ATTAR, S., et al. Hybrid flexible flowshop scheduling problem with unrelated parallel machines and limited waiting times. En: The International Journal of Advanced Manufacturing Technology, 2013, vol. 68, no 5-8, p. 1583-1599.

BŁAŻEWICZ, Jacek, et al. Handbook on scheduling: from theory to applications. Springer Science & Business Media, 2007.

BOZORGIRAD, Mir Abbas y LOGENDRAN, Rasaratnam. Hybrid flowshop scheduling problem with a bi-criteria objective and group technology assumption. En: IIE Annual Conference. Proceedings. Institute of Industrial Engineers-Publisher, 2013. p. 3004.

CHEN, Chun-Lung y CHEN, Chuen-Lung. A bottleneck-based heuristic for minimizing makespan in a flexible flow line with unrelated parallel machines. En: Computers & Operations Research, 2009, vol. 36, no 11, p. 3073-3081.

COSTA, Antonio et al. A novel genetic algorithm for the hybrid flow shop scheduling with parallel batching and eligibility constraints. En: The International Journal of Advanced Manufacturing Technology, 2014, vol. 75, no 5-8, p. 833-847.

CRUZ, Marco, et al. Aplicación de la teoría de la complejidad en optimización combinatoria. Inventio, la génesis de la cultura universitaria en Morelos, 2014, no 20, p. 35.

DAI, Min, et al. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. En: Robotics and Computer-Integrated Manufacturing, 2013, vol. 29, no 5, p. 418-429.

DE FATIMA MORAIS, Marcia, et al. Multicriteria hybrid flow shop scheduling problem: literature review, analysis, and future research. En Independent Journal of Management & Production, 2014, vol. 5, no 4, p. 1004-1031.

DE MICHELI, Mauro, et al. Algoritmo de búsqueda local iterativa para la programación de piezas en un sistema flow shop híbrido. 2009.

DE SIQUEIRA, Eduardo, et al. um algoritmo iterated greedy search aplicado à minimização do makespan no problema flowline híbrido e flexível.

DEFERSHA, Fantahun y CHEN, Mingyuan. Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem. En: The International Journal of Advanced Manufacturing Technology, 2012, vol. 62, no 1-4, p. 249-265.

DUARTE, Abraham, et al. Metaheurísticas. Madrid: Dykinson, 2007. p 1.

ELMI, Atabak y TOPALOGLU, Seyda. A scheduling problem in blocking hybrid flow shop robotic cells with multiple robots. En: & Operations Research, 2013, vol. 40, no 10, p. 2543-2555.

FIGIELSKA, Ewa. A genetic algorithm and a simulated annealing algorithm combined with column generation technique for solving the problem of scheduling in the hybrid flowshop with additional resources. En: Computers & Industrial Engineering, 2009, vol. 56, no 1, p. 142-151.

FIGIELSKA, Ewa. A heuristic for scheduling in a two-stage hybrid flowshop with renewable resources shared among the stages. En: European Journal of Operational Research, 2014, vol. 236, no 2, p. 433-444.

FIGIELSKA, Ewa. A new heuristic for scheduling the two-stage flowshop with additional resources. En: Computers & Industrial Engineering, 2008, vol. 54, no 4, p. 750-763.

FIGIELSKA, Ewa. Heuristic algorithms for preemptive scheduling in a two-stage hybrid flowshop with additional renewable resources at each stage. En: Computers & Industrial Engineering, 2010, vol. 59, no 4, p. 509-51.

FISHBACK, Paul E. Linear and nonlinear programming with Maple: an interactive, applications-based approach. CRC Press, 2009.

GÓMEZ, Pedro et al. An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimise makespan. En: Expert Systems with Applications, 2012, vol. 39, no 9, p. 8095-8107.

GÓMEZ, Pedro. Programación de la producción en un taller de flujo híbrido sujeto a incertidumbre: arquitectura y algoritmos. Aplicación a la industria cerámica. 2010. Tesis Doctoral.

GÓMEZ, Pedro. Un nuevo Algoritmo Genético Basado en un Sistema Multiagente para la Programación de la Producción en un Taller de Flujo Híbrido. En XI Congreso de Ingeniería de Organización. 2007. p. 1675-1685.

IGLESIAS, José et al. Complejidad Computacional. Grado Ingeniería informática. Teoría de Autómatas y Lenguas formales

JENABI, Masoud, et al. Two hybrid meta-heuristics for the finite horizon ELSP in flexible flow lines with unrelated parallel machines. En: Applied Mathematics and Computation, 2007, vol. 186, no 1, p. 230-245.

JIMENEZ, Ángela Patricia. Solución del problema de flow shop flexible aplicando el algoritmo genético de Chu-Beasley. En: Entre Ciencia e Ingeniería, 2013, no 13, pp, 34-40.

JIMÉNEZ, Mario y CAPARRINI, Fernando. Máquinas moleculares basadas en ADN. Universidad de Sevilla, 2003.pag 23.

JOHNSON, S.M. Optimal two-and three-stage production schedules with setup times included. En: Naval Research Logistics Quarterly. 1954, no 1. p. 61-68.

JUN, Sungbum y PARK, Jinwoo. A hybrid genetic algorithm for the hybrid flow shop scheduling problem with nighttime work and simultaneous work constraints: A case study from the transformer industry. En: Expert Systems with Applications, 2015, vol. 42, no 15, p. 6196-6204.

JUNGWATTANAKIT, Jitti, et al. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: Computers & Operations Research, 2009, vol. 36, no 2, p. 358-378.

JUNGWATTANAKIT, Jitti, et al. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: Computers & Operations Research, 2009, vol. 36, no 2, p. 358-378.

JUNGWATTANAKIT, Jitti, et al. Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. En: The International Journal of Advanced Manufacturing Technology, 2008, vol. 37, no 3-4, p. 354-370.

JUNGWATTANAKIT, Jitti, et al. Constructive and simulated annealing heuristics for hybrid flow shops with unrelated parallel machines. En: Proceedings of the 3rd OR-CRN Operations Research Conference, Bangkok, Thailand. 2006. p. 110-121.

KIM, H. W y LEE, D. H. Heuristic algorithms for re-entrant hybrid flow shop scheduling with unrelated parallel machines. Proceedings of the Institution of Mechanical Engineers, En: Part B: Journal of Engineering Manufacture, 2009, vol. 223, no 4, p. 433-442.

LI, Jun-qing y PAN, Quan-ke. Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. En: Information Sciences, 2014.

LI, Zhan-tao, et al. Approximation algorithms for the three-stage flexible flow shop problem with mid group constraint. En: Expert Systems with Applications, 2015, vol. 42, no 7, p. 3571-3584.

LI, Zhan-tao, et al. Scheduling rules for two-stage flexible flow shop scheduling problem subject to tail group constraint. En: International Journal of Production Economics, 2013, vol. 146, no 2, p. 667-678.

LINN, Richard y ZHANG, Wei. Hybrid flow shop scheduling: a survey. En: Computers & industrial engineering, 1999, vol. 37, no 1, p. 57-61.

LOW, Chinyao et al. A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines. En: Computers & Operations Research, 2008, vol. 35, no 3, p. 845-853

LOW, Chinyao. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. En: Computers & Operations Research, 2005, vol. 32, no 8, p. 2013-2025.

LUGO, Pedro, et al. Algoritmo genético com busca local para a programação da produção em sistemas flow-shop híbridos. En: Congreso Latino-Iberoamericano de Investigación Operativa (44:24-28, septiembre, Rio de Janeiro, Brasil). Publicaciones. Universidad Federal de Sao Carlos. 2012, p. 19-29.

LUGO, Pedro, et al. Um modelo de programação inteira mista para a programação da produção em flowshop híbrido com buffers limitados. En: Simpósio Brasileiro de Pesquisa Operacional (45: 16-19 septiembre, Natal, Brasil). Publicaciones. Universidad Federal de Sao Carlos. 2013, p. 154-165.

MARTÍ, Rafael y REINELT, Gerhard. The linear ordering problem: exact and heuristic methods in combinatorial optimization. Springer Science & Business Media, 2011.

MARTI, Rafael. Procedimientos metaheurísticos en optimización combinatoria. *Matemátiques*

MENDOZA, Franklin, et al. Herramientas computacionales “CI-API-1”: una alternativa para la solución de problemas en sistemas de distribución de energía eléctrica. En: Memorias de las IX Jornadas de Investigación 2011-UNEXPO Puerto Ordaz, 2011, p. 10-22.

MONTGOMERY, Douglas C. Diseño y análisis de experimentos. Limusa Wiley. 2 ed., 2004.

PARRA Javier y MEJÍA, Gonzalo. Desarrollo de un modelo para la secuenciación de trabajos en la mediana industria ladrillera de la localidad XIX. 2006.

RABIEE, M., et al. An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines. En: The International Journal of Advanced Manufacturing Technology, 2014, vol. 71, no 5-8, p. 1229-1245

RAHMAN, Riza et al. Hybrid Differential Evolution and Bottleneck Heuristic Algorithm to Solve Bi-Objective Hybrid Flow Shop Scheduling Unrelated Parallel Machines Problem.

RIBAS, Imma y PASCUAL, Ramón. Programación de la Producción en un sistema flow shop híbrido sin esperas y tiempos de preparación dependientes de la secuencia. En: ETSEIB-Departament d'Organització d'Empreses, 2006.

RIBAS, Imma, et al. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. En: Computers & Operations Research, 2010, vol. 37, no 8, p. 1439-1454.

RUIZ, Rubén et al. Modeling realistic hybrid flexible flowshop scheduling problems. En: Computers & Operations Research, 2008, vol. 35, no 4, p. 1151-1175

RUIZ, Ruben y MAROTO, Concepción. A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. En: European Journal of Operational Research, 2006, vol. 169, no 3, p. 781-800.

SOLTANI, S. Abolfazl y KARIMI, Behrooz. Cyclic hybrid flow shop scheduling problem with limited buffers and machine eligibility constraints. En: The International Journal of Advanced Manufacturing Technology, 2014, p. 1-17

UETAKE, Toshifumi, et al. A production scheduling system in a hybrid flow shop. En: International journal of production economics, 1995, vol. 41, no 1, p. 395-398.

WANG, Wenpeng. Review on Hybrid Flow Shop Scheduling. En: Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on. IEEE, 2011. p. 7-10

XU, Ye, et al.. An effective shuffled frog leaping algorithm for solving hybrid flow-shop scheduling problem. En: Advanced Intelligent Computing. Springer Berlin Heidelberg, 2012. p. 560-567.

YAGMAHAN, Betul y YENISEY, Mehmet Mutlu. Scheduling practice and recent developments in flow shop and job shop scheduling. En Computational intelligence in flow shop and job shop scheduling. Springer Berlin Heidelberg, 2009. p. 261-300

YEURIMA, Victor et al. Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. En: Computers & Industrial Engineering, 2009, vol. 56, no 4, p. 1452-1463.

YEURIMA, Victor, et al. Hybrid flowshop with unrelated machines, sequence dependent setup time and availability constraints: An enhanced crossover operator for a genetic algorithm. Parallel processing and applied mathematics. En: Springer Berlin Heidelberg, 2008. p. 608-617.

YU, Xinjie y GEN, Mitsuo. Introduction to evolutionary algorithms. Springer Science & Business Media, 2010. P. 263.

ZANDIEH, M., et al. Mansour. A robust genetic algorithm for scheduling realistic hybrid flexible flow line problems. En: Journal of Intelligent Manufacturing, 2010, vol. 21, no 6, p. 731-743.

ANEXOS

Anexo A. Formulación del modelo matemático en la estructura del lenguaje GAMS

SETS

En esta sección se definen los subíndices utilizados para la formulación del modelo, para el caso de estudio corresponden a:

i etapas de procesamiento o centros de trabajo
m máquinas disponibles en cada etapa de procesamiento
k trabajos a ser procesados.

ALIAS

La declaración ALIAS que ofrece GAMS permite declarar varios nombres para un conjunto, para el presente modelo esta declaración es utilizada para el conjunto de trabajos como se muestra a continuación:

J=k=h

PARAMETERS

p (k,m,i): tiempo de procesamiento del trabajo k en la máquina m de la etapa i. En unidades de tiempo.

s (k,j,m,i): tiempo de alistamiento del trabajo k si lo precede el trabajo j en la maquina m en la etapa i

SCALAR

Para la formulación de una de las restricciones de secuencia se definió el siguiente escalar:

U: Penalización con un valor muy grande el cual corresponde a la suma de los tiempos de procesamiento de todos los trabajos en todas las etapas. Este valor

varía en cada una de las instancias con las cuales fue validado el modelo de programación lineal.

VARIABLES

C (k,i): Variable positiva que representa el instante en que finaliza el procesamiento del trabajo k en la etapa i.

Cmax: instante de completamiento de un trabajo.

Makespan: medida de desempeño a optimizar, la cual corresponde al makespan.

BINARY VARIABLES

En esta sección se detalla la variable binaria utilizada en el modelo, la cual se describe a continuación:

Y (k,j,m,i): Toma el valor de 1 si el trabajo j precede al trabajo k en la máquina m de la etapa i, toma el valor de 0 en el caso contrario.

EQUATIONS

En esta sección se definen los 9 bloques de ecuaciones utilizados en el modelo de programación lineal entera mixta.

Formulación del modelo matemático en el lenguaje de GAMS

A continuación se presenta el modelo en lenguaje de GAMS para la instancia de 5 trabajos, 3 etapas de procesamiento y dos máquinas por etapa de procesamiento.

SETS

K trabajos /0*5/

i etapas /1,2,3/

m maquinas /maq1*maq2/;

alias(j,k,h);

DISPLAY k;

table

p (k,m,i) "tiempo de procesamiento de los trabajos k en las maquinas m de la etapa i"

	1	2	3
0.maq1	0	0	0
0.maq2	0	0	0
1.maq1	92	114	91
1.maq2	89	88	121
2.maq1	117	86	95
2.maq2	115	87	108
3.maq1	88	96	119
3.maq2	106	120	118
4.maq1	83	99	117
4.maq2	94	105	89
5.maq1	112	90	79
5.maq2	97	100	115;

table

s(k,j,m,i) "tiempo de alistamiento del trabajo k si lo precede el trabajo j en la maquina m en la etapa i"

	1	2	3
1.0.maq1	43	58	36
1.0.maq2	40	27	59
1.2.maq1	59	28	35
1.2.maq2	58	74	34
1.3.maq1	55	36	55
1.3.maq2	35	46	39
1.4.maq1	39	60	58
1.4.maq2	27	33	39
1.5.maq1	49	29	71
1.5.maq2	26	33	70
2.0.maq1	67	68	62

2.0.maq2	65	39	65
2.1.maq1	27	71	63
2.1.maq2	58	47	44
2.3.maq1	38	35	34
2.3.maq2	60	64	29
2.4.maq1	40	53	48
2.4.maq2	58	69	47
2.5.maq1	40	31	28
2.5.maq2	29	65	70
3.0.maq1	50	41	57
3.0.maq2	67	71	33
3.1.maq1	28	55	42
3.1.maq2	46	69	36
3.2.maq1	63	47	31
3.2.maq2	35	63	42
3.4.maq1	61	59	53
3.4.maq2	39	55	46
3.5.maq1	63	25	39
3.5.maq2	48	42	30
4.0.maq1	74	62	55
4.0.maq2	59	64	73
4.1.maq1	40	67	32
4.1.maq2	43	70	47
4.2.maq1	27	41	70
4.2.maq2	49	71	36
4.3.maq1	28	34	47
4.3.maq2	26	59	31
4.5.maq1	37	35	29
4.5.maq2	74	41	37
5.0.maq1	47	48	61
5.0.maq2	41	45	74
5.1.maq1	43	30	68
5.1.maq2	62	35	61
5.2.maq1	37	62	60
5.2.maq2	26	70	69
5.3.maq1	66	63	54
5.3.maq2	69	48	64
5.4.maq1	52	72	74
5.4.maq2	34	61	51;

scalar u "penalizacion" /3030/;

VARIABLES

y (k,j,m,i) Si al trabajo k lo precede el trabajo j en la máquina m de la etapa i

c(k,i) instante de finalización de la tarea k en la etapa i

cmax instante de completamiento

makespan makespan;

binary variable y

positive variables c(k,i), cmax, makespan;

Equations

e0 makespan

e1 asegura que cada tarea debe ser precedida solamente por una tarea en apenas un procesador en cada estación.

e2 asegura que cada tarea tiene máximo un sucesor

e3 indica que si una tarea es procesada en una máquina en una estación entonces ella debe tener una predecesora en la misma máquina.

e4 la tarea cero no puede ser predecesora de más de un trabajo en cada máquina de cada etapa

e5 evita precedencias cruzadas

e6 garantiza que la tarea ficticia cero sea finalizada en un tiempo cero en todas las estaciones.

e7 garantiza que la tarea k inicia su procesamiento en la máquina m de la etapa i cuando la máquina m haya sido liberada del trabajo anterior

e8 indica que la tarea j no puede iniciar su procesamiento en la etapa i si no ha terminado su operación en la etapa i-1

e9 calcula el makespan;

e0..makespan=e=cmax;

e1(k,i)\$ (ord(k)>1).. sum((m,j)\$ (not sameas(j,k)), y(k,j,m,i))=E=1;

e2(k,i)\$ (ord(k)>1)..sum((j,m)\$ (not sameas(j,k)), y(j,k,m,i))=I=1;

e3(k,i,m,j)\$ (not sameas(j,k)and (ord(j)>1)and (ord(k)>1)).. sum((h)\$ (not sameas(j,h)and (not (sameas(h,k))))), y(j,h,m,i))=g=y(k,j,m,i);

e4(m,i,j)\$ (ord(j)=1)..sum((k)\$ (ord(k)>1),y(k,j,m,i))=l=1;

e5(k,j,i)\$ (not sameas(j,k) and (ord(k)>1))..sum((m),y(k,j,m,i)+y(j,k,m,i))=l=1;

e6(i,m)..c('0',i)=e=0;

e7(k,m,j,i)\$ (not sameas(j,k) and (ord(k)>1))..c(k,i)+u*(1-y(k,j,m,i))=g=c(j,i)+y(k,j,m,i)*s(k,j,m,i)+p(k,m,i)*y(k,j,m,i);

e8(k,m,j,i)\$ (not sameas(j,k) and (ord(k)>1 and(ord(i)>1)))..c(k,i)-c(k,i-1)=g=y(k,j,m,i)*(s(k,j,m,i)+p(k,m,i));

e9(i,m,k)\$ (ord(i)>1).. cmax=g=c(k,i);

MODEL flowshoph /e0,e1,e2,e3,e4,e5,e6,e7,e8,e9/;

flowshoph.OptFile=1;

option resLim= 100000

option optcr = 0.0000001;

solve flowshoph using mip minimizing makespan;

Anexo B. Instancias utilizadas para la validación del modelo

La información presentada a continuación corresponde a las instancias generadas por los autores del presente trabajo, tomando como referencia lo expuesto por Lugo et al⁸⁷, las cuales fueron utilizadas para la validación del modelo matemático y del algoritmo propuesto.

En cada una de las instancias se presentan los tiempos de alistamiento y de procesamiento.

La notación utilizada en las tablas se presenta a continuación:

k trabajo a procesar

j trabajo precedente

m máquinas

i etapas

Skjmi tiempo de alistamiento del trabajo *k* si es precedido por el trabajo *j* en la máquina *m* de la etapa *i*.

Pkmi tiempo de procesamiento del trabajo *k* en la máquina *m* de la etapa *i*

⁸⁷ LUGO, Pedro, et al. Algoritmo genético com busca local para a programação da produção em sistemas flow-shop híbridos. En: Congreso Latino-Iberoamericano de Investigación Operativa (44:24-28, septiembre, Rio de Janeiro, Brasil). Publicaciones. Universidad Federal de Sao Carlos. 2012, p. 19-29.

Tabla 31. Instancia $k5m2i3$

		<i>Skjmi</i>												<i>Pkmi</i>	
		<i>m2</i>						<i>m1</i>						<i>m1</i>	<i>m2</i>
		<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>		
<i>i1</i>	<i>k1</i>	43		40	55	39	49	40		58	35	27	26	92	89
	<i>k2</i>	67	27		38	40	40	65	58		60	58	29	117	115
	<i>k3</i>	50	28	63		61	63	67	46	35		39	48	88	106
	<i>k4</i>	74	40	27	28		37	59	43	49	26		74	83	94
	<i>k5</i>	47	43	37	66	52		41	62	26	69	34		112	97
<i>i2</i>	<i>k1</i>	58		28	36	60	29	27		74	46	33	33	114	88
	<i>k2</i>	68	71		35	53	31	39	47		64	69	65	86	87
	<i>k3</i>	41	55	47		59	25	71	69	63		55	42	96	120
	<i>k4</i>	62	67	41	34		35	64	70	71	59		41	99	105
	<i>k5</i>	48	30	62	63	72		45	35	70	48	61		90	100
<i>i3</i>	<i>k1</i>	36		35	55	58	71	59		34	39	39	70	91	121
	<i>k2</i>	62	63		34	48	28	65	44		29	47	48	95	108
	<i>k3</i>	47	42	31		53	39	33	36	42		46	30	119	118
	<i>k4</i>	55	32	70	47		29	73	47	36	29		37	117	89
	<i>k5</i>	61	68	60	54	74		74	61	69	64	51		79	115

Tabla 32. Instancia $k5m2i5$

		Skjmi												Pkmi	
		m1						m2						m1	m2
		j0	j1	j2	j3	j4	j5	j0	j1	j2	j3	j4	j5		
i1	k1	31		42	48	29	58	31		53	61	44	32	96	109
	k2	55	50		33	42	29	30	43		70	69	70	78	99
	k3	52	28	31		60	43	52	64	29		58	37	90	76
	k4	60	39	34	42		55	28	51	71	50		70	87	78
	k5	57	70	57	64	40		40	40	55	58	73		94	84
i2	k1	69		71	39	62	34	59		63	66	37	74	94	115
	k2	61	40		34	69	40	69	61		57	32	46	124	116
	k3	73	37	71		33	41	33	67	63		53	71	82	113
	k4	29	61	65	50		71	28	35	62	71		70	116	117
	k5	27	36	44	26	46		37	60	72	45	52		98	81
i3	k1	60		67	34	53	38	50		60	28	28	43	80	104
	k2	44	68		29	73	40	61	48		74	28	61	96	124
	k3	30	43	65		74	56	70	26	29		66	39	85	78
	k4	55	27	44	68		54	61	74	26	37		67	114	92
	k5	42	70	68	41	73		60	52	38	30	41		109	82
i4	k1	73		32	46	28	42	68		36	39	64	65	123	117
	k2	50	32		62	57	53	69	28		49	42	67	80	119
	k3	61	68	28		65	26	38	29	70		72	25	77	113
	k4	46	43	56	34		49	71	60	58	50		59	79	111
	k5	72	26	50	37	62		27	30	63	70	29		85	101
i5	k1	64		65	57	29	71	54		46	62	26	41	119	103
	k2	53	31		27	36	35	62	39		66	51	41	111	94
	k3	56	25	36		61	46	58	54	33		34	71	113	110
	k4	39	36	70	38		27	59	61	63	70		39	94	96
	k5	68	43	28	62	73		45	68	44	35	53		120	79

Tabla 33. Instancia *k5m3i3*

		<i>Skjmi</i>																		<i>Pkmi</i>		
		<i>m1</i>					<i>m2</i>					<i>m3</i>					<i>m1</i>	<i>m2</i>	<i>m3</i>			
		<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>			
<i>i1</i>	<i>k1</i>	60		74	58	27	50	33		40	66	59	27	36		59	73	27	63	77	110	92
	<i>k2</i>	73	59		57	62	50	61	25		72	74	41	52	41		49	68	29	77	112	80
	<i>k3</i>	47	41	56		44	49	42	26	59		69	58	52	56	54		72	34	86	110	90
	<i>k4</i>	58	63	66	28		64	73	64	38	32		39	65	62	30	26		39	99	76	119
	<i>k5</i>	32	39	60	35	46		52	70	28	62	67		34	32	37	36	63		124	79	85
<i>i2</i>	<i>k1</i>	71		64	62	44	72	73		47	64	73	30	57		38	30	26	46	103	92	100
	<i>k2</i>	58	65		31	59	49	47	61		53	29	63	39	28		67	50	36	86	92	119
	<i>k3</i>	72	35	36		63	25	58	59	51		30	27	70	36	28		64	35	99	121	88
	<i>k4</i>	41	40	69	38		47	43	36	40	57		62	74	58	74	60		39	114	84	120
	<i>k5</i>	57	53	68	49	46		63	69	40	57	73		48	71	43	44	29		120	98	90
<i>i3</i>	<i>k1</i>	73		65	50	71	49	66		69	42	65	34	54		64	63	50	39	111	116	83
	<i>k2</i>	54	26		69	47	49	58	73		34	57	67	35	39		53	63	58	90	113	92
	<i>k3</i>	45	66	43		73	51	57	26	53		70	26	59	41	27		50	47	87	103	124
	<i>k4</i>	61	33	62	53		63	64	27	69	53		70	67	58	33	39		72	86	79	111
	<i>k5</i>	36	46	60	37	67		38	65	71	57	70		26	47	71	68	47		113	112	120

Tabla 34. Instancia $k5m3i5$

		Skjmi																		Pkmi		
		m1					m2					m3					m1	m2	m3			
		j0	j1	j2	j3	j4	j5	j0	j1	j2	j3	j4	j5	j0	j1	j2	j3	j4	j5			
<i>i1</i>	<i>k1</i>	43		53	34	61	35	60		59	44	36	69	44		43	42	71	66	93	90	123
	<i>k2</i>	49	73		27	38	60	69	70		65	34	52	52	62		41	73	44	107	105	75
	<i>k3</i>	71	51	30		73	41	26	72	28		41	61	28	42	38		66	26	77	83	90
	<i>k4</i>	71	38	70	40		47	73	28	57	53		63	60	70	31	33		47	119	77	115
	<i>k5</i>	33	68	26	42	37		71	48	46	49	49		65	58	46	73	49		81	103	107
<i>i2</i>	<i>k1</i>	49		51	32	65	72	64		42	38	43	61	71		53	54	36	27	85	99	83
	<i>k2</i>	40	44		41	62	35	58	30		64	64	64	27	34		72	61	51	88	125	112
	<i>k3</i>	69	63	39		33	44	58	37	68		30	74	50	27	43		52	26	118	83	87
	<i>k4</i>	69	64	69	43		41	27	61	63	74		31	71	26	27	48		44	104	88	124
	<i>k5</i>	36	40	60	60	72		35	29	64	31	54		72	41	58	60	36		90	84	102
<i>i3</i>	<i>k1</i>	56		60	67	32	42	59		64	65	60	47	43		49	69	41	54	109	88	100
	<i>k2</i>	26	37		32	52	29	57	40		26	46	61	57	70		56	66	35	77	104	89
	<i>k3</i>	27	59	43		69	35	45	45	29		48	29	50	44	74		32	65	99	113	93
	<i>k4</i>	40	56	30	58		53	42	56	50	46		50	71	74	62	65		72	100	88	99
	<i>k5</i>	65	45	30	42	31		25	29	28	68	62		40	38	48	33	28		100	109	122
<i>i4</i>	<i>k1</i>	47		51	32	42	53	65		68	41	26	37	72		28	36	33	67	100	92	80
	<i>k2</i>	69	39		31	48	57	50	66		34	29	63	40	55		65	27	67	101	115	122
	<i>k3</i>	59	31	66		25	36	60	27	25		60	54	29	68	29		27	47	99	104	96
	<i>k4</i>	73	36	42	58		58	59	72	44	45		42	37	64	49	66		33	100	77	103
	<i>k5</i>	70	60	60	28	31		64	41	72	54	51		55	67	71	52	71		117	109	95
<i>i5</i>	<i>k1</i>	71		52	59	31	53	63		72	42	56	69	64		36	35	27	44	82	104	120
	<i>k2</i>	42	48		61	41	34	34	58		70	65	45	27	54		41	49	39	121	114	101
	<i>k3</i>	30	50	48		51	55	64	31	33		42	59	36	49	53		38	46	101	112	75
	<i>k4</i>	42	62	40	27		73	50	64	38	26		72	72	61	29	26		37	103	94	116
	<i>k5</i>	67	54	42	42	60		47	30	28	67	36		60	36	29	55	31		100	100	110

Tabla 35. Instancia $k7m2i3$

		Skjmi															Pkmi		
		m1							m2							m1	m2		
		<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>		
<i>i1</i>	<i>k1</i>	27		27	35	59	47	53	37	30		71	49	38	30	29	45	112	79
	<i>k2</i>	65	73		34	60	55	54	30	39	44		28	73	42	52	53	90	108
	<i>k3</i>	70	31	30		45	42	31	65	48	30	73		30	36	35	41	90	118
	<i>k4</i>	52	46	71	31		45	52	68	55	61	69	50		54	33	26	118	103
	<i>k5</i>	33	37	55	40	40		31	48	55	32	74	63	55		62	56	112	79
	<i>k6</i>	68	47	42	72	36	55		26	43	55	37	70	70	30		40	87	102
	<i>k7</i>	44	69	51	55	69	40	54		61	44	30	33	64	50	45		107	118
<i>i2</i>	<i>k1</i>	62		70	26	34	25	54	27	40		26	43	59	50	67	69	76	89
	<i>k2</i>	52	44		27	56	65	57	54	47	55		37	71	46	56	42	80	95
	<i>k3</i>	39	34	49		36	55	59	58	64	49	26		31	35	70	53	98	80
	<i>k4</i>	70	60	52		58	45	55	72	62	38	28	36		50	29	71	75	97
	<i>k5</i>	30	71	53	61	47		32	61	37	54	26	69	27		27	31	79	103
	<i>k6</i>	68	51	48	59	43	29		56	55	59	60	61	72	41		29	117	83
	<i>k7</i>	50	41	69	39	62	39	31		26	48	69	48	50	43	55		98	76
<i>i3</i>	<i>k1</i>	60		53	60	66	65	31	41	31		51	35	26	45	53	53	108	88
	<i>k2</i>	53	49		37	74	61	36	70	53	35		53	64	37	48	37	89	78
	<i>k3</i>	48	72	69		36	35	68	45	65	40	47		71	55	35	35	80	97
	<i>k4</i>	27	28	27	52		38	63	68	32	42	45	34		51	35	43	82	114
	<i>k5</i>	43	55	40	61	55		34	28	49	50	43	31	31		62	63	82	112
	<i>k6</i>	58	39	64	40	74	46		55	36	28	38	33	43	27		50	112	111
	<i>k7</i>	74	32	26	28	65	67	41		47	28	31	73	40	63	48		119	102

Tabla 36. Instancia $k7m2i5$

		Skjmi																Pkmi	
		m1								m2								m1	m2
		j0	j1	j2	j3	j4	j5	j6	j7	j0	j1	j2	j3	j4	j5	j6	j7		
i1	k1	61		26	46	70	55	34	28	64		34	49	55	42	26	65	106	105
	k2	38	42		42	38	67	57	54	53	58		43	39	66	59	44	105	121
	k3	62	62	54		47	41	49	48	29	32	55		69	68	64	61	102	113
	k4	29	63	27	68		56	37	57	39	39	36	40		41	25	73	77	83
	k5	27	55	31	27	65		33	54	40	38	32	27	50		41	34	95	112
	k6	61	27	38	44	34	32		48	43	56	43	47	41	47		71	86	105
	k7	27	28	63	66	65	67	37		61	73	41	42	62	50	37		110	113
i2	k1	47		30	51	41	25	55	31	41		32	65	71	36	55	70	84	100
	k2	56	57		70	46	53	31	25	54	35		46	35	46	57	25	77	108
	k3	64	33	59		42	61	36	60	50	46	50		40	26	68	34	119	112
	k4	25	67	30	64		52	57	63	72	29	38	60		37	59	65	107	122
	k5	67	56	55	30	71		36	33	58	49	58	47	40		65	60	120	123
	k6	51	35	37	28	52	45		65	44	43	34	43	29	73		36	81	94
	k7	52	67	46	61	53	31	59		34	31	56	69	34	33	50		108	93
i3	k1	54		70	27	36	45	61	37	73		56	46	50	50	52	34	77	103
	k2	29	54		48	42	34	28	50	73	41		44	64	39	40	71	106	116
	k3	43	57	59		32	45	37	32	57	28	57		37	30	72	67	121	90
	k4	61	39	45	34		63	36	73	68	64	70	34		61	58	45	100	92
	k5	59	49	66	64	41		49	40	72	66	66	58	36		44	58	103	78
	k6	61	42	72	63	39	57		59	72	71	53	74	73	34		43	78	96
	k7	63	35	38	41	47	44	48		47	43	49	49	53	25	62		100	121
i4	k1	31		44	35	35	36	65	30	56		42	74	36	33	47	39	102	106
	k2	43	40		58	30	48	55	64	55	36		36	44	58	60	33	99	79
	k3	40	63	59		56	72	56	55	69	67	70		59	59	70	67	108	116
	k4	32	61	65	26		61	61	56	70	52	55	68		67	69	33	97	100
	k5	34	57	48	32	47		50	60	72	36	41	57	44		35	42	94	121
	k6	64	56	52	59	70	32		73	68	44	71	70	68	43		72	119	81
	k7	69	37	70	61	42	47	40		72	45	27	57	42	26	29		96	80
i5	k1	59		27	40	37	36	59	51	49		53	46	48	38	30	32	115	124
	k2	34	26		39	63	33	48	63	54	69		58	37	62	59	32	96	103
	k3	52	68	66		43	69	73	57	35	27	30		58	32	46	30	86	104
	k4	55	60	65	74		66	73	70	40	56	63	51		63	53	46	87	125
	k5	49	55	51	38	45		69	69	69	31	36	30	67		73	36	108	75
	k6	48	27	42	52	65	71		31	29	55	59	61	54	51		64	105	98
	k7	69	63	72	70	45	59	33		72	28	65	57	64	33	50		81	76

Tabla 37. Instancia $k7m3i3$

		<i>Skjmi</i>																					<i>Pkmi</i>					
		<i>m1</i>							<i>m2</i>							<i>m3</i>							<i>m1</i>	<i>m2</i>	<i>m3</i>			
		<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>			
<i>i1</i>	<i>k1</i>	70		42	60	67	65	56	53	53		53	64	65	61	73	56	27		71	44	61	35	56	51	76	86	119
	<i>k2</i>	58	71		65	71	47	48	64	66	40		74	52	60	32	60	27	66		26	71	25	30	36	115	98	82
	<i>k3</i>	49	64	46		64	73	74	53	44	50	36		30	31	31	44	60	50	37		73	61	37	45	93	122	111
	<i>k4</i>	27	28	56	46		37	72	57	42	47	27	31		69	73	58	44	53	37	30		29	67	72	94	92	103
	<i>k5</i>	46	67	31	48	40		61	29	46	67	31	48	40		61	29	41	46	52	55	58		73	43	110	103	105
	<i>k6</i>	27	29	55	44	32	41		61	51	41	68	51	69	49		39	73	61	30	31	56	63		54	80	119	104
	<i>k7</i>	43	58	45	72	51	73	41		64	61	59	55	71	65	43		28	33	48	57	38	43	36		84	104	105
<i>i2</i>	<i>k1</i>	60		43	52	47	74	37	35	66		66	26	57	67	31	65	39		59	67	43	44	34	73	92	83	87
	<i>k2</i>	26	40		28	54	32	35	66	64	56		57	70	55	26	26	57	44		54	59	33	39	28	94	108	91
	<i>k3</i>	52	64	73		32	58	63	56	53	43	53		45	59	41	44	60	43	57		71	58	60	57	86	90	113
	<i>k4</i>	38	60	69	27		52	30	65	42	65	58	47		45	54	58	36	30	36	53		36	35	69	81	78	100
	<i>k5</i>	39	42	48	53	73		32	66	65	44	61	60	73		25	27	56	58	26	47	65		50	59	96	75	84
	<i>k6</i>	70	41	28	32	49	44		64	46	38	38	29	26	39		73	40	63	26	56	70	35		50	82	114	81
	<i>k7</i>	49	46	39	46	27	69	55		68	62	28	66	42	69	51		70	35	69	29	48	48	64		106	117	124
<i>i3</i>	<i>k1</i>	30		50	54	29	59	43	72	44		44	35	56	56	53	53	28		53	29	42	59	55	73	115	110	87
	<i>k2</i>	29	64		74	48	72	62	53	31	48		36	30	36	60	41	56	37		44	36	71	47	70	80	94	118
	<i>k3</i>	50	66	60		29	43	70	41	38	71	62		64	64	55	45	44	58	43		36	48	66	34	91	86	115
	<i>k4</i>	68	53	48	57		26	42	45	58	60	65	44		27	34	66	58	58	30	44		57	27	51	103	80	118
	<i>k5</i>	47	73	28	42	51		68	65	66	38	74	32	70		48	70	46	43	47	47	70		47	47	120	79	107
	<i>k6</i>	30	44	37	44	37	65		29	73	62	43	56	28	63		52	73	64	54	29	39	74		67	118	83	90
	<i>k7</i>	37	38	58	55	30	32	47		59	60	27	29	47	42	55		36	53	31	34	65	26	52		86	123	101

Tabla 38. Instancia $k7m3i5$

		<i>Skjmi</i>																					<i>Pkmi</i>					
		<i>m1</i>							<i>m2</i>							<i>m3</i>							<i>m1</i>	<i>m2</i>	<i>m3</i>			
		<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>			
<i>i1</i>	<i>k1</i>	52		68	27	71	37	43	57	26		31	47	31	45	42	61	62		44	42	71	54	35	52	113	107	121
	<i>k2</i>	43	31		59	74	32	40	46	49	58		38	33	59	30	63	37	57		43	64	37	54	74	85	75	77
	<i>k3</i>	34	49	29		43	25	57	61	38	47	60		58	53	34	58	52	68	31		50	72	70	57	105	85	97
	<i>k4</i>	43	54	36	52		67	26	30	29	70	32	28		67	47	48	64	62	48	55		28	55	51	119	84	114
	<i>k5</i>	48	40	56	40	38		63	72	32	49	41	69	60		34	71	64	57	61	54	32		39	36	124	103	82
	<i>k6</i>	53	60	60	42	28	27		36	72	68	48	50	67	45		70	62	33	42	40	70	27		31	102	80	122
	<i>k7</i>	42	47	66	53	50	34	66		38	71	32	67	31	69	56		51	70	63	32	34	34	60		79	90	93
<i>i2</i>	<i>k1</i>	39		54	32	42	54	70	33	40		36	40	32	33	38	53	41		67	38	71	40	42	42	117	104	124
	<i>k2</i>	73	52		41	68	37	48	34	69	69		69	48	25	27	45	44	36		53	55	64	58	42	124	84	103
	<i>k3</i>	26	35	38		48	46	42	26	52	41	39		63	54	71	60	25	68	62		71	69	53	39	92	86	121
	<i>k4</i>	58	38	29	53		70	46	69	66	64	39	73		52	49	48	54	35	62	74		26	71	41	94	91	119
	<i>k5</i>	64	56	65	67	61		58	54	54	46	60	45	40		69	30	32	27	44	51	73		72	39	77	104	115
	<i>k6</i>	64	71	25	47	55	51		68	38	43	68	30	73	55		63	51	43	64	54	27	43		49	121	112	110
	<i>k7</i>	72	48	66	54	71	48	33		51	55	40	63	73	61	60		28	67	32	61	51	69	68		83	113	114
<i>i3</i>	<i>k1</i>	53		56	40	36	50	49	61	56		38	37	62	28	49	53	63		26	42	39	62	64	54	77	121	104
	<i>k2</i>	55	46		26	26	49	68	71	54	64		50	54	34	71	69	46	31		35	44	70	62	43	86	122	93
	<i>k3</i>	35	53	62		59	50	46	62	45	73	53		40	54	41	39	33	40	42		37	31	46	43	125	118	95
	<i>k4</i>	31	70	44	57		37	27	46	67	64	26	46		64	37	57	51	40	56	28		67	64	56	113	91	86
	<i>k5</i>	47	64	56	48	26		35	37	50	41	66	66	62		37	36	41	63	33	32	29		33	34	84	82	81
	<i>k6</i>	29	50	32	59	43	31		28	34	64	67	59	39	27		51	58	72	32	49	35	32		47	77	119	112
	<i>k7</i>	68	69	42	27	41	65	52		35	65	69	73	31	28	59		59	40	44	53	66	63	68		105	113	80
<i>i4</i>	<i>k1</i>	58		41	73	57	61	35	29	67		36	33	55	41	52	30	44		68	60	65	53	29	52	88	111	117
	<i>k2</i>	68	47		74	41	71	42	63	51	70		31	47	53	59	30	30	27		70	30	66	48	61	116	78	98
	<i>k3</i>	54	67	61		56	43	29	51	26	26	49		65	48	50	40	73	69	64		40	65	67	26	115	93	112
	<i>k4</i>	68	62	54	73		38	61	48	49	29	58	66		58	45	31	59	40	69	33		57	73	71	99	120	83
	<i>k5</i>	38	61	66	29	32		42	39	61	51	41	28	71		65	55	73	55	36	70	36		33	41	119	120	95
	<i>k6</i>	63	57	70	73	72	27		63	28	46	50	43	36	41		49	61	43	33	26	41	42		59	96	88	117
	<i>k7</i>	39	56	41	36	60	28	39		44	43	64	46	60	48	51		65	37	33	57	40	39	41		116	81	84
<i>i5</i>	<i>k1</i>	30		61	56	30	36	42	70	34		72	37	35	74	42	37	54		61	48	39	52	34	31	111	107	92
	<i>k2</i>	35	35		34	59	27	59	72	39	33		63	32	25	63	36	51	67		29	40	48	33	65	99	80	115
	<i>k3</i>	40	36	33		68	63	30	67	49	47	68		36	42	53	34	50	46	72		37	63	60	45	83	81	98
	<i>k4</i>	34	34	26	63		55	68	37	67	64	36	25		57	26	25	61	49	41	38		43	54	49	96	99	86
	<i>k5</i>	39	44	69	63	33		45	74	56	64	28	48	28		28	33	47	58	31	72	71		31	31	120	104	108
	<i>k6</i>	39	39	71	48	58	39		33	53	62	42	30	55	50		73	27	27	39	36	63	39		48	108	124	80
	<i>k7</i>	70	56	72	63	44	34	60		42	26	57	66	63	29	30		64	70	34	50	29	27	32		102	106	113

Tabla 39. Instancia $k9m2i3$

		Skjmi																			Pkmi		
		m1									m2									m1	m2		
		j0	j1	j2	j3	j4	j5	j6	j7	j8	j9	j0	j1	j2	j3	j4	j5	j6	j7	j8	j9		
<i>i1</i>	<i>k1</i>	60		43	59	70	56	27	58	42	53	36		51	57	58	47	32	27	53	46	78	84
	<i>k2</i>	35	60		69	29	63	30	45	39	32	51	59		74	58	59	48	73	74	49	77	100
	<i>k3</i>	71	57	50		69	62	41	29	52	65	47	54	64		48	59	46	27	51	57	121	84
	<i>k4</i>	50	66	55	43		65	66	55	71	44	36	40	37	69		44	52	67	35	61	95	76
	<i>k5</i>	28	45	39	33	43		64	64	42	38	66	35	42	31	57		36	66	42	64	94	86
	<i>k6</i>	59	74	63	55	43	29		72	32	51	59	40	69	36	70	68		50	47	57	100	83
	<i>k7</i>	51	49	36	43	61	43	64		69	32	36	49	33	42	70	57	31		33	28	116	85
	<i>k8</i>	29	39	32	33	26	37	55	55		58	61	36	48	45	28	73	57	64		35	75	82
	<i>k9</i>	49	37	61	70	39	34	57	67	56		64	46	27	57	48	55	67	33	71		76	88
<i>i2</i>	<i>k1</i>	43		29	52	36	28	26	26	49	67	27		30	64	39	39	66	46	27	36	80	92
	<i>k2</i>	63	32		71	31	36	51	68	51	57	48	66		41	54	67	38	42	63	40	107	110
	<i>k3</i>	44	49	41		72	34	39	58	59	37	58	50	33		49	42	64	50	54	45	88	80
	<i>k4</i>	51	68	63	36		54	31	62	42	46	41	67	70	34		72	56	69	50	69	81	105
	<i>k5</i>	43	30	33	70	27		72	37	68	74	71	67	30	70	34		59	36	43	48	108	93
	<i>k6</i>	70	60	42	32	55	28		56	61	68	64	45	38	34	36	34		44	67	72	82	100
	<i>k7</i>	51	40	50	27	65	42	40		63	30	57	26	41	61	32	37	53		60	61	121	110
	<i>k8</i>	52	49	59	44	42	25	39	70		62	73	39	54	47	56	49	54	50		50	118	75
	<i>k9</i>	74	47	33	64	44	63	53	56	54		67	49	41	41	43	52	61	55	35		96	102
<i>i3</i>	<i>k1</i>	71		54	26	61	38	25	39	60	57	70		32	66	35	50	53	36	70	67	84	99
	<i>k2</i>	62	36		38	42	27	57	55	58	59	59	71		44	56	30	41	28	47	37	124	122
	<i>k3</i>	39	35	43		29	62	31	71	49	61	48	28	70		73	71	27	49	37	39	83	81
	<i>k4</i>	33	52	56	66		25	48	58	68	49	57	68	35	33		51	47	44	38	31	88	124
	<i>k5</i>	70	48	30	33	44		71	34	50	53	34	38	65	62	55		49	29	35	70	76	123
	<i>k6</i>	65	66	61	54	69	73		43	62	71	69	64	45	29	68	47		67	63	44	122	85
	<i>k7</i>	32	70	30	34	72	49	65		51	46	68	72	51	49	40	66	37		61	64	92	77
	<i>k8</i>	65	62	39	56	71	30	60	47		53	52	43	45	74	52	43	53	65		41	111	116
	<i>k9</i>	61	66	39	71	34	38	55	56	68		39	56	44	58	29	36	59	38	51		94	122

Tabla 40. Instancia *k9m2i5*

<i>l</i>		<i>Skjmi</i>																				<i>Pkmi</i>	
		<i>m1</i>										<i>m2</i>										<i>m1</i>	<i>m2</i>
		<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j8</i>	<i>j9</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j8</i>	<i>j9</i>		
<i>i1</i>	<i>k1</i>	60		41	57	47	43	66	36	33	60	33		42	49	31	46	36	44	26	52	123	80
	<i>k2</i>	32	40		28	39	48	38	48	48	48	60	35		25	32	32	53	39	46	51	120	76
	<i>k3</i>	28	32	43		37	59	50	35	34	28	27	65	65		35	41	32	58	50	53	82	110
	<i>k4</i>	40	41	29	44		29	29	57	26	44	31	66	31	28		54	48	43	41	40	87	104
	<i>k5</i>	70	29	58	55	27		61	57	45	31	46	67	31	67	29		69	59	26	41	92	106
	<i>k6</i>	45	68	28	32	70	70		34	70	67	30	73	60	54	70	34		53	69	50	101	106
	<i>k7</i>	45	41	54	71	53	63	73		55	65	62	70	57	50	54	35	59		45	48	114	101
	<i>k8</i>	69	59	74	26	70	60	37	50		33	28	34	53	32	39	66	40	58		29	79	77
	<i>k9</i>	72	52	56	43	37	63	33	70	44		35	63	44	38	62	55	59	48	50		88	94
<i>i2</i>	<i>k1</i>	51		71	70	47	35	51	74	26	56	29		74	72	45	44	42	66	66	56	111	104
	<i>k2</i>	58	36		58	63	58	33	73	57	53	31	63		35	25	42	42	40	59	27	100	116
	<i>k3</i>	57	61	35		42	51	35	62	49	67	28	56	71		64	62	66	59	43	65	99	102
	<i>k4</i>	31	30	32	28		38	49	45	45	51	64	69	51	47		29	30	41	72	61	80	88
	<i>k5</i>	47	52	33	28	42		34	41	30	41	58	57	40	60	40		29	51	49	53	85	75
	<i>k6</i>	44	52	59	74	53	51		30	33	27	28	71	46	41	44	32		48	38	73	106	81
	<i>k7</i>	30	64	73	46	41	56	45		39	29	28	66	74	68	65	30	38		38	46	115	82
	<i>k8</i>	27	30	54	54	35	50	54	48		58	33	36	26	45	73	45	71	70		28	99	98
	<i>k9</i>	30	73	42	47	40	44	28	67	70		28	53	66	59	65	28	45	54	30		108	122
<i>i3</i>	<i>k1</i>	38		39	35	54	49	53	60	43	62	67		67	39	65	27	55	45	51	72	77	102
	<i>k2</i>	34	35		42	27	67	59	32	54	42	26	66		31	34	55	64	51	41	39	104	90
	<i>k3</i>	49	71	61		67	68	54	28	32	30	55	45	43		69	47	57	27	61	56	101	105
	<i>k4</i>	44	70	45	41		56	42	42	45	44	51	53	74	40		55	30	48	46	61	88	88
	<i>k5</i>	45	51	45	57	52		27	57	45	37	36	60	66	35	34		62	30	49	71	123	107
	<i>k6</i>	40	70	49	58	29	68		29	47	38	50	47	34	25	61	55		55	63	48	117	109
	<i>k7</i>	31	55	35	41	56	74	38		69	47	51	25	66	40	55	67	46		34	42	112	81
	<i>k8</i>	64	31	44	74	37	31	42	56		62	64	28	51	28	52	63	69	42		71	94	88
	<i>k9</i>	56	48	66	61	53	41	51	63	28		29	50	54	29	68	32	32	26	25		90	108

Tabla 14. (Continuación)

/	Skjmi																					Pkmi	
	m1											m1											
	j0	j1	m1	m2	j4	j5	j6	j7	j8	j9	j0	j1	j2	j3	j4	j5	j6	j7	j8	j9	m1	m2	
i4	k1	34		74	67	26	70	74	30	68	37	48		59	55	48	36	49	28	51	70	102	105
	k2	65	39		65	44	33	41	49	58	34	34	69		67	45	54	44	63	65	60	118	82
	k3	32	55	37		51	53	66	54	64	48	54	47	35		62	34	25	68	61	30	88	77
	k4	48	33	39	47		52	30	35	38	25	60	31	26	45		67	73	74	30	28	101	116
	k5	64	53	50	43	49		49	50	60	31	53	58	72	68	47		57	44	29	60	104	90
	k6	62	26	27	50	71	27		40	51	52	72	38	67	35	47	69		27	66	68	114	119
	k7	63	30	49	31	53	61	45		45	58	32	57	49	49	59	31	57		42	43	121	104
	k8	65	32	31	38	40	50	34	26		42	67	32	51	36	65	29	27	73		49	102	80
	k9	66	26	49	31	35	37	28	48	60		32	48	56	42	29	47	38	71	66		94	112
i5	k1	28		61	44	29	57	57	29	72	65	69		52	41	48	74	53	45	45	55	90	104
	k2	72	56		58	52	44	40	26	51	38	41	74		37	37	48	26	44	27	39	103	92
	k3	37	65	26		51	56	46	37	49	61	74	60	45		49	34	41	47	48	74	120	104
	k4	30	72	42	72		56	74	60	64	26	51	42	39	66		57	38	61	56	52	125	104
	k5	42	32	39	67	39		69	72	71	64	59	31	39	40	68		28	67	47	36	96	115
	k6	31	72	65	37	51	32		50	70	44	36	64	54	41	67	45		50	59	31	111	78
	k7	50	57	67	45	57	40	35		65	55	68	26	32	56	55	49	48		66	46	75	113
	k8	28	48	55	73	70	57	29	52		43	40	61	42	49	39	52	37	38		64	112	82
	k9	32	37	28	26	56	44	37	67	61		35	36	51	51	46	44	37	70	53		113	124

Tabla 41. Instancia $k9m3i3$

		Skjmi																											Pkmi					
		m1									m2									m3														
		j0	j1	j2	j3	j4	j5	j6	j7	j8	j9	j0	j1	j2	j3	j4	j5	j6	j7	j8	j9	j0	j1	j2	j3	j4	j5	j6	j7	j8	j9	m1	m2	m3
i1	k1	40		42	58	30	38	38	65	42	57	39		52	34	40	60	68	28	55	73	48		43	53	33	70	33	61	71	39	115	99	81
	k2	30	43		40	65	57	43	39	65	56	62	61		43	67	74	34	40	41	50	63	44		36	32	57	42	64	39	67	121	93	89
	k3	73	46	50		34	72	49	30	48	59	45	71	67		29	45	40	35	36	54	26	72	44		28	72	64	73	72	58	107	89	84
	k4	51	44	32	74		54	33	26	57	29	47	37	38	48		49	27	39	74	74	53	61	71	68		36	65	34	39	62	90	93	111
	k5	55	57	40	51	57		45	53	27	28	49	48	56	69	53		37	64	65	42	73	29	62	74	67		30	39	37	57	91	83	80
	k6	53	30	37	50	31	38		62	40	39	62	34	55	62	53	30		51	40	63	71	72	61	60	34	59		66	49	28	111	90	112
	k7	49	43	41	59	32	40	31		26	53	47	73	43	74	26	52	71		65	42	49	46	54	61	73	67	39		45	70	89	97	105
	k8	35	43	45	53	64	39	54	48		25	70	47	61	28	69	45	35	33		39	41	53	47	66	41	55	36	62		51	110	80	100
	k9	72	60	61	32	39	64	70	26	46		27	34	63	28	73	73	73	66	48		31	47	52	33	39	48	69	64	42		114	122	118
i2	k1	56		35	53	68	68	57	66	57	33	72		40	72	69	57	32	29	56	72	48		65	35	73	30	44	58	71	40	90	97	111
	k2	70	74		35	69	35	50	36	48	68	35	52		55	27	49	53	60	46	34	32	29		39	28	58	33	65	48	64	92	116	101
	k3	30	67	25		47	30	49	71	50	73	59	26	33		65	54	66	50	60	57	53	40	38		61	64	31	36	29	32	112	101	117
	k4	39	29	62	30		61	56	40	66	66	30	51	35	42		65	66	42	45	44	28	28	66	56		70	62	64	46	71	100	80	99
	k5	43	30	70	54	34		56	35	31	33	30	50	33	29	73		52	69	58	59	43	26	58	67	61		36	52	31	40	90	75	111
	k6	51	73	28	27	49	33		64	46	44	51	34	71	57	34	44		30	27	35	58	54	73	70	52	38		39	37	70	77	105	81
	k7	35	35	51	36	33	55	66		54	44	70	47	54	73	71	43	30		50	67	63	69	65	52	44	40	73		72	54	93	101	123
	k8	38	46	58	46	33	55	42	53		49	51	53	56	48	42	60	38	64		53	43	69	53	30	49	65	66	41		57	79	96	105
	k9	56	40	71	62	56	38	65	48	38		32	54	41	72	32	34	53	49	64		41	26	42	68	30	50	50	31	71		114	102	108
i3	k1	26		28	39	55	74	32	69	61	45	34		33	41	31	74	69	35	54	28	31		39	49	71	34	69	66	34	33	104	90	115
	k2	43	35		70	60	58	73	43	33	74	59	74		65	62	71	73	45	63	71	30	51		34	61	56	30	52	30	50	110	103	87
	k3	67	36	61		55	51	31	51	37	68	35	65	71		27	47	57	49	32	49	68	73	40		63	54	57	50	35	33	85	119	109
	k4	63	67	40	46		73	37	72	67	70	66	71	64	28		28	69	32	51	56	58	55	29	66		45	40	50	73	32	117	109	88
	k5	56	28	74	49	37		43	53	55	47	33	38	68	54	31		69	71	29	31	63	69	32	41	53		41	55	47	52	109	98	85
	k6	35	65	33	41	69	41		31	38	42	63	53	36	29	28	44		45	48	45	45	62	32	27	51	57		35	55	56	82	105	87
	k7	32	56	47	57	27	73	70		51	27	38	28	58	55	43	64	57		37	35	65	43	62	34	40	56	37		72	60	94	119	115
	k8	38	46	58	46	33	55	42	53		49	63	56	63	49	68	56	71	71		57	69	39	60	57	57	50	66	65		66	95	76	115
	k9	30	63	26	31	62	30	26	49	38		42	46	26	60	50	44	63	54	44		31	37	38	64	39	37	37	39	35		92	105	93

Tabla 42. Instancia *k9m3i5*

		<i>Skjmi</i>																											<i>Pkmi</i>					
		<i>m1</i>									<i>m2</i>									<i>m3</i>									<i>m1</i>	<i>m2</i>	<i>m3</i>			
		<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j8</i>	<i>j9</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j8</i>	<i>j9</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j8</i>	<i>j9</i>			
<i>i1</i>	<i>k1</i>	59		59	28	27	41	38	36	54	41	62		32	37	67	41	62	40	73	44	39		41	45	52	41	49	59	64	25	108	117	89
	<i>k2</i>	31	25		66	60	59	49	58	29	48	64	51		38	66	59	46	26	32	29	29	45		65	56	74	54	57	38	69	96	96	95
	<i>k3</i>	67	56	65		41	30	44	73	50	69	28	52	42		32	48	45	68	57	45	51	37	54		42	36	51	46	53	40	101	86	94
	<i>k4</i>	61	30	49	60		68	64	36	60	59	65	29	33	41		28	45	32	27	34	25	35	31	65		65	61	33	36	72	87	83	123
	<i>k5</i>	50	45	64	72	58		55	63	26	71	73	60	61	58	71		65	34	44	52	55	54	52	48	34		42	71	69	43	98	96	78
	<i>k6</i>	59	45	40	60	43	55		66	31	27	62	58	29	63	28	74		69	39	63	39	32	69	53	55	33		25	45	59	109	94	96
	<i>k7</i>	74	49	55	59	69	45	64		47	26	54	44	42	36	53	33	47		54	56	51	39	63	38	45	54	55		63	38	97	124	78
	<i>k8</i>	64	38	28	36	47	39	50	30		68	41	47	74	70	28	55	63	40		32	69	35	64	39	42	32	50	30		59	110	100	122
	<i>k9</i>	33	35	61	33	34	36	38	56	39		26	47	40	54	42	27	61	48	43		52	58	28	35	54	58	44	45	53		119	107	108
<i>i2</i>	<i>k1</i>	39		55	37	54	38	63	47	39	51	66		61	26	30	25	31	42	46	42	26		33	48	65	51	29	33	51	29	88	77	98
	<i>k2</i>	28	37		29	65	65	63	73	61	64	42	48		45	45	41	50	42	44	47	30	45		32	51	56	28	38	45	56	110	111	98
	<i>k3</i>	47	52	45		41	72	48	51	32	60	56	50	41		71	41	40	66	73	35	26	58	60		62	40	45	40	64	28	76	108	109
	<i>k4</i>	60	28	44	43		55	68	39	35	49	55	69	49	58		66	67	33	57	64	63	28	64	60		61	38	61	28	73	116	121	81
	<i>k5</i>	65	48	73	64	29		71	68	74	54	48	72	36	68	43		53	66	42	31	27	30	48	54	72		57	49	29	66	76	78	87
	<i>k6</i>	35	49	69	43	60	28		59	72	56	68	67	27	46	57	57		67	64	66	65	54	41	42	57	58		58	62	33	94	94	124
	<i>k7</i>	59	73	68	53	55	46	64		58	32	46	35	48	64	48	70	61		31	54	26	44	59	40	36	35	64		63	71	103	119	119
	<i>k8</i>	36	71	29	28	53	49	44	53		73	48	58	47	51	27	41	43	50		62	40	44	71	43	33	42	44	50		37	82	113	86
	<i>k9</i>	55	37	54	50	41	26	31	48	42		37	50	53	48	48	34	61	55	70		71	52	50	63	52	43	37	25	61		96	86	100
<i>i3</i>	<i>k1</i>	65		72	69	67	62	71	55	26	34	40		39	45	42	58	40	56	72	63	48		30	62	70	30	63	48	41	56	80	95	93
	<i>k2</i>	42	63		35	45	40	66	41	67	48	44	59		48	57	68	29	49	53	45	35	28		36	57	72	59	42	46	63	78	107	85
	<i>k3</i>	72	71	52		32	37	63	37	61	61	71	51	53		61	58	29	73	56	56	63	61	66		65	32	62	32	27	68	97	117	75
	<i>k4</i>	55	44	56	63		34	25	51	31	48	68	43	66	61		57	66	62	58	29	73	31	28	62		42	31	51	69	56	123	91	80
	<i>k5</i>	70	39	72	41	51		34	41	51	26	46	50	59	31	44		34	61	58	29	25	59	58	34	51		37	69	28	30	92	75	86
	<i>k6</i>	64	50	68	53	62	71		72	30	67	44	27	47	57	41	33		40	47	65	63	58	38	54	69	33		41	34	54	103	105	114
	<i>k7</i>	51	31	55	70	40	32	69		41	32	69	62	44	49	27	25	60		53	29	68	26	64	30	48	51	30		44	36	86	90	106
	<i>k8</i>	48	64	26	27	35	31	39	32		69	68	39	50	36	54	57	65	56		57	26	57	66	60	37	35	45	65		68	123	82	112
	<i>k9</i>	52	69	51	28	59	40	67	61	59		51	68	28	46	72	44	57	57	34		39	33	25	54	47	55	57	63	45		78	77	104

Tabla 16. (Continuación)

		<i>Skjmi</i>																											<i>Pkmi</i>					
		<i>m1</i>									<i>m2</i>									<i>m3</i>									<i>m1</i>	<i>m2</i>	<i>m3</i>			
		<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j8</i>	<i>j9</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j8</i>	<i>j9</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j3</i>	<i>j4</i>	<i>j5</i>	<i>j6</i>	<i>j7</i>	<i>j8</i>	<i>j9</i>			
<i>i4</i>	<i>k1</i>	27		50	64	53	50	62	32	49	29	49		36	61	30	50	36	37	40	34	55		42	73	56	71	54	54	34	66	116	100	118
	<i>k2</i>	59	28		29	68	53	25	43	51	67	29	54		50	43	38	45	68	27	63	44	50		47	50	62	55	47	45	42	114	95	86
	<i>k3</i>	72	32	73		35	28	44	62	55	47	30	29	63		29	52	73	48	66	30	59	73	41		28	29	41	45	69	34	106	81	115
	<i>k4</i>	40	68	56	56		39	68	48	69	60	48	70	43	57		60	29	31	28	63	55	64	57	71		58	68	55	33	51	102	82	114
	<i>k5</i>	68	46	72	58	64		71	65	37	40	38	33	30	36	73		65	42	73	70	36	27	59	28	44		50	72	57	38	85	92	103
	<i>k6</i>	61	49	56	73	71	57		46	66	48	58	38	38	71	38	49		70	40	57	32	73	54	59	44	66		32	29	48	100	115	112
	<i>k7</i>	57	29	33	67	70	40	42		50	61	58	40	69	47	59	69	46		44	30	68	37	74	54	62	65	45		64	46	107	121	92
	<i>k8</i>	28	50	25	60	29	42	52	42		40	43	38	57	41	58	42	67	68		66	57	45	38	69	62	31	46	73		70	114	83	77
	<i>k9</i>	55	37	51	32	33	66	55	34	25		54	26	53	61	60	62	71	30	40		73	49	59	60	65	34	44	61	44		96	115	115
<i>i5</i>	<i>k1</i>	68		47	38	56	68	29	38	70	61	40		43	36	53	51	58	65	38	26	71		40	44	53	45	45	48	69	60	109	117	99
	<i>k2</i>	31	61		46	66	57	65	58	74	64	30	73		62	53	28	60	57	26	56	62	68		38	53	64	51	40	39	29	92	86	118
	<i>k3</i>	61	63	36		65	55	67	35	44	45	56	60	39		47	36	67	62	41	26	69	40	71		41	29	46	38	50	58	113	113	119
	<i>k4</i>	44	41	43	29		29	29	30	61	31	73	56	51	51		49	29	53	61	50	53	37	46	51		34	62	67	27	31	80	92	78
	<i>k5</i>	60	38	31	54	49		47	47	53	63	32	59	43	56	53		40	74	38	37	57	69	56	73	55		55	38	30	35	101	87	83
	<i>k6</i>	46	65	58	48	73	62		65	37	25	25	61	41	58	35	31		34	51	72	54	35	43	53	34	41		26	73	44	100	104	101
	<i>k7</i>	36	68	52	33	71	52	33		42	28	58	53	48	66	65	68	40		49	51	68	46	68	36	43	36	50		32	41	111	104	119
	<i>k8</i>	49	28	42	68	42	27	68	38		28	57	41	68	62	70	44	46	41		33	38	59	69	71	44	33	25	64		70	112	104	89
	<i>k9</i>	43	32	54	31	61	30	65	66	36		63	34	58	54	33	46	68	48	71		57	69	39	56	53	38	32	31	67		93	78	122

Anexo C. Guía para la validación de un modelo matemático en GAMS

La guía que se presenta a continuación está diseñada para servir como referente para la validación de modelos de programación lineal entera mixta en lenguaje GAMS. Se sugiere al estudiante seguir los pasos en el orden propuesto.

- Paso 1. Seleccione una instancia pequeña

En este paso se debe seleccionar una instancia pequeña, la cual permita validar el modelo, el tamaño de la instancia seleccionada debe permitirle escribir cada una de las restricciones de forma desagregada, a continuación se presentan los datos de la instancia seleccionada:

Tabla 43. Tiempos de procesamiento y alistamiento de la instancia k2m2i2

		<i>Sk_{mi}</i>						<i>Pk_{mi}</i>	
		<i>m1</i>			<i>m2</i>			<i>m1</i>	<i>m2</i>
		<i>j0</i>	<i>j1</i>	<i>j2</i>	<i>j0</i>	<i>j1</i>	<i>j2</i>		
<i>i1</i>	<i>k1</i>	69		36	25		72	115	76
	<i>k2</i>	60	58		56	25		104	110
<i>i2</i>	<i>k1</i>	49		32	32		69	75	104
	<i>k2</i>	49	33		47	52		97	91

- Paso 2. Escriba el modelo matemático

En este paso se debe escribir cada conjunto de restricciones de forma desagregada, para compararlo con las ecuaciones generadas en el software GAMS, a continuación se presentan cada uno de los bloques de ecuaciones para la instancia mencionada anteriormente, teniendo en cuenta los parámetros definidos en el capítulo 5.

$$\sum_m \sum_j Y_{k,j,m,i} = 1 \quad \forall k, i; \quad j \neq k; \quad k \neq 0$$

$$Y_{1011} + Y_{1021} + Y_{1211} + Y_{1221} = 1$$

$$Y_{1012} + Y_{1022} + Y_{1212} + Y_{1222} = 1$$

$$Y_{2011} + Y_{2021} + Y_{2111} + Y_{2121} = 1$$

$$Y_{2012} + Y_{2022} + Y_{2112} + Y_{2122} = 1$$

$$\sum_m \sum_j Y_{k,j,m,i} \leq 1 \quad \forall k, i; \quad j \neq k; \quad k \neq 0$$

$$Y_{0111} + Y_{0121} + Y_{2111} + Y_{2121} \leq 1$$

$$Y_{0112} + Y_{0122} + Y_{2112} + Y_{2122} \leq 1$$

$$Y_{0211} + Y_{0221} + Y_{1211} + Y_{1221} \leq 1$$

$$Y_{0212} + Y_{0222} + Y_{1212} + Y_{1222} \leq 1$$

$$\sum_h Y_{j,h,m,i} \geq Y_{k,j,m,i} \quad \forall k, i, m, j; \quad j \neq k; \quad j \neq h; \quad k \neq h; \quad k \neq 0$$

$$Y_{2011} \geq Y_{1211}$$

$$Y_{2021} \geq Y_{1221}$$

$$Y_{2022} \geq Y_{1222}$$

$$Y_{2012} \geq Y_{1212}$$

$$Y_{1011} \geq Y_{2111}$$

$$Y_{1011} \geq Y_{2111}$$

$$Y_{1021} \geq Y_{2121}$$

$$Y_{1012} \geq Y_{2112}$$

$$Y_{1022} \geq Y_{2122}$$

NOTA ACLARATORIA: como se puede ver en la ecuación escrita en su forma general, se está sumando la variable Y en el conjunto del subíndice h, el cual corresponde a los trabajos, dado que se definió que tanto j, h y k son diferentes, por este motivo en este ejemplo no hay sumas en cada una de las ecuaciones, ya que solo hay tres subíndices para los trabajos (0, 1, 2).

$$\sum_k Y_{k,j,m,i} \leq 1$$

$$\forall j, i, m; \quad j \neq k; \quad k \neq 0; j = 0$$

$$Y_{1011} + Y_{2011} \leq 1$$

$$Y_{1021} + Y_{2021} \leq 1$$

$$Y_{1012} + Y_{2012} \leq 1$$

$$Y_{1022} + Y_{2022} \leq 1$$

$$\sum_m (Y_{k,j,m,i} + Y_{j,k,m,i}) \leq 1$$

$$\forall k, j, i; \quad k \neq j; \quad k \neq 0$$

$$Y_{1011} + Y_{0111} + Y_{1021} + Y_{0121} \leq 1$$

$$Y_{1012} + Y_{0112} + Y_{1022} + Y_{0122} \leq 1$$

$$Y_{1211} + Y_{2111} + Y_{1221} + Y_{2121} \leq 1$$

$$Y_{1212} + Y_{2112} + Y_{1222} + Y_{2122} \leq 1$$

$$Y_{2011} + Y_{0211} + Y_{2021} + Y_{0221} \leq 1$$

$$Y_{2012} + Y_{0212} + Y_{2022} + Y_{0222} \leq 1$$

$$Y_{2111} + Y_{1211} + Y_{2121} + Y_{1221} \leq 1$$

$$Y_{2112} + Y_{1212} + Y_{2122} + Y_{1222} \leq 1$$

$$C_{k,i} = 0$$

$$\forall k, i; \quad k = 0$$

$$C_{01} = 0$$

$$C_{02} = 0$$

$$C_{k,i} + U(1 - Y_{k,j,m,i}) \geq C_{j,i} + Y_{k,j,m,i}(S_{k,j,m,i} + P_{k,m,i}) \quad \forall k, i, m, j; \quad j \neq k; \quad k \neq 0$$

$$C_{11} + 772 * (1 - Y_{1011}) \geq C_{01} + Y_{1011} * (69 + 115)$$

$$C_{11} + 772 * (1 - Y_{1021}) \geq C_{01} + Y_{1021} * (25 + 76)$$

$$C_{11} + 772 * (1 - Y_{1211}) \geq C_{21} + Y_{1211} * (36 + 115)$$

$$C_{11} + 772 * (1 - Y_{1221}) \geq C_{21} + Y_{1221} * (72 + 76)$$

$$C_{12} + 772 * (1 - Y_{1012}) \geq C_{02} + Y_{1012} * (75 + 49)$$

$$C_{12} + 772 * (1 - Y_{1022}) \geq C_{02} + Y_{1022} * (104 + 32)$$

$$C_{12} + 772 * (1 - Y_{1212}) \geq C_{22} + Y_{1212} * (32 + 75)$$

$$C_{12} + 772 * (1 - Y_{1222}) \geq C_{22} + Y_{1222} * (69 + 104)$$

$$C_{21} + 772 * (1 - Y_{2011}) \geq C_{01} + Y_{2011} * (60 + 104)$$

$$C_{21} + 772 * (1 - Y_{2021}) \geq C_{01} + Y_{2021} * (56 + 110)$$

$$C_{21} + 772 * (1 - Y_{2111}) \geq C_{11} + Y_{2111} * (58 + 104)$$

$$C_{21} + 772 * (1 - Y_{2121}) \geq C_{11} + Y_{2121} * (25 + 110)$$

$$C_{22} + 772 * (1 - Y_{2012}) \geq C_{02} + Y_{2012} * (49 + 97)$$

$$C_{22} + 772 * (1 - Y_{2022}) \geq C_{02} + Y_{2022} * (47 + 91)$$

$$C_{22} + 772 * (1 - Y_{2112}) \geq C_{12} + Y_{2112} * (33 + 97)$$

$$C_{22} + 772 * (1 - Y_{2122}) \geq C_{12} + Y_{2122} * (52 + 91)$$

$$C_{k,i} - C_{k,i-1} \geq Y_{k,j,m,i}(S_{k,j,m,i} + P_{k,m,i}) \quad \forall k, i, m, j; j \neq k; k \neq 0; i \neq 1$$

$$C_{12} - C_{11} \geq Y_{1012} * (49 + 75)$$

$$C_{12} - C_{11} \geq Y_{1022} * (32 + 104)$$

$$C_{12} - C_{11} \geq Y_{1212} * (32 + 75)$$

$$C_{12} - C_{11} \geq Y_{1222} * (69 + 104)$$

$$C_{22} - C_{21} \geq Y_{2012} * (49 + 97)$$

$$C_{22} - C_{21} \geq Y_{2022} * (47 + 91)$$

$$C_{22} - C_{21} \geq Y_{2112} * (33 + 97)$$

$$C_{22} - C_{21} \geq Y_{2122} * (52 + 91)$$

$$C_{max} \geq C_{k,i}$$

$$\forall k,i \quad i = \text{última estación}$$

$$C_{max} \geq C_{02}$$

$$C_{max} \geq C_{12}$$

$$C_{max} \geq C_{22}$$

- Paso 3. Habilite el diccionario de ecuaciones en GAMS

Una vez escrito el modelo matemático en lenguaje GAMS, habilite el diccionario de ecuaciones de GAMS, el cual le permitirá comparar las ecuaciones generadas por el software con las escritas en el paso anterior, para ello realice el siguiente procedimiento:

1. Habilite la opción del diccionario: cuando se haya definido que ecuaciones se van a tener en cuenta en el modelo, escriba: option mip=convert; como se muestra a continuación

Figura 12. Habilitación de diccionario en GAMS

```
e0..makespan=e*cmak;
e1(k,i)$(ord(k)>1)..sum(m,j)$(not sameas(j,k)),y(k,j,m,i)=E=1;
e2(k,i)$(ord(k)>1)..sum(j,m)$(not sameas(j,k)),y(j,k,m,i)=1=1;
e3(k,i,m,j)$(not sameas(j,k)and(ord(j)>1)and(ord(k)>1))..sum(h)$(not sameas(j,h)and(not sameas(h,k))),y(j,h,m,i)=q=y(k,j,m,i);
e4(m,i,j)$(ord(j)=1)..sum(k)$(ord(k)>1),y(k,j,m,i)=1=1;
e5(k,j,i)$(not sameas(j,k)and(ord(k)>1))..sum(m),y(k,j,m,i)+y(j,k,m,i)=1=1;
e6(i,m)..c('0',i)=e=0;
e7(k,m,j,i)$(not sameas(j,k)and(ord(k)>1))..c(k,i)+u*(1-y(k,j,m,i))=q=c(j,i)+y(k,j,m,i)*m(k,j,m,i)+p(k,m,i)*y(k,j,m,i);
e8(k,m,j,i)$(not sameas(j,k)and(ord(k)>1)and(ord(i)>1))..c(k,i)-c(k,i-1)=q=y(k,j,m,i)*[m(k,j,m,i)+p(k,m,i)];
e9(i,m,k)$(ord(i)>1)..cmak=q=c(k,i);

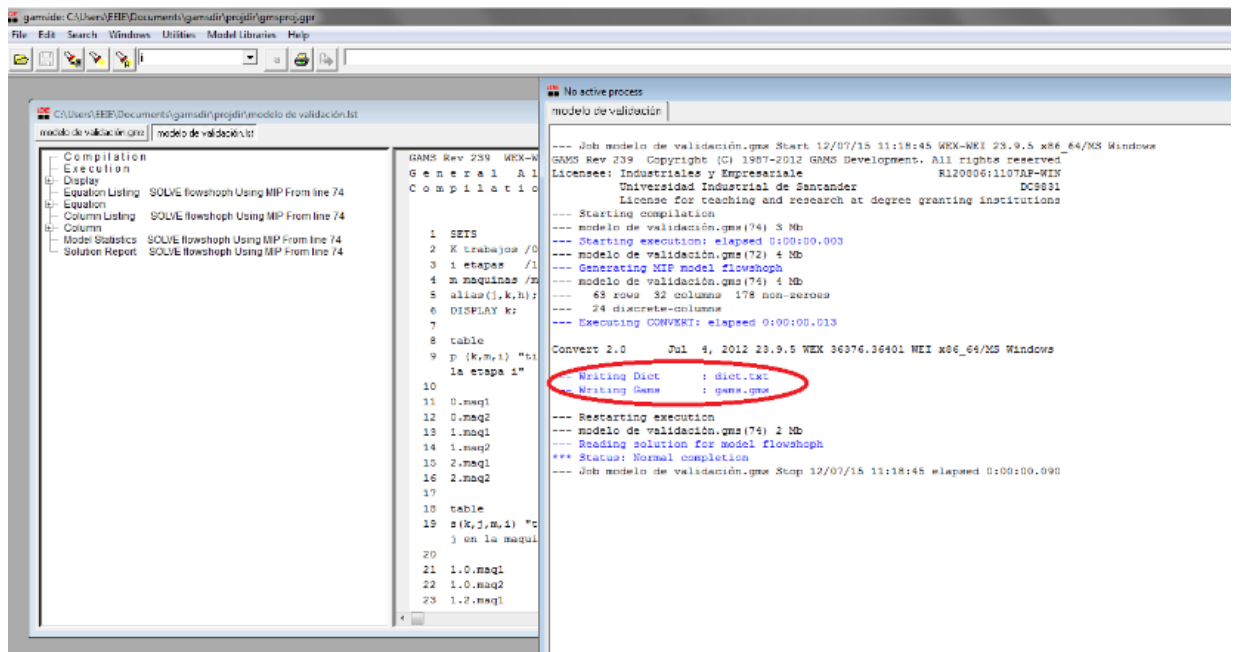
MODEL flowshoph /e0,e1,e2,e3,e4,e5,e6,e7,e8,e9/

option mip=convert;

solve flowshoph using mip minimizing makespan;
```

2. Una vez habilitada la opción mencionada, ejecute el modelo; aparecerá la siguiente ventana:

Figura 13. Creación de archivos gams.gms y dict.txt



Como se puede observar en la figura anterior, se crearon dos archivos: uno llamado gams.gms el cual contiene las ecuaciones generadas por el programa, y

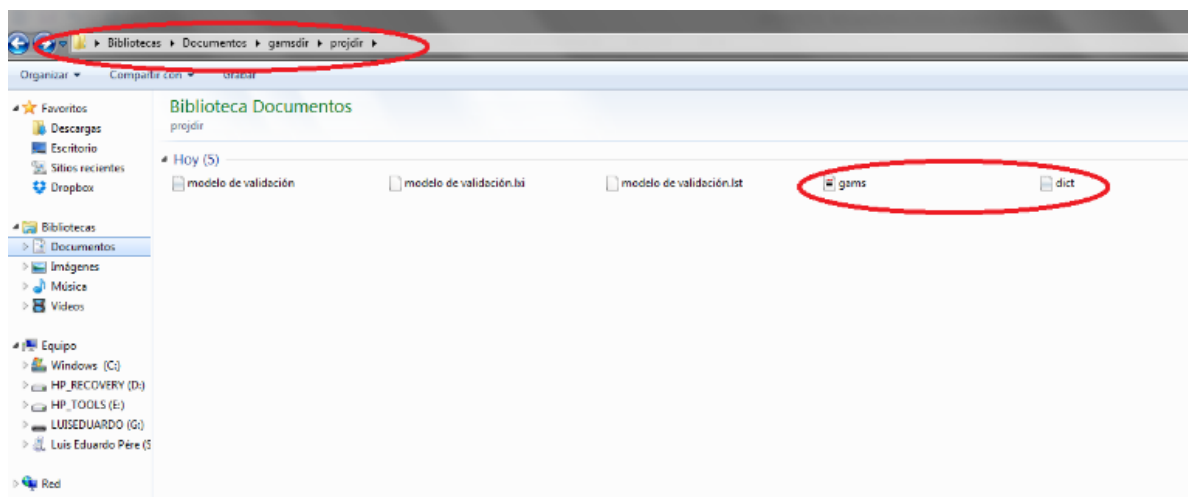
otro archivo llamado dict.txt el cual contiene el nombre de cada una de las variables, por defecto los dos archivos se guardan en la carpeta projdir.

- Paso 4. Reescriba las ecuaciones generadas en GAMS

Abra los archivos gams.gms y dict.txt. Para abrir el archivo .gms de doble clic en el nombre y para el archivo .txt siga la siguiente ruta:

Bibliotecas > Documentos > gamsdir > projdir

Figura 14. Ruta del archivo dict.txt



Como se puede observar en la figura 16, el archivo gams.gms muestra cada una de las variables y las ecuaciones generadas para la instancia de prueba, las cuales corresponden a 63, para poder validar el modelo, remítase al documento dict.txt donde puede encontrar el valor de cada una de las variables, tal como se muestra en las figura 17.

Figura 15. Archivo gams.gms

```

gamside: C:\Users\EEIE\Documents\gamsdir\projdir\gmsproj.gpr - [C:\Users\EEIE\Documents\gamsdir\projdir\gams.gms]
IDE File Edit Search Windows Utilities Model Libraries Help
[Icons] [i] [a] [Print] [Run]
modelo de validación.gms | gams.gms | modelo de validación.lst
+      32      8      24      0      0      0      0      0
+ FX      0      0      0      0      0      0      0      0
+
+ Nonzero counts
+ Total const NL DLL
+ 178 178 0 0
+
+ Solve m using MIP minimizing x32;

Variables b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17,b18,b19
,b20,b21,b22,b23,b24,x25,x26,x27,x28,x29,x30,x31,x32;

Positive Variables x25,x26,x27,x28,x29,x30;

Binary Variables b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17
,b18,b19,b20,b21,b22,b23,b24;

Equations e1,e2,e3,e4,e5,e6,e7,e8,e9,e10,e11,e12,e13,e14,e15,e16,e17,e18,e19
,e20,e21,e22,e23,e24,e25,e26,e27,e28,e29,e30,e31,e32,e33,e34,e35,e36
,e37,e38,e39,e40,e41,e42,e43,e44,e45,e46,e47,e48,e49,e50,e51,e52,e53
,e54,e55,e56,e57,e58,e59,e60,e61,e62,e63;

e1.. - x31 + x32 =E= 0;

e2.. b9 + b11 + b13 + b15 =E= 1;

e3.. b10 + b12 + b14 + b16 =E= 1;

e4.. b17 + b19 + b21 + b23 =E= 1;

e5.. b18 + b20 + b22 + b24 =E= 1;

e6.. b1 + b3 + b21 + b23 =L= 1;

e7.. b2 + b4 + b22 + b24 =L= 1;

e8.. b5 + b7 + b13 + b15 =L= 1;

e9.. b6 + b8 + b14 + b16 =L= 1;

e10.. - b13 + b17 =G= 0;

e11.. - b15 + b19 =G= 0;

e12.. - b14 + b18 =G= 0;

e13.. - b16 + b20 =G= 0;

e14.. b9 - b21 =G= 0;

```

14: 66 Insert

Figura 16. Archivo dict.txt

dict: Bloc de notas									dict: Bloc de notas				
Archivo	Edición	Formato	Ver	Ayuda	Archivo	Edición	Formato	Ver	Ayuda				
MIP written by GAMS Convert at 12/07/15 11:18:45													
Equation counts													
Total	E	G	L	N	X	C	B						
63	9	38	16	0	0	0	0						
Variable counts													
Total	x	b	i	s1s	s2s	sc	s1						
32	8	24	0	0	0	0	0						
FX	0	0	0	0	0	0	0						
Nonzero counts													
Total	const	NL	DLL										
178	178	0	0										
Equations 1 to 63													
e1	e0												
e2	e1(1,1)												
e3	e1(1,2)												
e4	e1(2,1)												
e5	e1(2,2)												
e6	e2(1,1)												
e7	e2(1,2)												
e8	e2(2,1)												
e9	e2(2,2)												
e10	e3(1,1,maq1,2)												
e11	e3(1,1,maq2,2)												
e12	e3(1,2,maq1,2)												
e13	e3(1,2,maq2,2)												
e14	e3(2,1,maq1,1)												
e15	e3(2,1,maq2,1)												
e16	e3(2,2,maq1,1)												
e17	e3(2,2,maq2,1)												
e18	e4(maq1,1,0)												
e19	e4(maq1,2,0)												
e20	e4(maq2,1,0)												
e21	e4(maq2,2,0)												
e22	e5(1,0,1)												
e23	e5(1,0,2)												
e24	e5(1,2,1)												
e25	e5(1,2,2)												
e26	e5(2,0,1)												
e27	e5(2,0,2)												
e28	e5(2,1,1)												
e29	e5(2,1,2)												
e30	e6(1,maq1)												
e31	e6(1,maq2)												
e32	e6(2,maq1)												
e33	e6(2,maq2)												
e34	e7(1,maq1,0,1)												
e35	e7(1,maq1,0,2)												
e36	e7(1,maq1,2,1)												
e37	e7(1,maq1,2,2)												
e38	e7(1,maq2,0,1)												
e39	e7(1,maq2,0,2)												
e40	e7(1,maq2,2,1)												
e41	e7(1,maq2,2,2)												
e42	e7(2,maq1,0,1)												
e43	e7(2,maq1,0,2)												
e44	e7(2,maq1,1,1)												
e45	e7(2,maq1,1,2)												
e46	e7(2,maq2,0,1)												
e47	e7(2,maq2,0,2)												
e48	e7(2,maq2,1,1)												
e49	e7(2,maq2,1,2)												
e50	e8(1,maq1,0,2)												
e51	e8(1,maq1,2,2)												
e52	e8(1,maq2,0,2)												
e53	e8(1,maq2,2,2)												
e54	e8(2,maq1,0,2)												
e55	e8(2,maq1,1,2)												
e56	e8(2,maq2,0,2)												
e57	e8(2,maq2,1,2)												
e58	e9(2,maq1,0)												
e59	e9(2,maq1,1)												
e60	e9(2,maq1,2)												
e61	e9(2,maq2,0)												
e62	e9(2,maq2,1)												
e63	e9(2,maq2,2)												
Variables 1 to 32													
b1	y(0,1,maq1,1)												
b2	y(0,1,maq1,2)												
b3	y(0,1,maq2,1)												
b4	y(0,1,maq2,2)												
b5	y(0,2,maq1,1)												
b6	y(0,2,maq1,2)												
b7	y(0,2,maq2,1)												
b8	y(0,2,maq2,2)												
b9	y(1,0,maq1,1)												
b10	y(1,0,maq1,2)												
b11	y(1,0,maq2,1)												
b12	y(1,0,maq2,2)												
b13	y(1,2,maq1,1)												
b14	y(1,2,maq1,2)												
b15	y(1,2,maq2,1)												
b16	y(1,2,maq2,2)												
b17	y(2,0,maq1,1)												
b18	y(2,0,maq1,2)												
b19	y(2,0,maq2,1)												
b20	y(2,0,maq2,2)												
b21	y(2,1,maq1,1)												
b22	y(2,1,maq1,2)												
b23	y(2,1,maq2,1)												
b24	y(2,1,maq2,2)												
x25	c(0,1)												
x26	c(0,2)												
x27	c(1,1)												
x28	c(1,2)												
x29	c(2,1)												
x30	c(2,2)												
x31	cmax												
x32	makespan												

Reescriba las ecuaciones, para de esta forma poder compararlas. A continuación se muestran las ecuaciones reescritas para la instancia evaluada en esta guía:

$$e1.. \text{makespan} = C_{max}$$

$$e2.. Y_{1011} + Y_{1021} + Y_{1211} + Y_{1221} = 1$$

$$e3.. Y_{1012} + Y_{1022} + Y_{1212} + Y_{1222} = 1$$

$$e4.. Y_{2011} + Y_{2021} + Y_{2111} + Y_{2121} = 1$$

$$e5.. Y_{2012} + Y_{2022} + Y_{2112} + Y_{2122} = 1$$

$$e6.. Y_{0111} + Y_{0121} + Y_{2111} + Y_{2121} \leq 1$$

$$e7.. Y_{0112} + Y_{0122} + Y_{2112} + Y_{2122} \leq 1$$

$$e8.. Y_{0211} + Y_{0221} + Y_{1211} + Y_{1221} \leq 1$$

$$e9.. Y_{0212} + Y_{0222} + Y_{1212} + Y_{1222} \leq 1$$

$$e10.. Y_{2011} \geq Y_{1211}$$

$$e11.. Y_{2021} \geq Y_{1221}$$

$$e12.. Y_{2012} \geq Y_{1212}$$

$$e13.. Y_{2022} \geq Y_{1222}$$

$$e14.. Y_{1011} \geq Y_{2111}$$

$$e15.. Y_{1021} \geq Y_{2121}$$

$$e16.. Y_{1012} \geq Y_{2112}$$

$$e17.. Y_{1022} \geq Y_{2122}$$

$$e18..Y_{1011} + Y_{2011} \leq 1$$

$$e19..Y_{1012} + Y_{2012} \leq 1$$

$$e20..Y_{1021} + Y_{2021} \leq 1$$

$$e21..Y_{1022} + Y_{2022} \leq 1$$

$$e22..Y_{0111} + Y_{0121} + Y_{1011} + Y_{1021} \leq 1$$

$$e23..Y_{0112} + Y_{0122} + Y_{1012} + Y_{1022} \leq 1$$

$$e24..Y_{1211} + Y_{1221} + Y_{2111} + Y_{2121} \leq 1$$

$$e25..Y_{1212} + Y_{1222} + Y_{2112} + Y_{2122} \leq 1$$

$$e26..Y_{0211} + Y_{0221} + Y_{2011} + Y_{2021} \leq 1$$

$$e27..Y_{0212} + Y_{0222} + Y_{2012} + Y_{2022} \leq 1$$

$$e28..Y_{1211} + Y_{1221} + Y_{2111} + Y_{2121} \leq 1$$

$$e29..Y_{1212} + Y_{1222} + Y_{2112} + Y_{2122} \leq 1$$

$$e30..C_{01} = 0$$

$$e31..C_{01} = 0$$

$$e32..C_{02} = 0$$

$$e33..C_{02} = 0$$

$$e34..-956 * Y_{1011} - C_{01} + C_{11} = -772$$

$$e35..-896 * Y_{1012} - C_{02} + C_{12} = -772$$

$$e36..-923 * Y_{1211} - C_{21} + C_{11} = -772$$

$$e37..-879 * Y_{1212} + C_{12} - C_{22} = -772$$

$$e38..-873 * Y_{1021} + C_{11} - C_{01} = -772$$

$$e39..-908 * Y_{1022} - C_{02} + C_{12} = -772$$

$$e40..-920 * Y_{1221} + C_{11} - C_{21} = -772$$

$$e41..-945 * Y_{1222} + C_{12} - C_{22} = -772$$

$$e42..-936 * Y_{2011} + C_{21} - C_{01} = -772$$

$$e43..-918 * Y_{2012} - C_{02} + C_{22} = -772$$

$$e44..-934 * Y_{2111} - C_{11} + C_{21} = -772$$

$$e45..-902 * Y_{2112} - C_{12} + C_{22} = -772$$

$$e46..-938 * Y_{2021} - C_{01} + C_{21} = -772$$

$$e47..-910 * Y_{2022} - C_{02} + C_{22} = -772$$

$$e48..-907 * Y_{2121} - C_{11} + C_{21} = -772$$

$$e49..-915 * Y_{2122} - C_{12} + C_{22} = -772$$

$$e50..-124 * Y_{1012} - C_{11} + C_{12} = 0$$

$$e51..-107 * Y_{1212} - C_{11} + C_{12} = 0$$

$$e52..-136 * Y_{1022} - C_{11} + C_{12} = 0$$

$$e53..-173 * Y_{1222} - C_{11} + C_{12} = 0$$

$$e54.. -146 * Y_{2012} - C_{21} + C_{22} = 0$$

$$e55.. -130 * Y_{2112} - C_{21} + C_{22} = 0$$

$$e56.. -138 * Y_{2022} - C_{21} + C_{22} = 0$$

$$e57.. -143 * Y_{2122} - C_{21} + C_{22} = 0$$

$$e58.. -C_{02} + C_{max} = 0$$

$$e59.. -C_{12} + C_{max} = 0$$

$$e60.. -C_{22} + C_{max} = 0$$

$$e61.. -C_{02} + C_{max} = 0$$

$$e62.. -C_{12} + C_{max} = 0$$

$$e63.. -C_{22} + C_{max} = 0$$

- Paso 5: Verifique el resultado

Una vez las ecuaciones generadas por el software GAMS concuerden con las ecuaciones desagregadas, se comprueba que el resultado obtenido sea correcto, para ello se sugiere revisar en primer lugar la asignación de los trabajos en las máquinas, la cual la puede consultar en el documento .lst, en la sección SOLVAR, como se muestra en las figuras 18 y 19.

Figura 17. Asignación de trabajos para la instancia k2m2i2

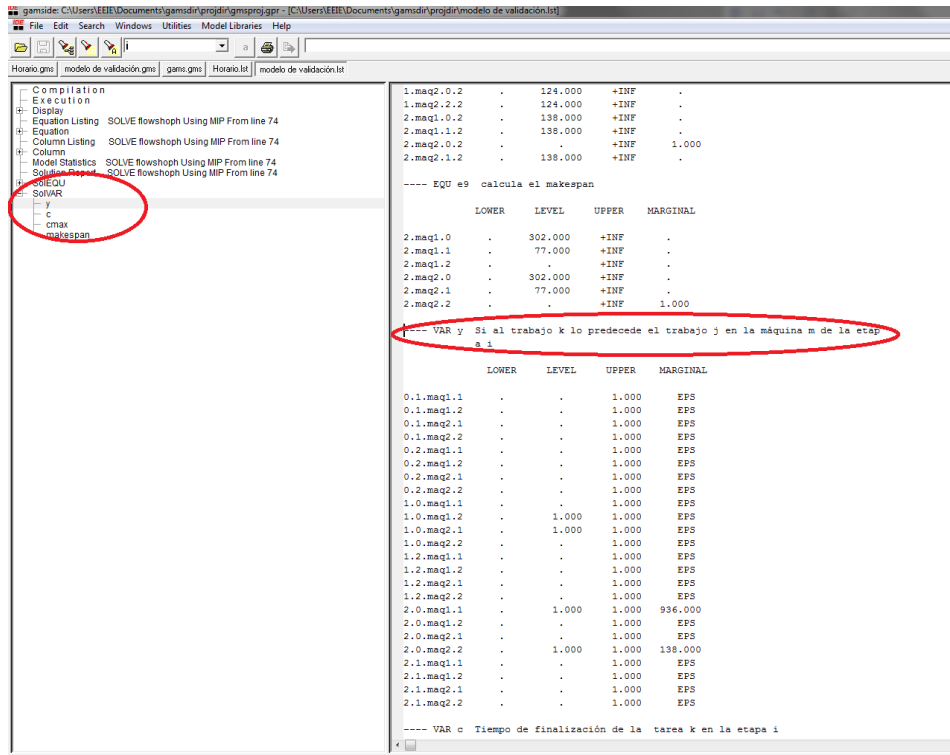
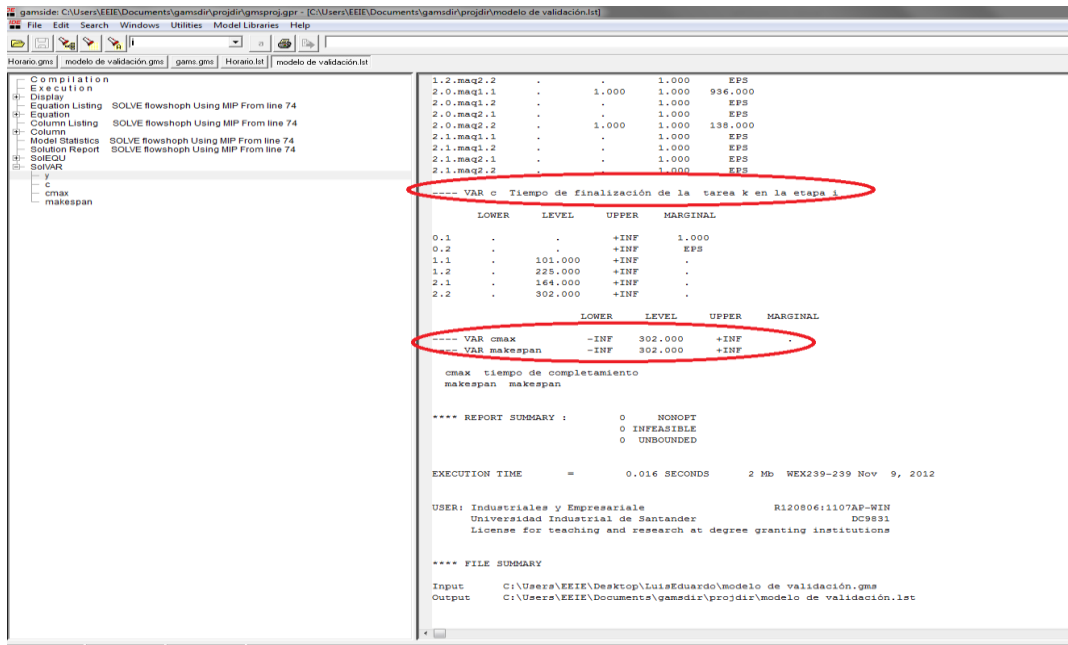
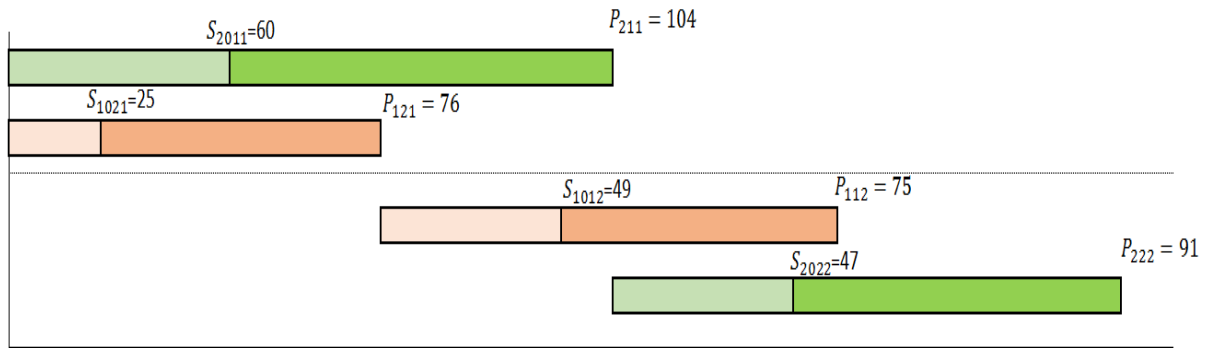


Figura 18. Tiempos de finalización y makespan de la instancia k2m2i2



Para su validación se realizó el diagrama de Gantt que se muestra a continuación:

Figura 19. Diagrama de Gantt para instancia k2m2i2



Anexo D. Códigos de programación del algoritmo competitivo imperialista

A continuación se presentan los códigos de programación del algoritmo competitivo imperialista, junto con las diferentes funciones utilizadas en el lenguaje de programación de Matlab, el cual fue aplicado a la solución del problema de *flow shop* híbrido con máquinas paralelas no relacionadas.

- **Función principal**

```
function [cos_bueno , solucion, Iter]=imperio(iteraciones_luego,
criterio, cantMaqu, cantPais, cantTrab, cantEtap, porcen_imperialistas,
par_chi , par_compartir, par_revolucion, costos)
Iter=0;

% IMPERIO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%**PASO 1 ** %%%%%%%%%

[costos_sepa]=separa_costos(costos,cantEtap, cantTrab,cantMaqu);
clear costos
%% generacion de paises
[Paises]=genera_paises(cantPais,cantMaqu, cantTrab, cantEtap);

%% calcula el makespan de cada pais
makespan =
calcula_makespan(Paises,cantPais,cantMaqu,cantTrab,cantEtap,costos_sepa);
%% identificacion de los imperialistas
[val, pos_pa]=sort(makespan);
cantImper=round(porcen_imperialistas*cantPais);

%% calcula costos y poder normalizados
[pod_nor] = costo_normalizado(cantImper,val);
clear val
%% calcula cantidad de colonias
```

```

cant_Colo=cantidad_colonias(cantImper,cantPais,pod_nor);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% **PASO 2 **%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% asignación de colonias
Imper= asignacion_colonias(cantImper,pos_pa,cant_Colo);
%% asimilacion
posi_imperialistas=pos_pa(1:cantImper);
while (size(Imper,1)>1)
Iter=Iter+1;
[Paises, makespan]= asimilacion(cantImper, cantMaqu, cantTrab,
cantEtap,posi_imperialistas,cant_Colo,Imper,Paises, costos_sepa,makespan);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PASO 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Compartir informacion entre colonias
[makespan , Paises]=compartir_colonias(costos_sepa, makespan, Paises,
par_compartir, cantImper, cantEtap, cantMaqu, cantTrab, cant_Colo,
Imper);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PASO 4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% revolucion
[makespan , Paises]=revolucion(costos_sepa, cantEtap, makespan, Paises,
cantImper, cant_Colo, par_revolucion, Imper,cantMaqu, cantTrab);

%% recalculo imperialista
posi_imperialistas=zeros(cantImper,1);
for k=1:cantImper
    [val , pos]=min(makespan(Imper{k}));
    aux_temp=Imper{k}(1,1);

```

```

    Imper{k}(1,1)=Imper{k}(pos,1);
    posi_imperialistas(k,1)=Imper{k}(1,1);
    Imper{k}(pos,1)=aux_temp;
end
clear aux_temp
clear val
%% recalcuulo costo imperio
costo_Imper=zeros(cantImper,1);
for k=1:cantImper

costo_Imper(k,1)=makespan(Imper{k}(1,1))+par_chi*mean(makespan(Imper{k}(2
:end)));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PASO 5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% guerra entre imperios
[Imper, cant_Colo, cantImper] = guerra_imperios(costo_Imper, Imper,
cant_Colo);
posi_imperialistas=zeros(cantImper,1);
for k=1:cantImper
    posi_imperialistas(k,1)=Imper{k}(1,1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
clear costo_Imper
clear k

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SEGUNDA PARTE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[cos_bueno , pais_buen]=min(makespan);
solucion=Paises(pais_buen,:);
switch (criterio)
    case 'Iteraciones'
        cont_iter=0;
        while (cont_iter <= iteraciones_luego)
            cont_iter=cont_iter+1;

```

```

        % asimilacion
        [Paises, makespan]= asimilacion(cantImper, cantMaqu,
cantTrab,
cantEtap, posi_imperialistas, cant_Colo, Imper, Paises, costos_sepa, makespan);
        % Compartir informacion entre colonias
        [makespan , Paises]=compartir_colonias(costos_sepa, makespan,
Paises, par_compartir, cantImper, cantEtap, cantMaqu, cantTrab,
cant_Colo, Imper);
        % revolucion
        [makespan , Paises]=revolucion(costos_sepa, cantEtap,
makespan, Paises, cantImper, cant_Colo, par_revolucion, Imper, cantMaqu,
cantTrab);

        % recalcuulo imperialista
        posi_imperialistas=zeros(cantImper,1);
        for k=1:cantImper
            [val , pos]=min(makespan(Imper{k}));
            aux_temp=Imper{k}(1,1);
            Imper{k}(1,1)=Imper{k}(pos,1);
            posi_imperialistas(k,1)=Imper{k}(1,1);
            Imper{k}(pos,1)=aux_temp;
        end

        clear val
        clear aux_temp
    end
case 'Default'
    cambio=1;
    while(cambio==1)
        % asimilacion
        [Paises, makespan]= asimilacion(cantImper, cantMaqu,
cantTrab,
cantEtap, posi_imperialistas, cant_Colo, Imper, Paises, costos_sepa, makespan);
        % Compartir informacion entre colonias
        [makespan , Paises]=compartir_colonias(costos_sepa, makespan,
Paises, par_compartir, cantImper, cantEtap, cantMaqu, cantTrab,
cant_Colo, Imper);

```

```

        % revolucion
        [makespan , Paises]=revolucion(costos_sepa, cantEtap,
makespan, Paises, cantImper, cant_Colo, par_revolucion, Imper,cantMaqu,
cantTrab);

        % recalcuulo imperialista
        posi_imperialistas=zeros(cantImper,1);
        for k=1:cantImper
            [val , pos]=min(makespan(Imper{k}));
            if(pos==1)
                cambio=0;
            end
            aux_temp=Imper{k}(1,1);
            Imper{k}(1,1)=Imper{k}(pos,1);
            posi_imperialistas(k,1)=Imper{k}(1,1);
            Imper{k}(pos,1)=aux_temp;
        end
        clear val
        clear aux_temp
    end

end

    cos_bueno=makespan(Imper{1}(1,1));
    solucion=Paises(Imper{1}(1,1),:);

solucion(2,:)=makespan_individual(1,Paises(Imper{1}(1,1),:),cantMaqu,cant
Trab,cantEtap,costos_sepa);
end

```

- **Funciones de apoyo**

- **Función costos_sepa**

```

function [costos_sepa]=separa_costos(costos,cantEtap, cantTrab,cantMaqu)
costos_sepa=zeros(cantTrab^2+cantTrab,cantEtap,cantMaqu);
for k=1:cantMaqu

```

```

        costos_sepa(:, :, k) = costos(k:cantMaqu:end, :);
end
end

```

- Función países

Permite crear la matriz de individuos que conformarán la población inicial.

```

function [Paises]=genera_paises(cantPais,cantMaqu, cantTrab, cantEtap)
Paises=zeros(cantPais,((cantMaqu+cantTrab)*cantEtap));
for s=1:cantPais
for k=1:cantEtap
Paises(s, (k1)*(cantMaqu+cantTrab)+2:(k)*(cantMaqu+cantTrab))=randperm(cantMaqu+cantTrab-1);
    % asignacion de trabajos en las maquinas
end
end
clear P
clear pos
end

```

- Función Makespan

Permite calcular el makespan.

```

function[makespan]=makespan_individual(k,Paises,cantMaqu,cantTrab,cantEtap,costos_sepa)
sumador = 0;
cont_etapa=0;
cont_maquina=1;
for s=1:cantEtap*(cantTrab+cantMaqu);
    opti=Paises(k,s);
    switch(opti)
        case 0 %% etapa nueva

```

```

        cont_etapa=cont_etapa+1;
        cont_maquina=1;
        sumador = 0;
        case num2cell(cantTrab+1:cantTrab+cantMaqu-1) %% hubo cambio
de maquina
            Paises(k,s)=0;
            cont_maquina=cont_maquina+1;
            sumador=0;
            case num2cell(1:cantTrab)
                if(cont_etapa==1)
                    sumador=sumador+costos_sepa(1+Paises(k,s-
1)+(cantTrab+1)*(Paises(k,s)-1),cont_etapa,cont_maquina);
                else
                    posi=(cont_etapa-
2)*(cantMaqu+cantTrab)+find(Paises(k,(cont_etapa-
2)*(cantMaqu+cantTrab)+1:(cont_etapa-1)*(cantMaqu+cantTrab))==opti);
                    ele=max(sumador, makespan(1,posi));
                    sumador=ele+costos_sepa(1+Paises(k,s-
1)+(cantTrab+1)*(Paises(k,s)-1),cont_etapa,cont_maquina);
                end
            end
            makespan(1,s)=sumador;
            % fprintf('valor %d etapa %d maquina %d sumador %d \n
',opti,cont_etapa, cont_maquina, sumador)
        end
        clear sumador
clear posi
clear ele
clear cont_maquina
end

function[makespan1]=calcula_makespan(Paises,cantPais,cantMaqu,cantTrab,ca
ntEtap,costos_sepa)

```

```

makespan1 = zeros(cantPais,1);
for k=1:cantPais

makespan1(k,1)=max(makespan_individual(k,Paises,cantMaqu,cantTrab,cantEta
p,costos_sepa));
end
end

```

- Funcion costo normalizado

```

function [pod_nor]=costo_normalizado(cantImper,val)
cos_nor=zeros(cantImper,1);
cos_nor(1:cantImper,1) = val(end,1)-val(1:cantImper,1);
pod_nor = cos_nor/ sum(cos_nor);
end

```

- Función cantidad de colonias

```

function [cant_Colo]=cantidad_colonias(cantImper,cantPais,pod_nor)
cant_Colo=ones(cantImper,1);
sum_aux=sum(cant_Colo);
sum_cant_Colo=sum(cant_Colo);
k=1;
while(sum_cant_Colo <= (cantPais-cantImper) && k <= cantImper)
    cant_Colo(k)=cant_Colo(k)+ min(round((cantPais-cantImper-
sum_aux)*pod_nor(k,1)), (cantPais-cantImper-sum_cant_Colo));
    sum_cant_Colo=sum_cant_Colo+cant_Colo(k)-1;
    k=k+1;
end
clear sum_aux
clear sum_cant_Colo
end

```

- Función asignación de colonias

```
function [Imper]= asignacion_colonias(cantImper,pos_pa,cant_Colo)
Imper=cell(cantImper,1);
asig_aux=pos_pa(cantImper+1:end,1);
a_aux=randperm(length(asig_aux));
asig_aux=asig_aux(a_aux);
Imper=cell(cantImper,1);
ind_aux=1;
for k=1:cantImper
    Imper{k}=[pos_pa(k,1) ; asig_aux(ind_aux:ind_aux+cant_Colo(k,1)-1)];
    ind_aux=ind_aux+cant_Colo(k);
end
clear asig_aux
clear a_aux
end
```

- Función de asimilación de colonias

```
function [Paises , makespan] = asimilacion(cantImper, cantMaqu, cantTrab,
cantEtap,pos_pa,cant_Colo,Imper,Paises,costos_sepa,makespan)
mascaras_binarias=round(rand(cantImper,((cantMaqu+cantTrab)*cantEtap)));
copias=Paises(pos_pa(1:cantImper),:).*mascaras_binarias;
for k=1:cantImper
    colo_asimi=(copias(k,:)'*ones(1,cant_Colo(k,1)))';
    pos=find(copias(k,)==0);
    mis_paises=Paises(Imper{k},:);
    for h=1:cant_Colo(k,1)
        for et=1:cantEtap
            mi_pos=pos(pos<=et*(cantMaqu+cantTrab)); % posiciones que hace
falta completar
            mi_pos=mi_pos(mi_pos >(et-1)*(cantMaqu+cantTrab));
            mi_pais=mis_paises(h+1,:);
```

```

        tengo= find(colo_asimi(h, ((et-
1) * (cantMaqu+cantTrab)+1:(et) * (cantMaqu+cantTrab)))>0);
        tengo1=colo_asimi(h,tengo+(et-1) * (cantMaqu+cantTrab)); %% los
valores que ya están guardados
        ind_mis_pos=2;
        ss=2;
        while (ss<=cantMaqu+cantTrab && ind_mis_pos <=size(mi_pos,2))
            if(find(tengo1==mi_pais(ss+(et-1) * (cantMaqu+cantTrab))))
                ss=ss+1;
            else
                colo_asimi(h,mi_pos(ind_mis_pos))=mi_pais(ss +(et-
1) * (cantMaqu+cantTrab));
                ind_mis_pos=ind_mis_pos+1;
                ss=ss+1;
            end
        end
        end
        end
        end
        Países(Imper{k}(2:end,1),:)=colo_asimi;
        % actualiza costos de asimilados
        for m=2:cant_Colo(k)+1

makespan(Imper{k}(m,1),1)=max(makespan_individual(Imper{k}(m,1),Países,ca
ntMaqu,cantTrab,cantEtap,costos_sepa));
        end
    end
clear pos ; clear mis_paises ; clear mi_pos ; clear mi_pais; clear
ind_mis_pos;
clear ss ; clear colo_asimi ; clear mascarar_binarias ; clear copias ;
end

```

- Función Compartir información entre colonias.

```

function [makespan , Paises]=compartir_colonias(costos_sepa, makespan,
Paises, par_compartir, cantImper, cantEtap, cantMaqu, cantTrab,
cant_Colo, Imper)
for k=1:cantImper
    if(cant_Colo(k,1)>=2) % verifica que haya al menos dos colonias para
compartir información
        cant_comp=max(2,2*round(cant_Colo(k,1)*par_compartir/2));
        for m=1:cant_comp/2
            pos=randperm(size(Imper{k},1)-1)+1;
            pos_col1=pos(1,1);
            pos_col2=pos(1,2);
            colo_aux1=zeros(size(Paises(1,:)));
            colo_aux2=zeros(size(Paises(1,:)));
            for s=1:cantEtap
                % elige el punto para compartir
                pun_com=round((cantMaqu+cantTrab-1)*rand(1,1)+1)+(s-
1)*(cantMaqu+cantTrab);
                colo_aux1((s-
1)*(cantMaqu+cantTrab)+1:pun_com)=Paises(Imper{k}(pos_col1,1),(s-
1)*(cantMaqu+cantTrab)+1:pun_com);
                colo_aux2((s-
1)*(cantMaqu+cantTrab)+1:pun_com)=Paises(Imper{k}(pos_col2,1),(s-
1)*(cantMaqu+cantTrab)+1:pun_com);
                ind_pos=2+(s-1)*(cantMaqu+cantTrab);
                ind_col=pun_com+1;
                %COLONIA 1
                while (ind_col<=s*(cantMaqu+cantTrab))
                    if(find(colo_aux1((s-
1)*(cantMaqu+cantTrab)+1:pun_com)==Paises(Imper{k}(pos_col2,1),ind_pos)))
                        ind_pos=ind_pos+1;
                    else
                        colo_aux1(1,ind_col)=Paises(Imper{k}(pos_col2,1),ind_pos);
                        ind_pos=ind_pos+1;
                    end
                end
            end
        end
    end
end

```

```

        ind_col=ind_col+1;
    end
end
%COLONIA 2
ind_pos=2+(s-1)*(cantMaqu+cantTrab);
ind_col=pun_com+1;
while (ind_col<=s*(cantMaqu+cantTrab))
    if(find(colo_aux2((s-
1)*(cantMaqu+cantTrab)+1:pun_com))==Paises(Imper{k}(pos_coll,1),ind_pos))
        ind_pos=ind_pos+1;
    else
        colo_aux2(1,ind_col)=Paises(Imper{k}(pos_coll,1),ind_pos);
        ind_pos=ind_pos+1;
        ind_col=ind_col+1;
    end
end
end
end
% calcula makespan y elige los dos menores
aux_cos=[ makespan(Imper{k}(pos_coll,1)) ;
makespan(Imper{k}(pos_col2,1)) ;
max(makespan_individual(1,colo_aux1,cantMaqu,cantTrab,cantEtap,costos_sep
a));
max(makespan_individual(1,colo_aux2,cantMaqu,cantTrab,cantEtap,costos_sep
a))];
[val_1 , pos_1]=sort(aux_cos);
opti=pos_1(1,1);
opti2=pos_1(2,1);
tempo_re=Paises(Imper{k}(pos_coll,1),:);
switch(opti)
    case 2

Paises(Imper{k}(pos_coll,1),:)=Paises(Imper{k}(pos_col2,1),:);
        makespan(Imper{k}(pos_coll,1),1)=val_1(1,1);
        break;
    case 3
        Paises(Imper{k}(pos_coll,1),:)=colo_aux1;

```



```

pos= pos(1,1:cant_revo);
for s=1:cant_revo % recorre las colonias que se revolucionaran
    mi_colonia=Paises(Imper{k}(pos(1,s)),:);
    for et=1:cantEtap
        pos1=randperm(cantMaqu+cantTrab-1)+1+(et-
1)*(cantMaqu+cantTrab);
        pos1=pos1(1:2);
        aux_tem=mi_colonia(1,pos1(1,1));
        mi_colonia(1,pos1(1,1))=mi_colonia(1,pos1(1,2));
        mi_colonia(1,pos1(1,2))=aux_tem;
    end
    Paises(Imper{k}(pos(1,s)),:)=mi_colonia;

makespan(Imper{k}(pos(1,s)),1)=max(makespan_individual(Imper{k}(pos(1,s))
,Paises,cantMaqu,cantTrab,cantEtap, costos_sepa));
end
end
end

```

- **Función guerra entre imperios.**

```

function [Imper, cant_Colo, cantImper] = guerra_imperios(costo_Imper,
Imper, cant_Colo)
[val, pos] = max(costo_Imper);
diffcost=max(costo_Imper)-costo_Imper;
number_C= (diffcost/(sum(diffcost)))*size(Imper{pos},1);
[val1 , pos1]=sort(number_C, 'descend');
val1=val1(1:end-1);
pos1=pos1(1:end-1);
val1=max(round(val1),1);
if(sum(val1) < size(Imper{pos},1))
    val1(1,1)=val1(1,1)+size(Imper{pos},1)-sum(val1);
end
% asignacion de colonias
pos_asig = randperm(size(Imper{pos},1));
k=1;

```

```

asignados=0;
while (asignados<size(Imper{pos},1))
    cant_a_asig=min(size(Imper{pos},1)-asignados,vall(k));

    Imper{pos1(k)}(end+1:end+cant_a_asig)=Imper{pos}(pos_asig(k:k+cant_a_asig
-1));
    cant_Colo(pos1(k))=cant_Colo(pos1(k))+cant_a_asig;
    asignados=asignados+cant_a_asig;
    k=k+1;
end
Imper(pos,:)=[];
cant_Colo(pos)=0;
cant_Colo=cant_Colo(cant_Colo>0);
cantImper=size(Imper,1);
end

```