

Información Importante

La Universidad de La Sabana informa que el(los) autor(es) ha(n) autorizado a usuarios internos y externos de la institución a consultar el contenido de este documento a través del Catálogo en línea de la Biblioteca y el Repositorio Institucional en la página Web de la Biblioteca, así como en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad de La Sabana.

Se permite la consulta a los usuarios interesados en el contenido de este documento, para todos los usos que tengan finalidad académica, nunca para usos comerciales, siempre y cuando mediante la correspondiente cita bibliográfica se le dé crédito al trabajo de grado y a su autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, La Universidad de La Sabana informa que los derechos sobre los documentos son propiedad de los autores y tienen sobre su obra, entre otros, los derechos morales a que hacen referencia los mencionados artículos.

BIBLIOTECA OCTAVIO ARIZMENDI POSADA
UNIVERSIDAD DE LA SABANA
Chía - Cundinamarca



El presente formulario debe ser diligenciado en su totalidad como constancia de entrega del documento para ingreso al Repositorio Digital (Dspace).

TITULO	APLICACIÓN DE LA META-HEURÍSTICA COLONIA DE HORMIGAS PARA LA RESOLUCIÓN DE PROBLEMAS MULTI-OBJETIVO DE PROGRAMACIÓN DE LA PRODUCCIÓN EN FLOWSHOPS HÍBRIDOS (FLEXIBLES)		
SUBTITULO			
AUTOR(ES) Apellidos, Nombres (Completo) del autor(es) del trabajo	FABRICIO ANDRES NIEBLES ATENCIO		
PALABRAS CLAVE (Mínimo 3 y máximo 6)	FLOWSHOP FLEXIBLE		
	ALGORITMO DE COLONIA DE HORMIGAS		
	OPTIMIZACION MULTIOBJETIVO		
RESUMEN DEL CONTENIDO (Mínimo 80 máximo 120 palabras)	<p>Esta tesis estudia el problema de programación de la producción en talleres seriales (flowshop) híbridos o flexibles. Puesto que el problema es conocido por ser NP-duro, se propone un algoritmo de colonia de hormigas (ACO). Se estudia un problema multi-objetivo, en el cual se desea minimizar el makespan, la tardanza total y el número de trabajos tardíos. Se desarrolla un estudio experimental empleando instancias conocidas en la literatura. Puesto que es la primera vez en la literatura científica que se estudia este problema multiobjetivo, el procedimiento propuesto es comparado con respecto a soluciones heurísticas para los respectivos problemas monoobjetivo. Los resultados experimentales muestran la eficiencia y eficacia del método ACO propuesto.</p>		

Autorizo (amos) a la Biblioteca Octavio Arizmendi Posada de la Universidad de La Sabana, para que con fines académicos, los usuarios puedan consultar el contenido de este documento en las plataformas virtuales de la Biblioteca, así como en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, "Los derechos morales sobre el trabajo son propiedad de los autores", los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

**APLICACIÓN DE LA META-HEURÍSTICA COLONIA DE HORMIGAS PARA
LA RESOLUCIÓN DE PROBLEMAS MULTI-OBJETIVO DE
PROGRAMACIÓN DE LA PRODUCCIÓN EN FLOWSHOPS HÍBRIDOS
(FLEXIBLES)**

Trabajo de grado presentado como requisito parcial para obtener el título de

Magíster en Gerencia de Operaciones
(Modalidad de investigación)

Fabricio Andrés Niebles Atencio

Director:

Elyn Lizeth Solano Charris, M.Sc

Co-director:

Jairo R. Montoya Torres, Ph.D

Presentado públicamente el día 14 del mes de septiembre 2012

Jurado:

Gonzalo Mejía, Universidad de los Andes

Juan Pablo Caballero, Pontificia Universidad Javeriana

Universidad de La Sabana
Escuela Internacional de Ciencias Económicas y Administrativas
Chía, Colombia
2012

*Cuatro cosas son de las más pequeñas de la tierra,
Y las mismas son más sabias que los sabios:
Las hormigas, pueblo no fuerte,
Y en el verano preparan su comida...”*

Proverbios 30:24-25

*A Dios, quien me dio la sabiduría e
inteligencia, para culminar esta tesis;*

*A mis padres, abuelas y hermanos, porque
siempre me apoyaron y estimularon con su
amor y fortaleza;*

AGRADECIMIENTOS

El autor expresa su agradecimiento:

A la profesora ELYN LIZETH SOLANO CHARRIS (M.Sc), Directora del proyecto, por su apoyo y paciencia constantes a pesar de los inconvenientes presentados y por el voto de confianza otorgado para el desarrollo y consecución de este proyecto.

Al profesor JAIRO RAFAEL MONTOYA TORRES (Ph.D), Co-Director del proyecto, por sus sugerencias, apoyo y paciencia y por su estímulo constante para la culminación de manera satisfactoria del presente proyecto.

A las personas que me ayudaron en la ejecución de las pruebas del software desarrollado, María, Milena, Andrea, Daniela, Margarita y Priscila, porque gracias a su valiosa ayuda y colaboración, fue posible la culminación de esta tesis en el tiempo establecido para tal fin.

A los docentes de la Maestría en Gerencia de Operaciones y de la Escuela Internacional de Ciencias Económicas y Administrativas (EICEA), que aportaron con su gran conocimiento a la realización del presente trabajo de investigación.

De igual manera expreso mis agradecimientos a mis compañeros de estudio de la Maestría en Gerencia de Operaciones por su apoyo y constante transmisión de conocimientos y aquellas personas de la Universidad de La Sabana que aportaron de una u otra manera al desarrollo de esta tesis.

Por último, agradezco la beca terminal de maestría que se me otorgó por medio del proyecto titulado: Programación de Talleres de Producción Seriales Flexibles con Múltiples Objetivos mediante procedimientos Meta-heurísticos, cuya responsable es la profesora: Ing. Elyn Lizeth Solano Charris, M.Sc.

RESUMEN

Con el fin de mejorar los niveles de competitividad, las empresas de manufactura y de servicio están obligadas a la implementación constante de procedimientos formales que les permitan optimizar sus procesos. En ese sentido, en lo referente a las operaciones de manufactura, la logística de producción, y más específicamente la programación de operaciones, juega un papel importante en cuanto al uso eficiente de los recursos. La programación de operaciones (*scheduling*, en inglés) es una rama de la optimización combinatoria que consiste en la asignación de recursos para la realización de un conjunto de actividades con el fin de optimizar uno o varios objetivos. Debido a la complejidad intrínseca en la mayoría de los problemas de programación de la producción, los cuales son del tipo NP-duro (esto es, el tiempo que requieren para resolver un caso particular de un problema crece en el peor de los casos de manera exponencial con respecto al tamaño del problema), los métodos exactos convencionales de resolución tales como: programación lineal, entera y mixta, entre otros, no son eficientes en términos del tiempo de cálculo para llegar a la solución óptima. Por lo tanto, se hace necesario el uso de enfoques alternativos para resolver este tipo de problemas en un tiempo razonablemente corto para el tomador de decisiones, sobre todo aquellas que se toman diariamente. Dentro de estos enfoques se encuentran las metaheurísticas, que consisten en procedimientos formales desarrollados con el fin de superar esta dificultad que se presenta con los métodos tradicionales. Los procedimientos meta-heurísticos más comunes para la resolución de problemas combinatorios son: los algoritmos genéticos, la búsqueda tabú, la colonia de hormigas y el recocido simulado entre otros. En ese sentido, el objetivo de esta tesis es desarrollar un sistema de soporte para la toma de decisiones que permita automatizar el proceso de programación de operaciones en configuraciones de producción tipo flowshop flexible, en donde se pretenda optimizar múltiples criterios, a través de una herramienta computacional basada en la metaheurística colonia de hormigas y así responder la siguiente pregunta de investigación: **¿De qué manera el uso de la metaheurística, Colonia de Hormigas, considerando múltiples objetivos afecta la calidad de las soluciones para programar las operaciones en un flowshop flexible?**

Se estudia un problema multiobjetivo de programación de la producción, en el cual se minimizan el lapso de fabricación o makespan, la tardanza total y el número de trabajos tardíos para una configuración tipo flowshop flexible. Se desarrolla un estudio experimental empleando instancias conocidas en la literatura. Puesto que no se ha encontrado evidencia en la literatura científica de que se haya estudiado este problema multiobjetivo con anterioridad, el procedimiento propuesto es comparado con respecto a soluciones heurísticas para los respectivos problemas mono-objetivo. El estudio experimental realizado muestra que el método propuesto en esta investigación proporciona buenos resultados para instancias con dos y cinco estaciones, aún para un número medianamente grande de tareas.

Palabras clave: Metaheurísticas, Colonia de hormigas, Optimización multiobjetivo, Flowshop Flexible, Programación de las operaciones.

ABSTRACT

In order to improve levels of competitiveness, manufacturing and service companies are required to constantly implement formal procedures that allow them to optimize their processes. In this regard, when it comes to the manufacturing operations, production logistics, and more specifically the operations scheduling, play an important role in the efficient use of the resources. The programming operations (scheduling, in English) is a branch of combinatorial optimization which involves allocating resources to perform a number of activities with the aim of optimizing one or more objectives. Due to the intrinsic complexity in most programming problems of production, which are of the NP-hard type (i.e. the time required to solve a particular case of problem grows exponentially in the worst case with respect to the size of the problem), the exact and conventional methods of resolution such as linear programming, integer and mixed programming, among others, are not efficient in terms of computing time to reach the optimal solution. Therefore, it becomes necessary to use alternative approaches to solve these problems in a reasonably short time for the decision maker, especially those that are taken daily. Meta-heuristics are within these approaches, and consist of formal procedures that are developed in order to overcome this difficulty encountered with traditional methods. The most common meta-heuristic procedures to solve combinatorial problems are: genetic algorithms, tabu search, ant colony and simulated annealing, among others. In this regard, the goal of this research is to develop a supporting system for decision making that automates the process of scheduling operations in production settings of the flexible flowshop type, where it is intended to optimize multiple criteria, through a computational tool based on the meta-heuristic ant colony and then answer the following research question: **How does the use of metaheuristic, ant colony, considering multiple objectives affect the quality of the solutions to schedule operations in a flexible flowshop?** It studied a problem of multi-objective production scheduling, in which the makespan, the total tardiness and number of tardy jobs for a hybrid flowshop type configuration are minimized. It developed an experimental study using known instances in the literature. Since no evidence has been found in the scientific literature that this

multi-objective problem was previously studied, the proposed method is compared with respect to heuristic solutions to the respective mono-objective problems. The experimental analysis accomplished shows that the method proposed in this study provides good results for instances with two and five stages, even for a fairly large number of tasks.

Keywords: Metaheuristics, Ant colony, Multiobjective optimization, Hybrid flowshop, Scheduling operations.

TABLA DE CONTENIDO

1. INTRODUCCIÓN	14
1.1. PLANTEAMIENTO DEL PROBLEMA	14
1.1.1. Contexto	14
1.1.2. Objetivo general	16
1.1.3. Objetivos específicos.....	16
1.1.4. Justificación	17
1.2. METODOLOGÍA DE INVESTIGACIÓN	18
1.2.1. Tipo de estudio.....	18
1.2.2. Método de investigación.....	18
1.2.3. Fuentes y técnicas	18
1.2.4. Tratamiento de la información.....	19
1.2.5. Metodología propuesta.....	19
1.3. ORGANIZACIÓN DEL DOCUMENTO	20
2. MARCO TEORICO	21
2.1. Programación de operaciones en planta	21
2.2. Optimización Multiobjetivo.....	23
2.3. Heurísticas y Metaheurísticas	27
2.4. OPTIMIZACION POR COLONIA DE HORMIGAS	30
3. REVISIÓN DE LA LITERATURA	34
4. RESOLUCION DEL PROBLEMA DEL FLOWSHOP FLEXIBLE MULTI- OBJETIVO A TRAVÉS DE LA METAHEURISTICA COLONIA DE HORMIGAS	39
4.1. PLANTEAMIENTO DEL MODELO FORMAL.....	39
4.2. EL ALGORITMO MULTI-OBJETIVO DE COLONIA DE HORMIGAS PROPUESTO	42
4.2.1. Grafo Disyuntivo	43
4.2.2. Procedimiento constructivo	45
4.2.3. Selección de operaciones	46

4.2.4. Modelo de representación de feromonas	46
4.2.5. Información Heurística	47
4.2.6. Selección de máquinas.....	48
4.2.7. Mecanismo de actualización de feromonas	49
5. EXPERIMENTACIÓN Y ANÁLISIS DE RESULTADOS.....	51
5.1. DISEÑO DE EXPERIMENTOS.....	52
5.2. COMPORTAMIENTO DEL ALGORITMO FF-MUCH PROPUESTO.....	54
5.3. ANALISIS COMPARATIVO FF-MUCH VS. LEKIN	59
6. CONCLUSIONES Y FUTUROS TRABAJOS	77
7. REFERENCIAS BIBLIOGRAFICAS.....	81
8. ANEXOS	95
ANEXO 1. DISEÑO DE EXPERIMENTOS REALIZADO.....	96
ANEXO 2. GUIA DE USUARIO DE LA HERRAMIENTA FF-MUCH.....	129

LISTA DE FIGURAS

Figura 1. Representación de un flowshop flexible (o híbrido) con dos estaciones....	23
Figura 2. Problema de optimización multi-objetivo max-max.	26
Figura 3. Experimento del doble puente	31
Figura 4. Pseudocódigo genérico de la metaheurística ACO.	31
Figura 5. Representación de un flowshop flexible por medio de un grafo disyuntivo.....	44
Figura 6. Algoritmo propuesto para resolver el problema del flowshop flexible	45
Figura 7. Medición de la convergencia del algoritmo en términos del número de iteraciones para el makespan.....	55
Figura 8. Medición de la convergencia del algoritmo en términos del número de iteraciones para la Tardanza total.....	56
Figura 9. Medición de la convergencia del algoritmo en términos del número de iteraciones para el número de trabajos tardíos.	57
Figura 10. Efecto del factor Alfa y efecto del factor Rho sobre el makespan.....	107
Figura 11. Interacción de segundo orden entre los factores feromonas y rho e interacción de tercer orden entre los factores Alfa, q_0 y Rho para el makespan....	116
Figura 12. Efecto del factor Alfa y efecto del factor Rho sobre la variable tardanza.....	116
Figura 13. Interacción de segundo orden entre los factores feromonas y entre q_0 y rho para la tardanza total.....	116
Figura 14. Gráfico de interacciones para Beta y Rho y para Alfa, Feromonas y q_0	124
Figura 15. Paso 1 de instalación del programa FF-MUCH.....	129
Figura 16. Paso 2 de instalación del programa FF-MUCH.....	130
Figura 17. Paso 3 de instalación del programa FF-MUCH.....	131
Figura 18. Paso 4 de instalación del programa FF-MUCH.....	132
Figura 19. Paso 5 de instalación del programa FF- MUCH.....	133
Figura 20. Paso 6 de instalación del programa FF-MUCH.....	134
Figura 21. Paso 7 de instalación del programa FF-MUCH.....	135
Figura 22. Paso 8 de instalación del programa FF-MUCH.....	135
Figura 23. Paso 1 de utilización del programa FF-MUCH.....	136
Figura 24. Paso 2 de utilización del programa FF-MUCH.....	137
Figura 25. Paso 3 de utilización del programa FF-MUCH.....	137
Figura 26. Paso 4 de utilización del programa FF-MUCH.....	138
Figura 27. Paso 5 de utilización del programa FF-MUCH.....	139
Figura 28. Paso 6 de utilización del programa FF-MUCH.....	140

Figura 29. Paso 7 de utilización del programa FF-MUCH.....	141
Figura 30. Paso 8 de utilización del programa FF-MUCH.....	142
Figura 31. Paso 9 de utilización del programa FF-MUCH.....	143
Figura 32. Paso 10 de utilización del programa FF-MUCH.....	144
Figura 33. Paso 11 de utilización del programa FF-MUCH.....	145
Figura 34. Paso 12 de utilización del programa FF-MUCH.....	146

LISTA DE TABLAS

Tabla 1. Diferencia porcentual entre LEKIN® y FF-MUCH para 10 instancias de 20 trabajos y 2 estaciones.....	63
Tabla 2. Diferencia porcentual entre LEKIN® y FF-MUCH para 10 instancias de 20 trabajos y 5 estaciones.....	64
Tabla 3. Diferencia porcentual entre LEKIN® y FF-MUCH para 10 instancias de 20 trabajos y 8 estaciones.....	65
Tabla 4. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 50 trabajos y 2 estaciones.....	66
Tabla 5. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 50 trabajos y 5 estaciones.....	67
Tabla 6. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 50 trabajos y 8 estaciones.....	68
Tabla 7. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 100 trabajos y 2 estaciones.....	69
Tabla 8. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 100 trabajos y 5 estaciones.....	70
Tabla 9. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 100 trabajos y 8 estaciones.....	71
Tabla 10. Escenario del flowshop flexible estudiado para la instancia P20S2T01.....	97
Tabla 11. Fechas de entrega para los trabajos de la instancia P20S2T01.....	99
Tabla 12. Escenario del flowshop flexible estudiado para la instancia P50S2T01.....	99
Tabla 13. Fechas de entrega para los trabajos de la instancia P50S2T01.....	99
Tabla 14. Escenario del flowshop flexible estudiado para la instancia PH1S2T1.....	99
Tabla 15. Fechas de entrega para los trabajos de la instancia PH1S2T1.....	99
Tabla 16. Escenario del flowshop flexible estudiado para la instancia seleccionada en el diseño experimental realizado	100
Tabla 17. Fechas de entrega para los trabajos de la instancia seleccionada para el diseño experimental realizado.....	100
Tabla 18. Matriz de datos del Diseño Experimental realizado.....	104
Tabla 19. Tabla ANOVA para la variable de respuesta MAKESPAN.....	106

Tabla 20. Prueba de comparaciones múltiples de Tukey para el factor rho	107
Tabla 21. Prueba normalidad de Kolmorov-Smirnov.....	111
Tabla 22. Prueba de correlación de los residuos tipificados para la variable makepan.....	112
Tabla 23. Prueba de Levene de igualdad de varianzas para el makespan.....	113
Tabla 24. Tabla ANOVA para la variable de respuesta Tardanza Total.....	114
Tabla 25. Prueba de comparaciones múltiples de Tukey para el factor rho.....	115
Tabla 26. Prueba de Kolmogorv-Smirnov para normalidad de residuos tipificados de la variable Tardanza Total.....	119
Tabla 27. Prueba de Correlación de los residuales de la variable Tardanza Total.....	120
Tabla 28. Prueba de Levene para igualdad de varianzas de la variable Tardanza total.....	121
Tabla 29. Tabla ANOVA para la variable de respuesta número de trabajos tardíos.	122
Tabla 30. Prueba de comparaciones múltiples de Tukey para Rho.	124
Tabla 31. Prueba de normalidad Kolmogorov-Smirnov para el número de trabajos tardíos.....	126
7	
Tabla 32. Prueba de correlación para el número de trabajos tardíos.....	1278
Tabla 33. Prueba de Levene para igualdad de varianzas.....	1289

1. INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

1.1.1. Contexto

Con el fin de mejorar los niveles de competitividad, las empresas de manufactura y de servicio se ven en la necesidad de implementar constantemente procedimientos formales que les permitan optimizar sus procesos. En particular, en lo referente a las operaciones de manufactura, la logística de producción juega un papel importante en cuanto al uso eficiente de los recursos. Esta eficiencia es comúnmente medida en términos monetarios a nivel estratégico, en indicadores globales de desempeño y a nivel táctico o en tasa de utilización de los recursos, a nivel operativo. Este proyecto se focaliza en este último nivel de toma de decisiones, y en particular en los procesos de programación de las actividades de producción. Esta consiste en la asignación de recursos para la realización de un conjunto de actividades con el fin de optimizar uno o varios objetivos (Pinedo, 2008). Fruto de las aplicaciones iniciales en problemas de manufactura, los recursos son comúnmente llamados “máquinas” y las actividades se llaman “tareas”, “trabajos” u “órdenes de producción”.

Debido a la complejidad intrínseca de la mayoría de problemas de *scheduling* (i.e., complejidad computacional de tipo NP-completo), la resolución a través de modelos matemáticos y otros métodos exactos es eficiente (en términos de tiempo de cálculo) únicamente para casos (instancias) de tamaño pequeño o mediano. Para casos de gran tamaño se ha preferido el uso de procedimientos metaheurísticos (Jourdan et al., 2009). El uso de metaheurísticas también es preferible al momento de resolver problemas más cercanos a la realidad, los cuales son cada vez más complejos debido a las características y restricciones que se desean modelar. El auge en el avance y mejoramiento en el desempeño de los computadores también ha favorecido el desarrollo de modelos y procedimientos para la programación de operaciones. Dentro de los procedimientos metaheurísticos más comunes para la resolución de problemas combinatorios se encuentran los

algoritmos genéticos, la búsqueda tabú, las colonia de hormigas, y el recocido simulado entre otros (T'kindt y Billaut, 2006; Jourdan et al, 2009; Solano-Charris et al., 2009,).

Dentro de la tipología de problemas de *scheduling* se encuentra el modelo de taller en serie híbrido o flexible (*flexible flowshop* o *hybrid flowshop problem*), es decir, los talleres en serie que se componen por más de una máquina por estación. Debido a que es un problema NP-completo (Gupta, 1988), los trabajos en la literatura se han concentrado mayoritariamente en la optimización de una sola función objetivo, y pocos estudios existen en los cuales se consideren múltiples objetivos (Solano-Charris et al., 2009). La metaheurística basada en Colonia de Hormigas (*ant colony optimization*) ha resultado ser un procedimiento muy eficiente para la resolución de problemas flowshop híbridos con múltiples objetivos (T'kindt et al., 2002; Solano-Charris, 2008; Solano-Charris et al., 2009).

El problema del flowshop flexible (FFSP) o híbrido es una generalización del problema del flowshop ordinario en el cual un conjunto de tareas deben ser ejecutadas en un conjunto de estaciones en serie, cada una con varias máquinas paralelas. La solución consiste básicamente en resolver dos subproblemas: por una parte, se debe resolver un problema de secuenciación (es decir, establecer el orden de procesamiento de los trabajos en las estaciones), y por otro lado se debe resolver un problema de asignación (es decir, determinar en la máquina de cada estación que ejecuta la operación).

Formalmente, se tiene un conjunto de n tareas o trabajos a ser procesados en s estaciones, cada una con $m_i \geq 1$ máquinas en paralelo. Cada trabajo j está compuesto de varias operaciones ($O_{1j}, O_{2j}, \dots, O_{sj}$), las cuales deben ser ejecutada en ese el mismo orden en las diferentes estaciones. La operación O_{ij} debe ser realizada por una máquina del conjunto disponibles en la i -ésima estación durante p_{ij} unidades de tiempo independiente de la máquina. No se autoriza la interrupción de los trabajos en curso de ejecución; los tiempos de alistamiento de las máquinas y el tiempo de transporte entre las operaciones son despreciables, las máquinas son independientes entre sí, al igual que los trabajos; y en un

instante determinado, una máquina sólo puede ejecutar una operación a la vez. El objetivo es determinar un programa de producción de los n trabajos en diferentes estaciones que optimice una o varias funciones objetivo. Para el caso de esta investigación, se considerarán como funciones objetivos el makespan, la tardanza total y el número de trabajos entregados tarde. El primer objetivo se calcula como el instante de terminación del último trabajo del programa en la última máquina; mientras que para los otros objetivos o criterios se necesita comparar el tiempo de terminación de cada trabajo, denotado como C_{sj} , con la fecha de entrega prometida el cliente, denotada como d_j . Si $d_j < C_{sj}$, entonces la tarea j es considerada como tardía.

Para el caso mono-objetivo de minimización del makespan o lapso de producción, el flowshop flexible es conocido por ser un problema de tipo NP-duro Gupta (1988), Santos et al. (1995). Para el caso multiobjetivo considerado en este trabajo, el problema es al menos igual de difícil. Por lo tanto, se justifica el uso de algoritmos meta-heurísticos para su resolución. En ese sentido, en esta investigación se propone un algoritmo de colonia de hormigas (ACO, Ant Colony Optimization) de concepción multiobjetivo, que permita optimizar la programación de operaciones en configuraciones de producción tipo flowshop flexible con múltiples criterios de optimización.

1.1.2. Objetivo general

Desarrollar una herramienta computacional, basada en la metodología de optimización por colonia de hormigas con el fin de automatizar el proceso de toma de decisiones para la programación de las operaciones, en empresas con configuraciones tipo flowshop flexible y múltiples criterios a optimizar.

1.1.3. Objetivos específicos

- Proponer un algoritmo de Colonia de Hormigas, e implementarlo en una herramienta computacional, para la solución del problema de programación de las operaciones en un flowshop flexible, cuyos criterios de optimización a considerar son: el makespan, la tardanza total y el número de trabajos tardíos.

- Realizar un estudio experimental a través de un software estadístico, con el fin de parametrizar el algoritmo propuesto.
- Comparar los resultados generados por la herramienta desarrollada, con aquellos obtenidos con el software LEKIN®, con el fin de poder validar el funcionamiento de dicha herramienta.
- Probar el algoritmo propuesto, utilizando instancias conocidas de la literatura, para evaluar su desempeño en términos de la calidad de las soluciones obtenidas, convergencia y el tiempo de ejecución empleado en su cálculo.

1.1.4. Justificación

El desarrollo e implementación de métodos de resolución avanzados para los problemas del tipo flow shop flexible con múltiples objetivos basados en algoritmos meta-heurísticos ha sido un área de investigación que ha cobrado un creciente interés en los últimos años. Se han realizado varias investigaciones y publicaciones con relación a esta temática, sin embargo aún es un tema abierto en la literatura, debido a que una de las limitaciones de comparar la heurística con el óptimo es el amplio espectro de posibles secuencias o soluciones para el problema bajo estudio (Solano-Charris et al., 2008).

Por otra parte, y como se mencionó anteriormente, la mayoría de los trabajos e investigaciones que se han realizado sobre el flow shop flexible se centran en la optimización de un solo objetivo o criterio; este por lo general es el makespan. Del mismo modo, hay una extensa cantidad de publicaciones e investigaciones para la optimización multicriterio para sistemas con configuraciones que no son del tipo flowshop flexible. Por lo general estos estudios se han centrado en estudiar los talleres de producción seriales (flowshop), probablemente el más estudiado, y jobshop. En ese sentido, se considera pertinente el trabajo de investigación desarrollado dentro del contexto académico y científico actual, por lo que se requiere implementar la herramienta propuesta en ambientes con restricciones totalmente diferentes a las que a la fecha se han determinado.

1.2. METODOLOGÍA DE INVESTIGACIÓN

1.2.1. Tipo de estudio

El tipo de estudio enmarcado en el proyecto se considera de tipo descriptivo, aplicativo, ya que dentro de la investigación, es necesario llegar a los factores o variables que desencadenan el problema. Así mismo, se plantea como solución factible la aplicación del algoritmo propuesto para dar solución al problema de programación de operaciones en empresas con sistemas de producción flexibles en serie y múltiples objetivos de programación, procediendo a la descripción última de su utilización y los beneficios que surgen de la aplicación del sistema a desarrollar (Pacheco-Espejel y Cruz-Estrada, 2006).

1.2.2. Método de investigación

Esta investigación es de tipo básica y a la vez tecnológica. Según Pacheco y Cruz (2006) para el primer tipo se pretende obtener la solución a un problema de la realidad a través de métodos formales. En este, se busca determinar la manera de programar las operaciones en un taller de producción tipo flowshop flexible multiobjetivo a través del planteamiento de un modelo matemático formal que permita representar y caracterizar el problema bajo estudio. Para el tipo de investigación tecnológica, se realiza el diseño de una herramienta computacional, que para el presente caso corresponde al software basado en la metaheurística colonia de hormigas, por medio de la cual se proporciona una solución al problema planteado.

1.2.3. Fuentes y técnicas

La información requerida para el proyecto se toma de fuentes secundarias debido a que se necesita información escrita que haya sido recopilada y transcrita por personas que han recibido tal información a través de otras fuentes escritas o por un participante en un suceso. Estas son de naturaleza diversa como: textos, documentos, tesis, artículos de revistas científicas e Internet (Bases de Datos).

1.2.4. Tratamiento de la información

La información, materia prima principal de esta investigación, se clasifica teniendo en cuenta la que esté relacionada con los objetivos del presente trabajo y que involucre los diferentes aspectos que caracterizan el problema estudiado bajo iguales o diferentes enfoques, de modo que estos sirvan de referencia o punto de comparación con miras a lograr la consecución de dichos objetivos.

1.2.5. Metodología propuesta

Para lograr el objetivo general de investigación se seguirán las siguientes etapas:

- Planteamiento del modelo matemático formal para representar el problema de optimización del flowshop híbrido multiobjetivo y caracterizar los elementos (variables, parámetros y restricciones) del sistema a modelar.
- Desarrollo del algoritmo de colonia de hormigas: descripción paso a paso del pseudocódigo; definición de los parámetros utilizados, la metodología alterna para comparar los resultados y la validación para el contexto del flowshop híbrido multiobjetivo según el caso particular de estudio.
- Diseño y desarrollo de la herramienta computacional e interfaz gráfica de usuario. Esta herramienta consistió en un software desarrollado en el lenguaje de programación Visual Basic 6.0 ® para resolver el problema planteado. Así mismo, el diseño de una interfaz amigable con el usuario que utilice la herramienta, de modo que se cumpliera con los requerimientos de funcionalidad, facilidad de uso, portabilidad y adaptabilidad (flexibilidad).
- Definición y utilización de instancias o conjuntos de datos, corridas del algoritmo y análisis de resultados que permitieran la validación de la herramienta y la medición de su desempeño y/o rendimiento. Las instancias con la que se probó la herramienta se

tomaron de la página web de OR-Library¹, en la sección “Multiprocessor task scheduling in multistage hybrid flowshops”

- Documentación y entrega de informes. Esto incluyó los informes de avance y final del trabajo de grado así como una guía de usuario para el uso de la herramienta computacional desarrollada.

1.3. ORGANIZACIÓN DEL DOCUMENTO

El presente documento está organizado de la siguiente manera: En el capítulo 2 se proporcionan los conceptos generales de programación de operaciones, optimización multiobjetivo y metaheurísticas. Del mismo modo, se realiza una descripción de la metaheurística colonia de hormigas. El capítulo 3 presenta una revisión de la literatura relacionada con el problema. En el capítulo 4 se describe el enfoque propuesto para resolver el problema bajo estudio. Los resultados experimentales se presentan en el capítulo 5. Finalmente, el capítulo 6 presenta las conclusiones del trabajo y algunas recomendaciones o perspectivas para investigaciones futuras.

¹<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/multiflowinfo.htm>

2. MARCO TEORICO

2.1. Programación de operaciones en planta

La programación de operaciones, conocida también como programación de tareas (en inglés, *scheduling*) consiste en la organización, elección y asignación de tiempos al uso de recursos para ejecutar todas las actividades (tareas u operaciones) requeridas, para producir las salidas deseadas en los tiempos deseados teniendo en cuenta las restricciones de tiempo y las relaciones entre las actividades y los recursos. (Morton y Pentico, 1993). Esta definición implica dos aspectos importantes: si los recursos no son limitados, no existe el problema de programar las operaciones. Por otro lado, la programación de operaciones, en sí misma, es un proceso de toma de decisiones que tiene como objetivo la optimización de uno o varios criterios (Montoya-Torres, 2010).

Los recursos y las tareas pueden ser de diferentes tipos. Los recursos pueden ser las máquinas en un taller de manufactura, la pista de aterrizaje en un aeropuerto, los empleados en un proyecto de construcción, los procesadores en un equipo de cómputo, y así sucesivamente. En cuanto a las tareas, éstas pueden ser las operaciones de fabricación de un producto, las operaciones de despegue y aterrizaje de aviones en el aeropuerto, las etapas del proyecto de construcción, la ejecución de los programas de cómputo, etc. (Montoya-Torres, 2010)

Como se mencionó anteriormente, la programación de tareas es un proceso de toma de decisiones que se presenta en prácticamente todos los sistemas productivos, ya sea de manufactura o de servicios, al igual que en los sistemas computacionales. La toma de decisiones en cuanto a la programación de tareas o actividades también se presenta en sistemas de transporte y distribución como también en otros tipos de sistemas productivos del área de servicios (Montoya-Torres, 2010).

La realización de un programa de ejecución de tareas puede ser difícil tanto desde el punto de vista técnico como desde la perspectiva de su implementación. Las dificultades

en los aspectos técnicos son básicamente las mismas que se presentan al resolver los problemas de optimización combinatoria y modelamiento estocástico. A nivel de la implementación, la mayor dificultad radica tanto en el modelado de los problemas reales de programación de tareas como en la recolección de la información pertinente (Montoya-Torres, 2010). Es decir, la enorme cantidad de combinaciones existentes en un sistema productivo, hace imposible la evaluación de todos los posibles programas para la determinación del programa de operaciones óptimo, incluso implementando el más poderoso sistema de cómputo que pueda ser adquirido en la actualidad. Estas condiciones han impulsado el desarrollo de técnicas inteligentes de optimización, las cuales no están diseñadas bajo la premisa de obtener la solución óptima de un problema, sino en cambio, sacrificar un pequeño margen de exactitud en beneficio de no requerir una desmesurada potencia de cómputo. Un importante grupo de técnicas de optimización inteligentes son las denominadas heurísticas y metaheurísticas (Pinedo, 2005).

Por otra parte, el modelo de programación de operaciones en talleres de manufactura consiste en configuraciones especializadas (o talleres especializados) en las cuales se tiene un conjunto de m recursos y la realización de cada trabajo consiste en la ejecución de varias operaciones. Según la ruta que las tareas o trabajos deben seguir dentro del sistema, se distinguen las configuraciones de flowshop, jobshop y openshop (Montoya-Torres, 2010).

Esta investigación se interesa en estudiar la configuración de tipo *flowshop híbrido (o flexible)*, en la cual la realización de un trabajo consiste en la ejecución de m operaciones en serie (Montoya-Torres, 2010). Cada operación es realizada en una estación en la cual se tienen varios recursos (idénticos o no) en paralelo. Una vez terminada la operación en una de las máquinas de la estación, el trabajo pasa a la siguiente estación. Se considera que cada trabajo será procesado en cada etapa una sola vez y solo una y que todos los trabajos siguen la misma ruta de procesamiento (e.g., primera estación, luego segunda estación, tercera estación, y así sucesivamente hasta terminar su procesamiento). El tiempo de procesamiento para cada producto en cada estación estará determinado por el

tiempo de alistamiento de la máquina, el tiempo en que la máquina transforma al producto (i.e. lo procesa) y el tiempo para retirar el producto de la máquina, antes de colocar el siguiente (Solano-Charris, 2008). La figura 1 ilustra este tipo de configuración.

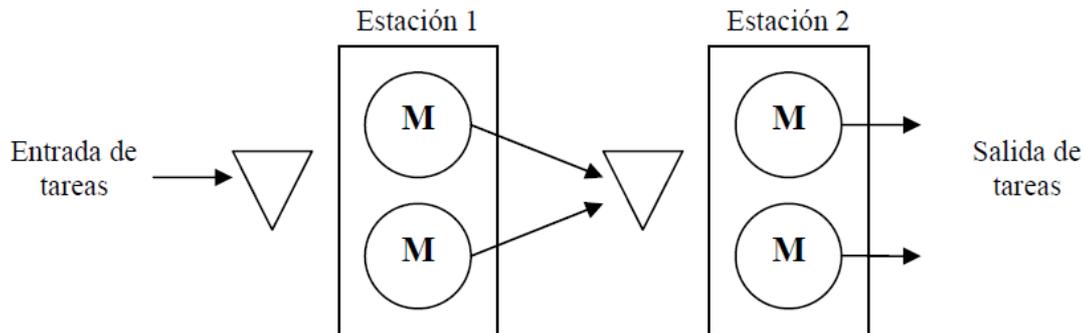


Figura 1. Representación de un flowshop flexible (o híbrido) con dos estaciones. Fuente: Montoya Torres. 2010.

En tal sentido, con esta investigación se propone una nueva metodología para resolver el problema de programación de las operaciones en flowshop flexibles. A diferencia de la mayoría de los trabajos publicados en la literatura, esta investigación se concentra en la optimización de varias funciones objetivos. A continuación se presentan algunos conceptos básicos sobre la optimización multiobjetivo.

2.2. Optimización Multiobjetivo

Problemas con múltiples objetivos han tomado lugar de manera natural en la mayoría de las disciplinas y sus soluciones han sido un desafío para los investigadores por mucho tiempo. A pesar de la considerable variedad de técnicas desarrolladas en Investigación de Operaciones y otras disciplinas para abordar estos problemas, la complejidad de sus soluciones sugieren enfoques alternativos.

La Optimización multiobjetivo puede ser definida como el problema de encontrar: “un vector de variables de decisión el cual satisface unas restricciones y optimiza un vector función cuyos elementos representan las funciones objetivos. Estas funciones constituyen

una descripción matemática de los diferentes criterios, los cuales generalmente están en conflicto entre sí. Por lo tanto, el término “optimizar” se refiere a encontrar una solución que proporcione los valores de todas las funciones objetivo de modo que sean aceptables para el tomador de decisiones.” (Coello, et al., 2007).

La optimización multiobjetivo no es tan reciente como se podría pensar. De hecho, algunos autores indican que la optimización multiobjetivo es una parte inherente del equilibrio económico y, en consecuencia, podría remontarse desde 1776 en el tratado de Adam Smith “La prosperidad de las naciones” (Coello, et al., 2007).

Aquí también merece mencionarse el concepto general de equilibrio económico a menudo atribuido a Léon Walras. Sin embargo, William Stanley Jevons, Carl Menger, Francis Ysidro Edgeworth y Wilfredo Pareto también realizaron importantes trabajos en este campo en el período comprendido entre 1874 y 1906 (Coello, et al., 2007).

De igual manera, e íntimamente relacionada a la optimización multi-objetivo está la teoría de los juegos psicológicos y la noción de estrategias de juego (basado en el análisis de la psicología del adversario), las cuales son atribuidas a Félix Édouard Émile Borel (Coello, et al., 2007).

Gran parte de los problemas del mundo real implican la optimización simultánea de dos o más criterios que generalmente presentan conflictos entre ellos; es decir, la mejora en uno conduce a un deterioro en el otro. La presencia de tales tipos de problemas es tan significativa que consume gran parte de nuestro tiempo cotidiano de decisión.

No obstante, las propuestas computacionales de automatización que se han presentado para resolverlos habitualmente se limitan a convertir el problema de objetivos múltiples en uno de un solo objetivo (Solano-Charris, 2008). Esta reducción es debida a los modelos matemáticos empleados y puede realizarse de varias maneras: por ejemplo, se prioriza uno de los objetivos y los demás se colocan como restricciones; o también se

genera un objetivo compuesto otorgando pesos a los objetivos en juego y armando una suma ponderada de los mismos. De todos modos, ninguna de estas reducciones refleja fielmente el problema y, por tanto, tampoco otorga soluciones completamente satisfactorias. A pesar de esto, el estado actual de la ciencia podría generar mejores resultados ya que existen modelos matemáticos que se ajustan mejor a la naturaleza de estos problemas. Tales modelos provienen de un área de la Investigación de Operaciones conocida como Optimización con Objetivos Múltiples o Multi-objetivo (Solano-Charris, 2008).

Fundamentación Matemática

Formalmente, la optimización multiobjetivo, se introduce para tratar problemas del tipo:

$$\begin{aligned} \text{Max} - \text{Min } f(x) &= (f_1(x), \dots, f_m(x))^T \\ \text{s. a. } x &\in \Omega \subseteq \mathbb{R}^n, \end{aligned} \quad (1)$$

Con una función $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ con $m \in \mathbb{N}$, $m \geq 2$, y $\Omega \subseteq \mathbb{R}^n$, un conjunto cerrado, denominado conjunto Pareto que toma cada vez más importancia y que se explicará posteriormente.

En otras palabras, el problema general multiobjetivo que requiere la optimización de k objetivos simultáneamente puede ser formulado como:

$$\begin{aligned} \text{Max (Min) } Z_j &= f_j(x) \text{ con } j = 1, 2, \dots, k \\ \text{s. a. } g_i(x) &\leq b_i. \end{aligned} \quad (2)$$

Donde x es un vector p -dimensional de variables en el espacio de decisión X y $g_i(x)$ son restricciones de desigualdad. En optimización, la solución “óptima” es aquella en la que se alcanzan los valores máximos o los valores mínimos de todos los objetivos simultáneamente. Es decir, que en un problema multiobjetivo la solución x^* es óptima si y solo si $x^* \in S$ donde S es la región factible y $f_i(x^*) \geq f_i(x)$ para todo i y para todo $x \in S$. Sin embargo, en el caso de la optimización con objetivos conflictivos, por lo general no

existe una única solución. En dicho caso puede ser mejor obtener lo que los economistas llaman soluciones “eficientes” o no dominadas (Bagchi, 1999).

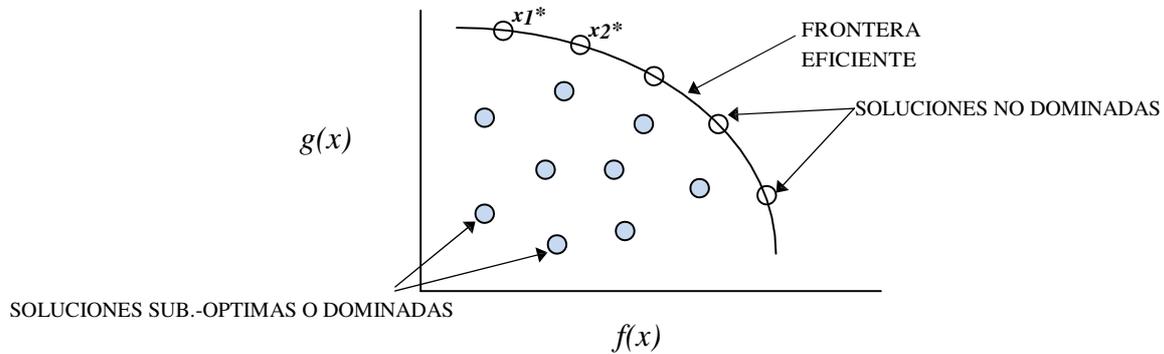


Figura 2. Problema de optimización multi-objetivo max-max. Fuente: Bagchi (1999).

Una solución eficiente (también llamada solución Óptima de Pareto) es una en la que ningún aumento puede ser obtenido en cualquiera de los objetivos sin causar una disminución simultánea en al menos uno de los objetivos restantes. Para el problema bi-objetivo de la **Figura 2**, cada solución x_i^* en la frontera eficiente es tal que no existe otra solución factible x para la cual $f(x_i) \geq f(x_i^*)$ y $g(x_i) \geq g(x_i^*)$. Para un problema multi-objetivo, x_i^* es una solución eficiente si y solo si no existe un $x \in S$ tal que $f_j(x) \geq f_j(x_i^*)$ para todo j y $f_j(x) > f_j(x_i^*)$ para al menos un j .

En general, no hay solo una mejor solución la que minimiza o maximiza todas las funciones objetivos al mismo tiempo ya que el conjunto solución, llamado conjunto eficiente o conjunto Pareto, es muy grande. Especialmente en aplicaciones de ingeniería, la información sobre el conjunto eficiente es importante. Además, teniendo la totalidad del conjunto solución disponible, el tomador de decisiones consigue adentrarse en la estructura misma del problema por lo que la meta es generar una aproximación representativa de este conjunto (Barichard, et al., 2009).

En síntesis, la solución adoptada para el problema de tomar en cuenta los criterios en conflicto, depende de la información que el tomador de decisiones puede suministrar. Desde el punto de vista del analista, es posible clasificar diferentes métodos para determinar el frente de Pareto por medio de esta información. El lector interesado en profundizar sobre estos métodos aplicados a problemas de programación de operaciones, puede consultar el libro de (T'Kindt, Billaut 2006).

Por otra parte, se puede desagregar un criterio en uno o varios criterios más generales adicionando nuevos parámetros (pesos, metas, etc.) al problema. Generalmente, lo más interesante del resultado para estos nuevos criterios es, la mayor dificultad de su aplicación (cambio de parámetros, complejidad algorítmica, etc.) Por lo tanto, la escogencia de un método necesita la conciliación entre la calidad de las soluciones calculables y la facilidad de su aplicación o implementación (T'Kindt, Billaut 2006).

En ese sentido, la metodología propuesta en el presente trabajo de grado como alternativa de solución al problema multiobjetivo planteado, obtiene las soluciones no dominadas para un conjunto de corridas o réplicas del algoritmo, lo que constituiría el Frente de Pareto para ese conjunto de soluciones; es decir, se tomó un subconjunto de todas las posibles soluciones de acuerdo a lo que se delimitó como número de corridas, puesto que para el problema en cuestión, dicho conjunto es muy grande, lo que dificulta la evaluación de todas las posibles soluciones (y obtener a su vez un frente de Pareto mucho más completo); sin embargo, los resultados de esta investigación y el análisis de los mismos, garantizan la calidad de las soluciones obtenidas para el conjunto de instancias estudiado.

2.3. Heurísticas y Metaheurísticas

Un campo muy amplio de la Investigación de Operaciones, son las técnicas de programación matemática, que comprenden entre otras: la programación lineal, entera, mixta, la Teoría de Grafos (Teoría Combinatoria) entre otras (Moreno, Moreno, 1999).

Muchas formulaciones de problemas reales usando estas técnicas generan modelos difíciles de resolver por los algoritmos clásicos de solución (simplex, primal-dual, transporte, etc.) con una complejidad computacional de forma no polinomial (NP). Para atender estos problemas se han desarrollado diversas heurísticas (Díaz, 1996).

Las heurísticas son métodos para resolver problemas complejos en forma aproximada. Su complejidad computacional es reducida en relación a la de los métodos exactos. Son flexibles y tienen como objetivo, encontrar soluciones de buena calidad en un tiempo computacional aceptable (Moreno, Moreno, 1999).

En los últimos años se ha producido un crecimiento importante en el desarrollo de procedimientos heurísticos para resolver problemas combinatorios. Este hecho puede ser constatado examinando el gran número de artículos en revistas de Investigación de Operaciones en los que se proponen y estudian métodos heurísticos (Díaz, 1996).

El auge que experimentan los métodos heurísticos se debe indudablemente a la necesidad de disponer de herramientas que permitan ofrecer soluciones rápidas a problemas reales. Es importante destacar el hecho de que los algoritmos heurísticos por sí solos no garantizan la optimización en la solución encontrada, aunque su propósito es encontrar una solución cercana al óptimo en un tiempo razonable. Sin embargo, la gran cantidad de publicaciones en donde problemas de gran dificultad son resueltos con gran rapidez (en muchos casos de manera óptima), avalan estos métodos (Moreno, Moreno, 1999).

A pesar de las ventajas que ofrecen los métodos heurísticos, hay que señalar que estos presentan ciertas deficiencias en el sentido de que muchas veces no consiguen superar las “trampas” de los extremos locales en los problemas de optimización. Esto quiere decir que la solución obtenida corresponde en muchos casos a óptimos locales, quedando así “atrapada” en estos puntos de la región factible culminando de esta manera el proceso de

búsqueda de más soluciones, eventualmente mejores o inclusive la óptima a nivel global. Además, la falta de una base teórica para dichos métodos, produce heurísticas muy especializadas desarrolladas para una categoría especial de problemas (Díaz, 1996).

Debido a lo anterior, se han desarrollado métodos alternativos más sofisticados tendientes a superar dicha dificultad. Adicional a esto, el interés primordial de los investigadores en el área ha sido el de diseñar métodos generales que sirvan para resolver clases o categorías de problemas. Dado que estos métodos generales sirven para construir y guiar el diseño de otros métodos que resuelvan problemas específicos, se les ha dado el nombre de Metaheurísticas (Moreno, Moreno, 1999).

Las metaheurísticas, llamadas también métodos inteligentes, son la reunión de los modelos rígidos de optimización con los métodos flexibles de la búsqueda heurística. Estos métodos facilitan el desarrollo de técnicas con una cierta rigidez matemática y con facilidades para incorporar nuevas soluciones, y lo más importante es que logran superar algunas de las deficiencias presentes en los métodos heurísticos (Díaz, 1996).

Necesidad de Metaheurísticas

La importancia que tienen las metaheurísticas se sintetiza en los siguientes aspectos (Díaz, 1996):

- Gran parte de los problemas en las ciencias puras y aplicadas y en la industria consiste en situaciones de búsqueda.
- Muchos de estos problemas poseen un alto grado de complejidad algorítmica (NP-Duros, NP-Completos), por lo que encontrar una solución óptima en un tiempo computacional razonable y aceptable es, en muchos casos, posible solamente para instancias (casos) pequeñas.

- A pesar del gran desarrollo teórico en el estudio de nuevos métodos para la solución de problemas complejos, este avance ha sido, en algunos casos, poco productivo en la solución de problemas reales.

Dentro de las metaheurísticas más conocidas se encuentran: Algoritmos genéticos, Recocido Simulado, Búsqueda Tabú, Algoritmos Meméticos, GRASP y Colonia de Hormigas, la cual es la metodología escogida para resolver el problema bajo estudio en el presente trabajo de grado. La descripción de esta metaheurística se realizará a continuación.

2.4. OPTIMIZACION POR COLONIA DE HORMIGAS

Ésta se basa en una técnica o metodología destinada originalmente a problemas de optimización combinatoria y basada en la teoría de optimización mediante procedimientos de aprendizaje reforzado (Paternina-Arboleda et al., 2005). Dicha metodología nació con la tesis doctoral de Marco Dorigo en el Politécnico de Milán en 1992 en su primera forma que fue el Ant System, que posteriormente sufrió algunas mejoras y adaptaciones.

El algoritmo principal en el cual se basa esta metaheurística es realmente un sistema multiagentes en el que las interacciones de bajo nivel entre agentes simples (llamados "hormigas") producen, en su conjunto, un comportamiento mucho más complejo, correspondiente a toda la colonia de hormigas (Solano-Charris, 2008). La idea se inspiró en colonia reales de hormigas. Las hormigas reales son capaces de encontrar el camino más corto entre una fuente de comida y su nido - por ejemplo - sin usar mecanismos visuales, sino sólo explotando el rastro de feromona (Dorigo, M., & Blum, C. 2005), (Dorigo, M., & Stützle, T. 2004).

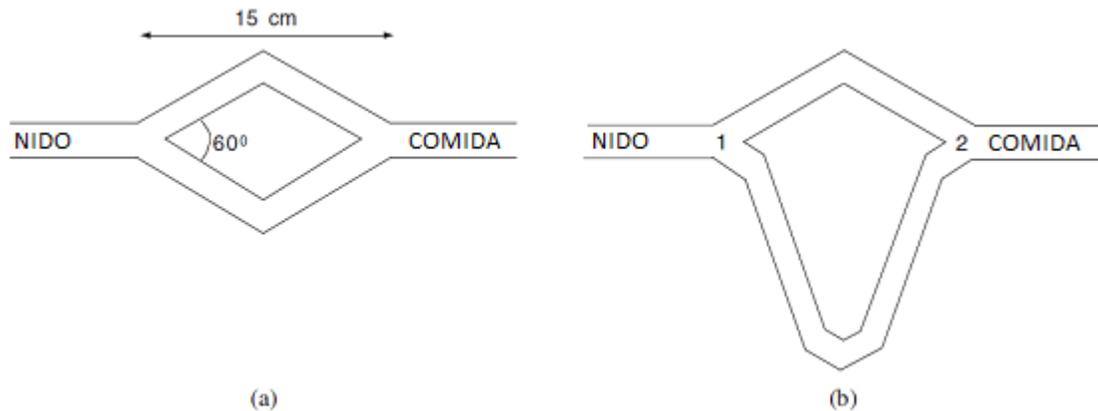


Figura 3. Experimento del doble puente realizado con el fin de determinar la ruta que escogen una colonia de hormigas al ir desde su nido hasta una fuente de comida. En (a) los dos puentes tienen la misma longitud. En (b) uno de los dos puentes es más largo que el otro Fuente: Dorigo & Stutzle. 2004.

De manera informal, un algoritmo de ACO puede ser visto como la interrelación de tres procedimientos (Dorigo y Stützle, 2004): construcción de soluciones por hormigas, actualización de la feromona y un servidor de acciones. La figura 4 muestra el pseudocódigo (genérico) de esta metaheurística.

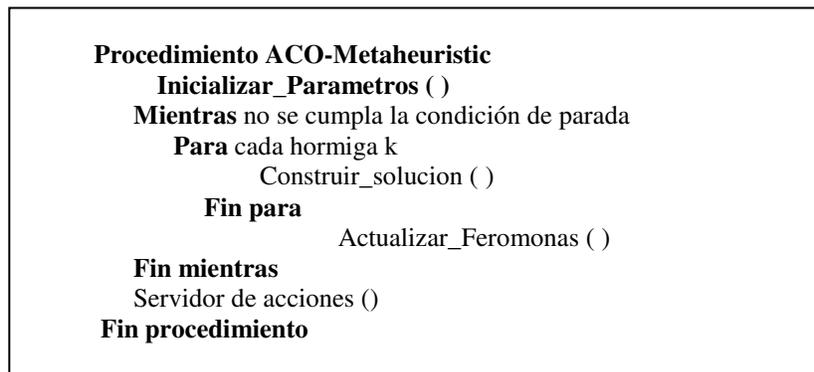


Figura 4. Pseudocódigo genérico de la metaheurística ACO. Fuente: El autor.

La construcción de soluciones administra una colonia de hormigas que visitan estados adyacentes de un problema considerado (previamente modelado). Las hormigas pueden moverse aplicando una política de decisión estocástica usando la información de los rastros

de feromona y la información heurística. De esta forma, las hormigas construyen incrementalmente una solución al problema.

La actualización de la feromona es el proceso mediante el cual los rastros de feromona son modificados. El valor del rastro puede incrementarse debido a que las hormigas depositan feromona en cada uno de los componentes o conexiones que usan para moverse de un nodo a otro del problema. Y el valor del rastro también puede disminuirse por medio de la simulación de la evaporación de feromona, lo que evita una convergencia prematura del algoritmo.

El servidor de acciones es un procedimiento utilizado para implementar acciones centralizadas las cuales no pueden ser desarrolladas por las hormigas en forma individual. Un ejemplo de estas acciones puede ser la activación de un procedimiento de optimización local o la compilación de información global que puede ser usada para tomar decisiones que modifiquen el comportamiento del algoritmo en forma general o parcial.

Desde la perspectiva de la Inteligencia Artificial, las Colonia de Hormigas son realmente técnicas de búsqueda local con registro histórico (como la búsqueda tabú) de las rutas recorridas más y/o menos promisorias (Solano-Charris, 2008). Combinada con algoritmos evolutivos y con aprendizaje mediante reforzamiento, esta técnica ha sido aplicada recientemente con gran éxito a problemas de diseño en ingeniería, a optimización combinatoria y a optimización no lineal en general. De la Cruz y Paternina-Arboleda (2004), (De la Cruz et al., (2004) y Rojas y Paternina-Arboleda (2004) han presentado varias aplicaciones en problemas de enrutamiento de vehículos con Colonia de Hormigas en problemas o instancias conocidas.

Como se mencionó en un párrafo anterior, se han propuesto varias modificaciones al procedimiento original Ant System con el fin de hacerlo más eficiente y eficaz en la

resolución de problemas cada vez más complejos; en ese sentido, se pueden mencionar enfoques como: Elitist Ant System, propuesto por Dorigo y Maniezzo (1991) y Dorigo (1992); Max-Min Ant System, desarrollado por Stützle y Hoos (1996); el Ant Colony System, propuesto por Dorigo y Gambardella (1997); y el Rank-Based Ant System, propuesto por Bullnheimer, hartl y Strauss (1999), entre los más importantes.

En este trabajo de grado se utilizó la metodología del Ant Colony System, cuya aplicación e implementación en el problema bajo estudio en la presente investigación se explicarán en el capítulo 4.

3. REVISIÓN DE LA LITERATURA

Muchos procedimientos exactos, algoritmos y heurísticas han sido propuestos para resolver el problema del flowshop flexible (Flexible Flowshop Problem, FFSP) ejemplo: Brah y Hunsucker (1991); Chen (1995); Haouari y M'Hallah (1997); Botta-Genoulaz (1997); Dessouky et al., 1998; Grangeon, Tanguy y Tchernev (1999); Genoulaz (2000); Moursli y Pochet, (2000); Yokoyama (2001); Oguz, et al., (2002); Lin y Liao (2003); Allaoui y Artiba (2004); Vob y Witt (2005); Chan, et al., (2006); Hmidal, et al., (2006); Hu, Ma, Ye (2006); Caricato, Grieco y Serino (2006); Kyparisis y Koulamas (2006); Hentous y Benhammedi (2006); Li y Li (2007); Jiang y Tang (2008); Choi, Lee (2008); Xie y Tang (2009); Yuan, Sauer, Sauvey (2009); Yang (2010). También se encuentran los trabajos de Hong et al., (2007); Paternina-Arboleda, Montoya-Torres, Herrera-Hernández, Acero-Domínguez (2008); Yaurima, Burtseva, Tchernykh (2008, 2009); Li y Zhao (2009); Wan y Choi (2009); Bozejko, Pemperaz y Smutnickix (2010). Estado del arte de este tipo de problema se han propuesto en Chen (1994); Linn y Zhang (1999) y Ruiz et al., (2010).

Del mismo modo, se han realizado desarrollos e investigaciones para resolver el problema en cuestión, que involucran el uso de metaheurísticas. Se pueden citar los trabajos de Xiao, et al., (2000); Chang, Yan y Shao (2004); Oguz y Ercan (2005); Jing, Yang e Ito (2006); Janiak, et al., (2007); Gao y Tang (2007); Urlings, Ruiz y Sivrikaya (2007, 2010); Shiau, Cheng y Huang (2008); Liu, Zou, Zhang (2008); Zhan y Qiu (2008); Liu, Cui, Li (2008); Zandieh, Mozaffari y Gholami (2009); Gholami, Zandieh y Tabriz (2009); Khalouli, Ghedjati y Hamzaoui (2009); Chang, et al., (2009); Feng, Lu, Li (2009); Su y Li (2009); Jouglet, Oguz, Sevaux (2009); Zhonghua, Haibo y Chang (2009); Amin-Naser y Beheshti-Nia (2009) y Jin y Li (2010) Kahramana, et al., (2010). Todos estos trabajos utilizaron en su mayoría Algoritmos Genéticos y Recocido Simulado, y en menor proporción Búsqueda Tabú y Colonia de Hormigas. Además, se han obtenido resultados destacados para el problema del flowshop flexible con dos estaciones.

Sin embargo, hay pocos problemas resueltos para configuraciones del Flowshop flexible con tres estaciones. Engin y Döyen (2004), Zandieh, Ghomi y Hussein (2006) y Liu, et al., (2009), proponen un enfoque novedoso al utilizar un sistema inmune artificial (cada uno con una adaptación diferente), para resolver el problema del flowshop flexible para minimizar el makespan. Estas propuestas tienen en común el hecho que se basan en una técnica algorítmica inspirada en los mecanismos de defensa de los vertebrados que involucra principios de selección clonal y maduración de la afinidad del sistema inmunológico, similar a la técnica de redes neuronales que está inspirada en el funcionamiento del sistema nervioso central. Del mismo modo, Li, et al., (2009) y Yalaoui, et al., (2010) propusieron enfoques basados en algoritmos evolutivos y lógica difusa, que simulan el comportamiento social de ciertas especies, para resolver el problema del flowshop flexible con el fin de minimizar el makespan, y el consumo de energía.

Con respecto a la optimización del flowshop flexible para más de un criterio de programación, se han realizado desarrollos de los cuales se pueden citar los trabajos presentados por Uetake et al., (1994); Nagar et al., (1995); Gupta et al., (2000); Allaoui, Artiba (2004); y Hoogoveen (2005). Del mismo modo, se encuentran también los trabajos de: Guinet y Solomon (1996), Gupta (1988), Gupta y Tung (1994); Lee y Vairaktarakis (1994), Haouari; M., & M'Hallah, R. (1997); T'kindt, V., Gupta, J. N. D., & Billaut, J. C. (2003); Janiak et al., (2005); Logendran, deSzoek y Barnard (2005); Tang, Liu, Liu (2005); Jungwattanakit et al., (2007); Mahdavi et al., (2008); Zandieh, Dorri y Khamseh (2008); Khalouli, Ghedjati y Hamzaoui (2008); Berrichi et al., (2009). Estos trabajos en su mayoría asumen, entre otros supuestos, trabajos no idénticos y máquinas paralelas idénticas.

En este recorrido, los aportes de Solano-Charris, Montoya-Torres y Paternina-Arboleda (2009) proponen un algoritmo Ant Colony para resolver el problema del flow shop flexible en una configuración de dos estaciones con el fin de optimizar dos criterios u objetivos: el

makespan y la suma de los tiempos de terminación de los trabajos. Así mismo, Montoya-Torres y Vargas-Nieto (2008) proponen un algoritmo genético para minimizar tanto el makespan y el número de trabajos tardíos en una configuración de dos etapas y en Montoya-Torres y Vargas-Nieto (2011) se retoman ciertos elementos del anterior y se presenta la herramienta propuesta en dicho trabajo.

También merecen mencionarse algunos enfoques más generales del problema del flowshop flexible con múltiples objetivos en los trabajos de: Kulcsár, Erdélyi y Hormyák (2007) quienes proponen un enfoque integrado. En éste, consideran un caso más general del flow shop flexible llamado EFFS (Extended Flexible Flow Shop) o Flow Shop Flexible extendido, el cual consta de una configuración en la cual hay máquinas paralelas no relacionadas y con diferentes capacidades, diferentes rutas o secuencias en las que se pueden realizar los trabajos. La metaheurística empleada en este caso fue Tabú Search.

Los autores Li, Xiao y Wang (2007) desarrollaron un algoritmo para resolver el problema del flowshop flexible, teniendo en cuenta la reprogramación de las operaciones en un contexto dinámico. Dugardin, Amodeo y Yalaoui (2009, 2010) presentan dos enfoques para resolver el problema del flow shop flexible, extendido al caso en el cual los trabajos o las tareas no se realizan una sola vez en cada estación, sino varias veces dependiendo de las características particulares del sistema. En ambos enfoques se usan algoritmos genéticos o evolutivos para resolver el problema. Se menciona también a Behnamian, Ghomi y Zandieh (2009, 2010) quienes proponen dos enfoques multifases para resolver el problema del flowshop flexible, el cual consiste en partir de un espacio de soluciones establecidas generalmente de manera aleatoria, las cuales se van mejorando en fases o etapas sucesivas. En este planteamiento se usa una metaheurística híbrida basada en algoritmos evolutivos y recocido simulado para resolver el problema.

Finalmente, se han realizado también investigaciones que tratan versiones del problema del flowshop flexible para situaciones particulares que involucran aplicaciones reales. Se mencionan por ejemplo los trabajos de: Jungwattanakit et al., (2005) quienes proponen un enfoque de optimización del flow shop flexible aplicado principalmente en la industria textil el cual considera una configuración de máquinas paralelas no relacionadas con diferentes tiempos de procesamiento. Aquí utilizaron la metaheurística Recocido Simulado para resolver el problema. Caricato, Grieco y Nucci (2008) propusieron un enfoque integrado entre optimización y simulación para resolver el problema del flowshop flexible para el caso real de una industria que fabrica partes mecánicas para el sector automotriz. Choi, Kim y Lee (2010) desarrollaron un algoritmo que utiliza un árbol de decisión para seleccionar la mejor regla de despacho en un flowshop flexible extendido en donde los trabajos o tareas se pueden procesar en cada estación varias veces; este caso particular se aplicó en la industria electrónica. Alfieri (2009) desarrolla un modelo de apoyo para la toma de decisiones (*Decision Support Models*, DSM) para resolver el problema del flow shop flexible aplicado a la industria del cartón. En este caso el sistema productivo constaba de múltiples estaciones en paralelo con máquinas no idénticas y en donde cada producto, dependiendo del tipo, debía ser procesado en cada estación, o sólo en algunas estaciones. La metaheurística empleada para esto, fue la Búsqueda Tabú. Khalouli, Ghedjati y Hamzaoui (2009) plantean un modelo para resolver el problema del flow shop flexible en un sistema de producción *justo a tiempo* o JIT (Just In Time por sus siglas en inglés). Este enfoque propone el uso de la metaheurística Colonia de Hormigas (Ant Colony) para resolver el problema. Li, Wang, Huo (2010) desarrollaron una heurística basada en un algoritmo genético modificado y en métodos de búsqueda local para resolver el problema del flowshop flexible de una industria metalmecánica para optimizar el proceso de tratamiento en frío y en donde la secuencia de programación dependía de los tiempos de alistamiento de las máquinas.

En conclusión, de acuerdo a la revisión de la literatura realizada, se puede decir que dentro de las técnicas metaheurísticas más empleadas para resolver el problema del flowshop flexible, se encuentran los algoritmos genéticos, seguida por recocido simulado y la búsqueda tabú. Sólo en cuatro de los numerosos casos mencionados, se hace uso de la metaheurística colonia de hormigas para optimizar el flowshop flexible, tres de ellos para enfoques multi-objetivos, en donde se minimizaron los criterios del makespan, el número de trabajos tardíos y el tiempo de flujo, entre los escogidos. Por tal razón, la metaheurística Colonia de Hormigas se escogió como la metodología a utilizar en el presente trabajo de grado, porque se considera que no se ha explotado lo suficiente su potencial para resolver el problema del flowshop flexible con múltiples objetivos de programación. Por otra parte, según lo visto en la revisión de la literatura realizada, no se encontró evidencia de que el problema planteado en el presente proyecto de grado se haya resuelto con anterioridad y menos con la metodología propuesta. Adicionalmente, hay ciertos aspectos acerca de la estructura y el funcionamiento de la metaheurística colonia de hormigas que permiten su aplicabilidad al problema en cuestión; estos aspectos se tocarán más adelante (Capítulos 3 y 4).

4. RESOLUCION DEL PROBLEMA DEL FLOWSHOP FLEXIBLE MULTI-OBJETIVO A TRAVÉS DE LA METAHEURISTICA COLONIA DE HORMIGAS

4.1. PLANTEAMIENTO DEL MODELO FORMAL.

Recordemos que esta investigación busca desarrollar una aplicación computacional de colonia de hormiga para la solución del problema $HF_k (PM^{(l)})_{l=1}^k \parallel F_l (C_{max}, \sum U_i, \sum T_i)$, definido como un flowshop flexible con k estaciones, $M(l)$ máquinas idénticas, por cada estación (etapa) l . El programa optimiza a través de metaheurísticas tres objetivos en producción: el lapso (C_{max}) de producción, el número de trabajos tardíos ($\sum U_i$) y la tardanza total ($\sum T_i$).

Se planteó un modelo formal que involucra a las funciones objetivo mencionadas y las correspondientes restricciones:

Parámetros:

n , Número de trabajos

k , Número de estaciones

$M(l)$, $l=1, \dots, k$, Número de máquinas por estación

p_{il} , $i=1, \dots, n$, $l=1, \dots, k$ el tiempo de procesamiento del trabajo i en la estación l

d_i , Fecha de entrega del trabajo i

HV , un número muy grande

Variables:

X_{ijl} , $i, j=1, \dots, n$, $u=1, \dots, M(l)$, $l=1, \dots, k$; variable binaria, igual a 1 si el trabajo i es procesado en la posición j en la máquina u en la estación l ; 0 de otra forma

C_{il} , $i=1, \dots, n$, $l=1, \dots, k$, tiempo de terminación del trabajo i en la estación l

C_{max} , Makespan

T_i tardanza del trabajo i

λ , función binaria que depende de la tardanza de los trabajos

U_i = Trabajo tardío i

Objetivos:

$$\min C_{max}$$

$$\min \sum_i T_i$$

$$\min \sum_i U_i$$

Restricciones:

$$\sum_{u=1}^{M(l)} \sum_{j=1}^n X_{ijul} = 1 \quad \forall i = 1, \dots, n; \forall l = 1, \dots, k \quad (A)$$

$$\sum_{u=1}^{M(l)} \sum_{i=1}^n X_{ijul} = 1, \quad \forall j = 1, \dots, n; \forall l = 1, \dots, k \quad (B)$$

$$\sum_{\substack{i=0 \\ i \neq l}}^n X_{ilu} - \sum_{\substack{j=0 \\ j \neq l}}^n X_{lju} = 0 \quad \forall l = 1, \dots, n; \forall u = 1, \dots, M(v);$$

$$\forall v = 1, \dots, k \quad (C)$$

$$C_{il} \geq C_{i0} + p_{il} + \left(\sum_{u=1}^{M(l)} (X_{ijul} - 1) \right) \times HV,$$

$$\forall l = 1, \dots, k; \forall i = 1, \dots, n; \forall j = 1, \dots, n \quad (D)$$

$$C_{il} \geq C_{i,l-1} + p_{il}, \quad \forall i = 1, \dots, n; \forall l = 1, \dots, k \quad (E)$$

$$C_{max} \geq C_{ik}, \quad \forall i = 1, \dots, n \quad (F)$$

$$T_i = \text{Max} \{0, C_{ik} - d_i\}, \quad \forall i = 1, \dots, n \quad (G)$$

$$\sum U_i = \sum_{i=1}^n \lambda(T_i) \quad (H)$$

$$\begin{cases} \lambda(Ti) = 1, & \text{si } Ti > 0; \\ \lambda(Ti) = 0, & \text{En caso contrario} \end{cases}$$

$$X_{ijul} \in \{0,1\}, \forall i = 1, \dots, n; \forall j = 1, \dots, n; \forall u = 1, \dots, M(l); \forall l = 1, \dots, k \quad (I)$$

$$C_{il} \geq 0, \quad \forall i = 1, \dots, n; \forall l = 1, \dots, k \quad (J)$$

$$Ti \geq 0, \quad \forall i = 1, \dots, n \quad (K)$$

El modelo 1 optimiza el siguiente problema: $HF_k (PM^{(l)})_{l=1}^k \parallel F_l (C_{max}, \sum Ti, \sum Ui)$, definido como un flowshop flexible con k estaciones, $M(l)$ máquinas idénticas, por cada estación (etapa) l , donde los objetivos en producción son: minimizar el lapso de producción (C_{max}), la tardanza total ($\sum Ti$) y el número de trabajos tardíos ($\sum Ui$).

La restricción (A) expresa que cada trabajo es procesado una sola vez en cada estación. La restricción (B) implica que cada trabajo debe estar asignado a una máquina en cada estación, independientemente de la posición de éste. La restricción (C) asegura que cada trabajo tiene un predecesor y un sucesor en la máquina donde el trabajo es procesado; es decir, si el trabajo h es procesado directamente después del trabajo i en la máquina m en la estación v , el trabajo j debe ser procesado directamente después del trabajo h en la máquina m en la estación v . Las restricciones (D) y (E) permiten calcular los tiempos de terminación en cada estación; en otras palabras, la restricción (D) implica que si el trabajo $i+1$ es procesado directamente después del trabajo i en la máquina u en la estación l , su tiempo de terminación es mayor o igual que la suma de los tiempos de terminación y los tiempos de procesamiento del trabajo i en la estación l ; análogamente, la restricción (E) implica que el tiempo de terminación del trabajo i en la estación l es mayor o igual a la suma de los tiempos de terminación del trabajo i en la estación $k-1$ y el tiempo de procesamiento del trabajo i en la estación l . Las restricciones (F) al (H) definen los criterios C_{max} , $\sum Ti$ y $\sum Ui$.

que serán minimizados. Finalmente, la restricción (I) al (K) establecen las condiciones de las variables de decisión.

El modelo formulado anteriormente surge de la necesidad de dar solución al problema mencionado y, para esto, se plantea una alternativa de solución a través de la **metaheurística Colonia de Hormigas de concepción multiobjetivo**, que permita optimizar la programación de operaciones en empresas con sistemas de producción flexibles en serie y múltiples objetivos de programación. Como se verá posteriormente, debido a que el problema bajo estudio puede modelarse a través de un grafo (condición indispensable para usar colonia de hormigas debido a la naturaleza y estructura de esta metaheurística) esta metodología es totalmente aplicable al problema en cuestión y no sólo eso, sino que ha sido utilizada con éxito en muchos de estos problemas.

4.2. EL ALGORITMO MULTI-OBJETIVO DE COLONIA DE HORMIGAS PROPUESTO

En esta investigación se propuso un algoritmo de colonia de hormigas basado en la técnica Ant Colony System (ACS). Para tratar con la configuración tipo taller flowshop flexible considerada, una serie de decisiones deben tomarse para tal efecto (Blum y Sampels, 2002). Por lo tanto, se deben definir los siguientes aspectos:

- Un modelo adecuado para la representación de feromonas
- El mecanismo de actualización de la cantidad de rastro de feromonas
- Un valor heurístico el cual puede brindar información acerca del problema en cuestión

En otras palabras, estos aspectos se utilizan para guiar la selección de una operación y, en consecuencia, influyen el desempeño del sistema. En aras de obtener soluciones

factibles, se deben respetar las restricciones de precedencia en cada paso de construcción de soluciones. Esto se asegura utilizando una lista restringida (llamada lista de candidatos) de todas las operaciones que pueden ser seleccionadas cada vez. Adicionalmente, el paso de construir soluciones debe considerar tanto la asignación de operaciones a máquinas, como la secuenciación de operaciones en las máquinas. A continuación, estos aspectos se expondrán en detalle.

4.2.1. Grafo Disyuntivo

Para aplicar la metodología de optimización por colonia de hormigas, el problema bajo estudio puede ser representado bajo la forma de un grafo disyuntivo, que es una manera de representar gráficamente los problemas de programación de operaciones. Aunque el grafo disyuntivo se ha utilizado principalmente para representar configuraciones tipo taller en las que no hay máquinas paralelas, este puede ser adaptado para representar el caso más general del flowshop flexible (Negenman, 1998). Dauzère-Pérès (1994), Dauzère-Pérès y Paulli (1994, 1997), y Nowicki and Smutnicki (1996a, b) consideraron problemas con máquinas paralelas y utilizaron este tipo de grafo.

El grafo disyuntivo $G = (V, A, E)$ para el problema del flowshop flexible se define de la siguiente manera:

- V es el conjunto de vértices o nodos, los cuales constan de todas las operaciones. El nodo O_{hj} indica la Operación h del trabajo j . Este nodo a su vez tiene un peso p_{hj} .
- A es el conjunto de arcos conjuntivos (dirigidos) los cuales representan las restricciones de precedencia entre las operaciones de un mismo trabajo. Por lo tanto, si (O_{kl}, O_{km}) forma parte del conjunto A , entonces la operación O_{kl} es la inmediata predecesora de la operación O_{km} .
- E representa el conjunto de arcos disyuntivos (no dirigidos), los cuales indican los pares de operaciones que se procesan en una misma estación.

En la figura 5 se muestra una representación de un grafo disyuntivo para un caso particular de un flowshop flexible para una configuración de tres trabajos con tres estaciones y una máquina en la estación 1 y dos máquinas para las estaciones 2 y 3; los números que se observan en los nodos se corresponden los tiempos de procesamiento de cada operación en cada estación respectivamente.

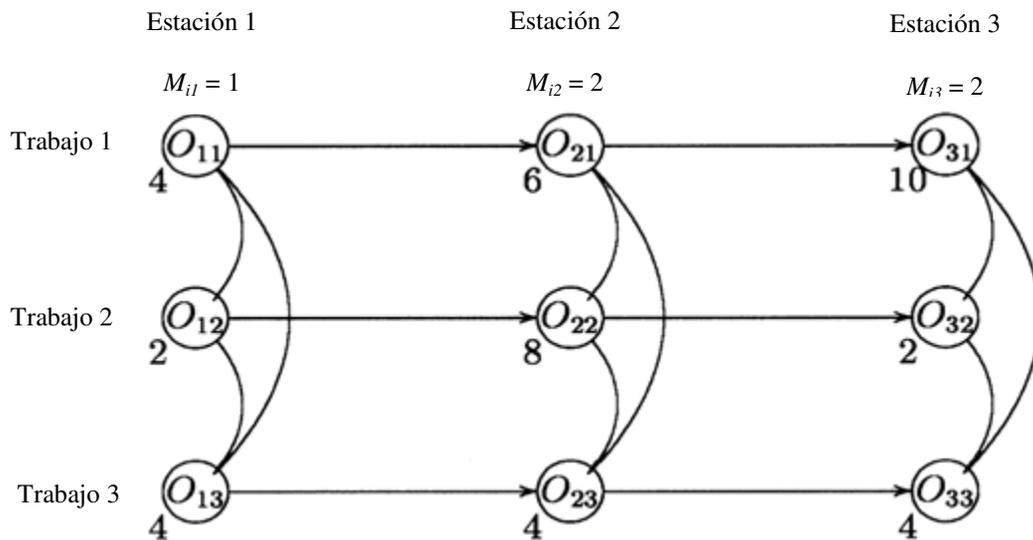


Figura 5. Representación de un flowshop flexible por medio de un grafo disyuntivo para una configuración de tres trabajos y tres estaciones. Las máquinas en paralelo no se visualizan gráficamente; sin embargo, el número de ellas se muestra para cada estación (1, 2 y 2 respectivamente). Fuente: Negenman, 2001.

Una solución factible es una selección de arcos disyuntivos del conjunto E que cumple que (i) dicha solución contiene un arco de cada par $(i, j) \leftrightarrow (k, j)$ y (ii) el grafo resultante: $G(S) = (V, A \cup S)$ es acíclico, donde S es el conjunto de arcos disyuntivos que se convirtieron en conjuntivos.

4.2.2. Procedimiento constructivo

Mientras se construye una solución factible, cada hormiga k realiza independientemente una secuencia de operaciones en el paso de construcción de soluciones. Por lo tanto, cada hormiga k tiene que tomar dos decisiones a saber: por un lado, la selección de una operación del conjunto de candidatos L_k y por el otro, la asignación de una máquina disponible para procesar la operación seleccionada. Para respetar las restricciones de precedencia en cada paso de construcción, en este enfoque se propone resolver el problema estación por estación. Por lo tanto, el procedimiento de construcción de soluciones es repetido para cada estación i hasta que todas las operaciones son seleccionadas y asignadas a una máquina elegible. Las operaciones seleccionadas se almacenan sucesivamente en una lista tabú, la cual además muestra la posición de estas en la secuencia que se está generando. Cabe resaltar que la lista tabú utilizada aquí es diferente del procedimiento de búsqueda tabú (Tabu Search) y el objetivo de dicha lista es evitar que una operación sea seleccionada dos veces para la misma secuencia.

En consonancia con lo anterior, la estructura del algoritmo multiobjetivo propuesto queda de la siguiente manera:

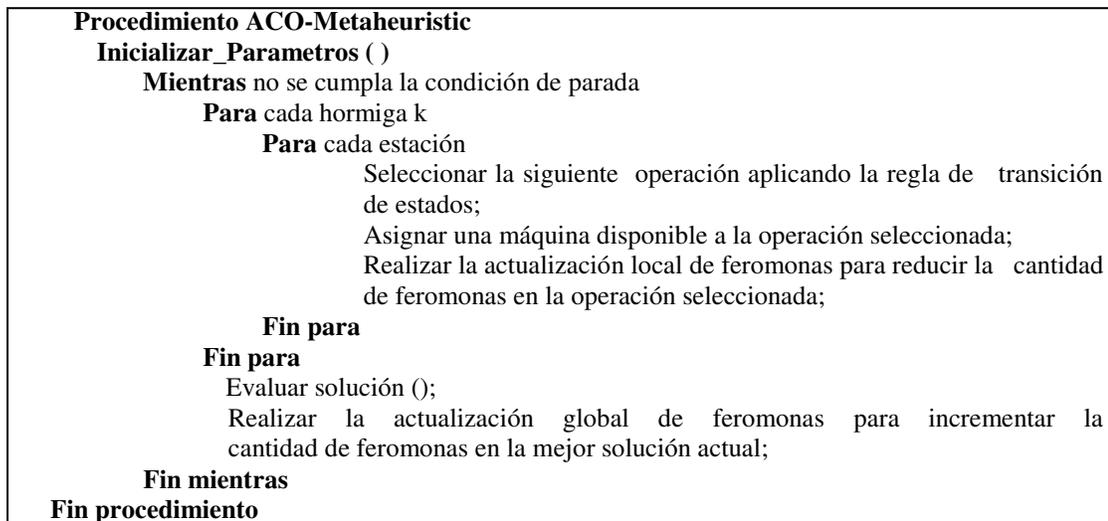


Figura 6. Algoritmo propuesto para resolver el problema del flowshop flexible mediante la metaheurística colonia de hormigas. Fuente: El autor.

4.2.3. Selección de operaciones

Para seleccionar una operación del conjunto L_k (que indica los nodos que faltan por ser visitados por la hormiga k) una regla de transición de estados, llamada regla aleatoria proporcional, es usada. Esta última suministra la probabilidad con que una hormiga k escoge una operación, basada en el rastro de feromonas τ y la información heurística η . Esta regla está dada formalmente por la siguiente expresión:

$$s = \begin{cases} \arg \max \{ |\tau(r, u)|^\alpha \cdot |\eta(r, u)|^\beta \}, & \text{si } q \leq q_0. \\ u \in L_k(r) \\ S, & \text{de otra manera} \end{cases}$$

S es una variable aleatoria escogida según la siguiente distribución de probabilidad (Dorigo y Gambardella, 1997)

$$P_k(O_{ru}, S) = \frac{|\tau(r, u)|^\alpha \cdot |\eta(r, u)|^\beta}{\sum_{u \in L_k(r)} |\tau(r, u)|^\alpha \cdot |\eta(r, u)|^\beta} \quad (3)$$

Donde α y β indican la importancia relativa de los valores $\tau(r, u)$ y $\eta(r, u)$ en la regla respectivamente; q , $0 \leq q \leq 1$, es un valor escogido aleatoriamente con probabilidad uniforme, y q_0 , $0 \leq q_0 \leq 1$, es un parámetro de selección el cual determina la importancia relativa entre la intensificación y la diversificación. Es decir, si $q > q_0$, el sistema tenderá hacia la diversificación, es decir que se explorarán diferentes soluciones en el espacio de búsqueda. Por otro lado, si $q < q_0$, el sistema tenderá hacia la intensificación, lo que quiere decir que el algoritmo explotará más información recolectada por el sistema.

4.2.4. Modelo de representación de feromonas: Como se mencionó previamente, se debe definir un adecuado modelo para representar el nivel de feromonas. En esencia, el nivel de feromonas establece la deseabilidad de que dos operaciones estén juntas una después de la

otra. Es decir, $\tau(O_{ij}, O_{hl})$ representa la conveniencia de colocar la operación O_{hl} después de la operación O_{ij} . Por lo tanto, el nivel de feromonas establece el orden relativo en que se secuenciarán las diferentes operaciones para cada estación considerada. También se utiliza para representar la experiencia pasada de las hormigas con respecto a la selección de una operación de la lista de candidatos.

4.2.5. Información heurística: La información Heurística $\eta(O_{ij}, O_{hl})$ proporciona información específica del problema en cuestión. Esta se utiliza para estimar la deseabilidad o conveniencia de ir de una operación O_{ij} a otra operación O_{hl} . En el caso que atañe a esta investigación, la información heurística se determina de manera diferente al problema del agente viajero, donde esta correspondía se calculaba en función de los costos asociados a las distancias entre cada una de las ciudades (el recíproco de este valor). Por lo tanto, para la primera estación, la información heurística viene dada por la distancia entre dos trabajos, la cual es proporcionada mediante la regla SPIRIT (Sequencing Problem Involving a Resolution by Integrated Taboo Search Techniques) presentada por Widmer y Hertz (1989). Esta heurística construye una solución inicial a través de un método de inserción, la cual se va mejorando mediante el uso de técnicas de búsqueda tabú. De acuerdo con esta regla, la distancia entre el trabajo i ($i=1, 2, \dots, n$) y el trabajo u ($u= i=1, 2, \dots, n$) con $i \neq u$ viene dada por la siguiente ecuación:

$$d_{iu} = \sum_{k=2}^S (m - k) * |p_{ik} - p_{u,k-1}| + p_{jm} \quad (4)$$

Donde S , es el número de estaciones y m el número de máquinas por cada estación. Por lo tanto, para el trabajo i ($i=1, 2, \dots, n$) y el trabajo u ($u= i=1, 2, \dots, n$) con $i \neq u$ la información heurística es:

$$\eta(i, u) = \frac{1}{d_{iu}} \quad (5)$$

Si bien la regla SPIRIT fue desarrollada para optimizar el makespan (como la mayoría de heurísticas), también se ha utilizado en enfoques multiobjetivos, inclusive con Colonia de Hormigas. En ese sentido, se considera más apropiado utilizar dicha heurística para la tardanza total y el número de trabajos tardíos, que las reglas de despacho EDD, WSPT, etc. utilizadas comúnmente para estos dos últimos criterios. Adicionalmente, el uso de una heurística en vez de una simple regla de despacho, proporciona mayor robustez a la herramienta desarrollada.

Por otro lado, a partir de la segunda estación, la información heurística se determina teniendo en cuenta la regla FCFS (First Come, First Served), la cual selecciona una operación en la estación h ($h \neq 1$), que pertenezca al trabajo con el menor tiempo de terminación en la estación inmediatamente anterior ($h-1$). Matemáticamente, la información heurística para este caso es equivalente a:

$$\eta(i, u) = \frac{1}{C_{h-1,u}} \quad (6)$$

Donde $C_{h-1,u}$ corresponde al tiempo de terminación del trabajo u en la estación $h-1$. Por lo tanto, como en el caso anterior, un valor pequeño de $C_{h-1,u}$ resultará en un valor mayor de $\eta(i, u)$ y en consecuencia, habrá una probabilidad mayor de escogencia para ese trabajo.

4.2.6. Selección de máquinas

Cada operación seleccionada debe ser asignada a una máquina elegible. En consecuencia, en este enfoque se propone una asignación basada en la regla FAM (First Available Machine)

4.2.7. Mecanismo de actualización de feromonas

4.2.7.1. Regla de actualización local de feromonas: Con el objetivo de no influenciar la escogencia para las otras hormigas, la regla de actualización local de feromonas es ejecutada después de que cada hormiga construye una solución; esta regla se lleva a cabo a través de un mecanismo de evaporación del nivel de feromonas de la operación seleccionada por una hormiga k . Por lo tanto, la operación seleccionada, se vuelve menos atractiva para las otras hormigas. Esto se hace para diversificar los caminos que deben recorrer las diferentes hormigas y evitar la convergencia hacia un óptimo local. Por lo tanto, la regla de actualización local se usa para modificar el nivel de feromonas en las operaciones seleccionadas. Esta modificación se hace a través de la siguiente expresión:

$$\tau(O_{ij}, O_{hl}) = (1-\rho_l) \cdot \tau(O_{ij}, O_{hl}) + \rho_l \cdot \tau_0 \quad (7)$$

Donde τ_0 es el nivel inicial de feromonas y ρ_l ($0 \leq \rho_l \leq 1$) es el parámetro de evaporación local de feromonas.

4.2.7.2. Regla de actualización global de feromonas: Una vez que todas las hormigas han finalizado sus recorridos se aplica la regla de actualización global, la cual intensifica el nivel de feromonas en los caminos que proporcionan la mejor solución, de modo que en la siguiente iteración su probabilidad de escogencia sea mucho mayor. Esta regla viene determinada por:

$$\tau(O_{ij}, O_{hl}) = (1-\rho_g) \cdot \tau(O_{ij}, O_{hl}) + \rho_g \cdot \Delta\tau(O_{ij}, O_{hl}) \quad (8)$$

$$\Delta\tau(O_{ij}, O_{hl}) = \begin{cases} (ET_b)^{-1}, & \text{si } (O_{ij}, O_{hl}) \text{ pertenece a la mejor solución} \\ 0, & \text{de otra manera} \end{cases}$$

En la anterior ecuación ET_b es la mejor solución obtenida la cual se determina multiplicando los objetivos a minimizar con sus respectivas ponderaciones. En otras palabras, ET_b es el menor valor hallado de todas las secuencias posibles determinadas por las hormigas. Adicionalmente, ρ_g ($0 \leq \rho_g \leq 1$) es el parámetro de evaporación global de feromonas.

5. EXPERIMENTACIÓN Y ANÁLISIS DE RESULTADOS

En este capítulo se incluyen el total de las pruebas que se le realizaron a la herramienta propuesta, la cual se denominó FF-MUCH (Flow-Shop Flexible Multi-objetivo Usando Colonia de Hormigas) con el fin de evaluar su desempeño. Estas pruebas se dividieron en tres etapas: en primera instancia se realizó la parametrización del algoritmo; es decir, se determinó la mejor combinación de parámetros de la metaheurística colonia de hormigas de modo que se pueda garantizar en gran medida la calidad de las soluciones encontradas. Esto se hizo para tres instancias representativas correspondientes a los tres escenarios evaluados: 20 trabajos, 50 trabajos y 100 trabajos (una instancia de cada escenario).

Posteriormente, definidos los parámetros del algoritmo, se realizaron pruebas comparativas para medir el desempeño de la herramienta propuesta para los criterios en mención, contra los resultados obtenidos mediante el software de dominio público LEKIN ® desarrollado en Stern School of Business, de la Universidad de New York en un proyecto dirigido por el profesor Michael L. Pinedo y el profesor asociado Xiuli Chao. Se utiliza esta herramienta debido a que el problema bajo estudio no se ha resuelto previamente, por lo que no se cuenta con un patrón de comparación.

Por último, se evalúa el desempeño del algoritmo en términos de convergencia de las soluciones encontradas para los criterios a optimizar, con base a diferentes números de iteraciones para los parámetros definidos en el diseño experimental. A continuación se ilustrarán estas etapas en detalle:

5.1. DISEÑO DE EXPERIMENTOS

Se desea conocer qué factores influyen en las variables de respuesta definidas. El número de hormigas se establece dependiendo el problema a solucionar y sus condiciones particulares, y por lo general es un número previamente determinado (Dorigo y Stützle, 2004). Sin embargo para probar este supuesto, se hicieron pruebas preliminares para 20, 50 y 100 hormigas, correspondientes al número total de trabajos, no encontrándose diferencias significativas en las soluciones encontradas, por lo que el número de hormigas para este caso no influía en las variables de respuesta definidas. Por lo tanto, se estableció un número de hormigas de 20 sólo por eficiencia computacional, ya que un número mayor de hormigas, consume más tiempo de cálculo. Del mismo modo, se definió un número fijo de iteraciones en 2000 debido a que para este valor el algoritmo presenta un comportamiento muy estable para dos de los tres objetivos considerados a diferencia de los otros valores considerados (ver capítulo 4). Además, como se mostró previamente en el marco teórico, el número de iteraciones ayuda a reforzar la mejor solución encontrada por las hormigas ya sea en la primera iteración u en otra en particular, de modo que en las siguientes iteraciones las hormigas tomen el camino de mejor solución, el cual viene determinado principalmente por la cantidad de feromonas presentes en él.

Por lo tanto, se definieron cinco factores como principales. A continuación el diseño experimental propuesto:

Factores y Niveles

- Parámetro alfa del algoritmo Colonia de Hormigas: con los niveles 1 y 2
- Parámetro beta del algoritmo Colonia de Hornigas: con los niveles 2 y 3
- Parámetro rho del algoritmo Colonia de Hornigas: con los niveles 0.001, 0.01 y 0.1

- Parámetro q_0 del algoritmo Colonia de hormigas: con los niveles 0.8 y $\text{Log}(\text{iter})/\text{Log}(\text{num_iter})$, donde iter y num_iter son la iteración actual y el número total de iteraciones respectivamente.
- Cantidad de feromonas: con los niveles 0.01 y 0.1

Estos factores tienen un gran impacto en el desempeño del algoritmo y, en consecuencia, en la calidad de las soluciones generadas (Dorigo y Gambardella, 1997).

Los niveles para los factores alfa, beta, rho, feromonas y q_0 se definieron según los estudios y publicaciones de Kalouli et. al (2009, 2010) los cuales se utilizaron para resolver un problema del flowshop flexible mono-objetivo para el makespan y uno multi-objetivo los criterios de la tardanza total y el adelanto. Estos dos estudios, si bien tratan dos problemas diferentes, incluyen dos de los criterios que se trabajan en la presente propuesta, aunque de manera separada. Sin embargo, sirven como punto de referencia para definir qué combinación de parámetros es la mejor para el desempeño del algoritmo propuesto, teniendo en cuenta los criterios a optimizar. Además, estos dos trabajos de manera conjunta constituyen la propuesta más cercana al problema bajo estudio con la metodología de colonia de hormigas.

Por otro lado, cabe aclarar que la instancia (conjunto de datos de un caso particular de estudio del problema en cuestión) no se incluye como factor, puesto que no es un parámetro de la metaheurística, además de que el objetivo de este diseño experimental es determinar la mejor combinación de parámetros para después utilizar ésta en los experimentos comparativos, donde sí se evalúan varios conjuntos de datos o instancias de la literatura. En ese sentido, se eligió una instancia representativa de cada escenario del problema bajo estudio (Instancias P20S2T01, P50S2T01 y PH1S2T1, págs. 63, 66 y 69) y se extrapolaron dichos resultados al conjunto de todas las instancias analizadas. Los resultados obtenidos

fueron muy similares a los obtenidos con otra instancia particular, la cual fue tomada de una tesis de maestría (ver anexo 1) y que no pertenece al conjunto de datos o escenarios considerados. En el anexo 1 se muestra en detalle el diseño experimental con los datos de esta última instancia y no de las otras consideradas como representativas para cada escenario, puesto que el análisis de estas es mucho más extenso y, como se mencionó anteriormente, los resultados obtenidos son muy similares para ambos estudios.

Del mismo modo, se definieron tres variables de respuesta, correspondientes a los criterios a optimizar. Esto es:

- Makespan
- Tardanza total
- Número de trabajos tardíos

Se realizó un diseño factorial mixto de cuatro réplicas el cual se desarrolló usando el software estadístico SPSS®. Los resultados del análisis experimental realizado mostraron que la mejor combinación de valores para los diferentes parámetros son respectivamente: $\alpha=2$, $\beta=3$, $\rho=0.01$, $\text{feromonas}=0.01$ y $q_0=0.8$. La instancia que se tomó para esta prueba (la cual también se tomó para medir y evaluar el comportamiento del algoritmo propuesto) y la estructura y desarrollo del diseño experimental realizado, junto con la comprobación de los supuestos del mismo, se muestran en detalle en el Anexo 1.

A continuación, se evalúa el comportamiento del algoritmo propuesto en términos de la convergencia de las soluciones para los criterios considerados.

5.2. COMPORTAMIENTO DEL ALGORITMO FF-MUCH PROPUESTO

En las siguientes gráficas se muestra el comportamiento del algoritmo propuesto en términos de las funciones objetivo evaluadas, con el fin de encontrar la convergencia de

estos valores con respecto al número de iteraciones. Esta evaluación se hizo para la instancia que se utilizó en el diseño experimental que permitió establecer los parámetros del algoritmo. Los datos de entrada y valores obtenidos para las diferentes corridas, que fueron un total de 50 para cada número de iteraciones considerado, se pueden observar en el Anexo 1.

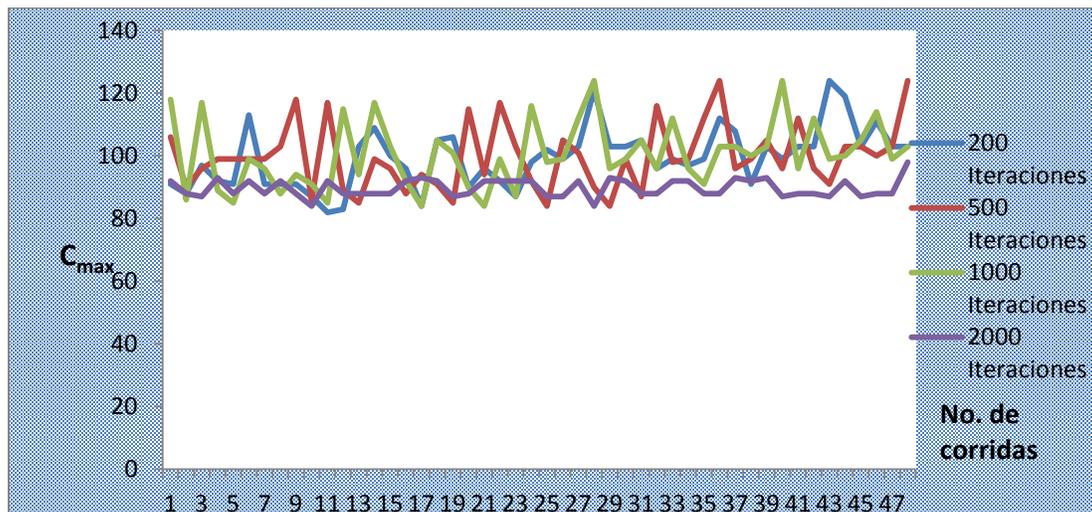


Figura 7. Medición de la convergencia del algoritmo en términos del número de iteraciones para el makespan. Fuente: El autor.

Con respecto al makespan, la gráfica muestra que el número de 2000 iteraciones proporciona una estabilidad muy marcada de este criterio para el número de iteraciones en cuestión, mientras que para el resto de número de iteraciones, hay mayor variabilidad en las soluciones encontradas. Sin embargo, no se muestra una clara convergencia del makespan para este número de iteraciones conforme aumenta el número de corridas. Adicionalmente, como se observa en la **figura 7**, los valores del makespan para 2000 iteraciones en la mayoría de los casos son mejores que para el resto de número de iteraciones, aunque para dos casos muy puntuales (corrida 11 para 200 iteraciones y corrida 25 para 500 iteraciones) el makespan obtenido fue mejor que el correspondiente para 2000 iteraciones. Sin embargo, no se podría afirmar con certeza que conforme el número de iteraciones aumenta, la calidad

de las soluciones encontradas mejora. Lo que sí muestra el anterior resultado, es que el makespan tiende a estabilizarse a medida que el número de iteraciones aumenta. A continuación se observa el comportamiento del algoritmo en términos de la tardanza para los diferentes números de iteraciones considerados:

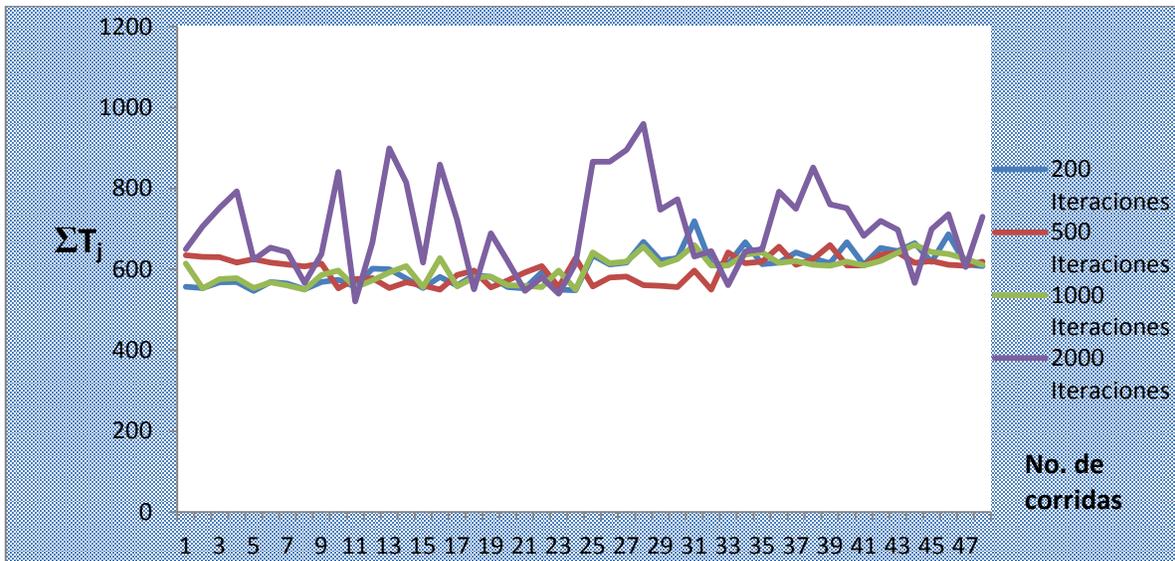


Figura 8. Medición de la convergencia del algoritmo en términos del número de iteraciones para la Tardanza total. Fuente: El autor.

La **figura 8** muestra que la tardanza total, tiene un comportamiento muy inestable para 2000 iteraciones (aunque mejora un poco la estabilidad a partir de la corrida número 31) mientras que tiene mayor estabilidad para el resto de número de iteraciones consideradas, las cuales tienen un comportamiento muy similar para las últimas 15 corridas ejecutadas. Tampoco se ve una clara convergencia para este criterio sino que por el contrario, muestra un comportamiento muy inestable conforme el número de iteraciones varía, lo cual de alguna manera corrobora lo concluido en las pruebas realizadas de que este criterio es el que peor desempeño tiene, ya que mientras el makespan y el número de trabajos tardíos tiende a estabilizarse para 2000 iteraciones, no ocurre lo mismo para la tardanza total. Esto se explica con base en la teoría de optimización multiobjetivo en la cual la mejora en un

criterio, genera una desmejora en otro, la cual se evidencia más a medida que aumenta el número de objetivos.

Por último, **la figura 9** muestra el comportamiento del algoritmo propuesto en términos del número de trabajos tardíos para los diferentes números de iteraciones considerados.

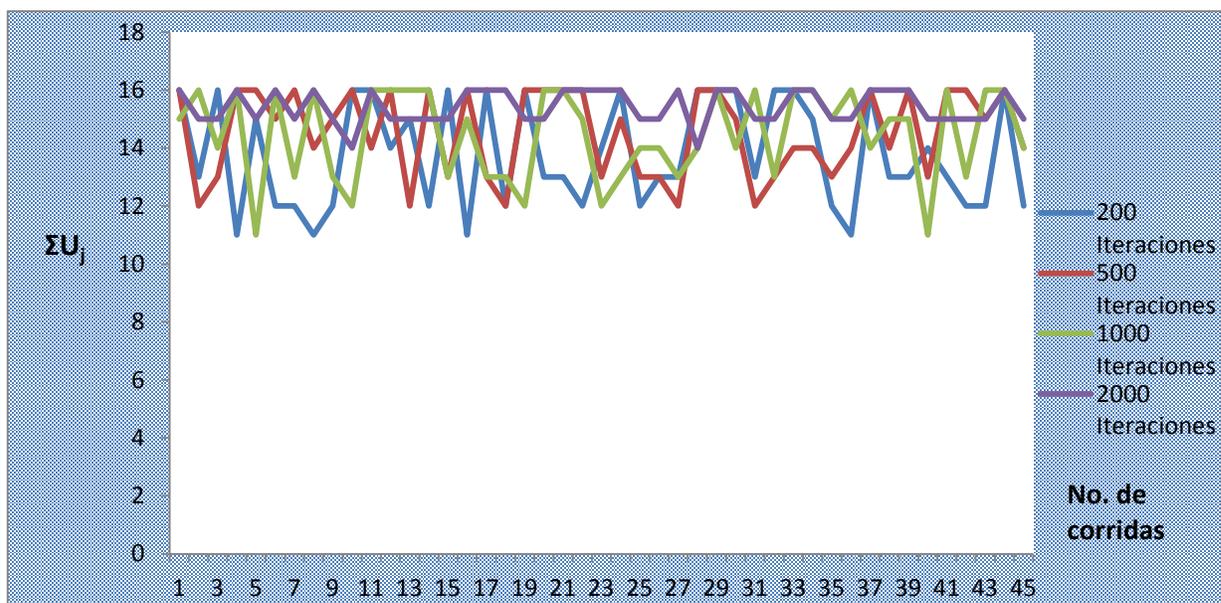


Figura 9. Medición de la convergencia del algoritmo en términos del número de iteraciones para el número de trabajos tardíos. Fuente: El autor.

En este caso, **la figura 9** muestra que el número de trabajos tardíos también presenta mayor estabilidad para 2000 iteraciones que para los demás números de iteraciones. Sin embargo, para la instancia considerada, los demás valores para los números de iteraciones generaron en casos muy puntuales mejores soluciones, especialmente para 200 iteraciones, donde el número de trabajos tardíos más pequeño fue de 11 (corridas 8, 16 y 35). También se observa en la gráfica que el mayor valor corresponde a 16, lo cual es el número total de trabajos de la instancia. Este resultado es apenas natural. Tampoco se muestra una clara

convergencia del algoritmo para este criterio, pero sí una mayor estabilidad con relación a los otros valores del número de iteraciones considerados. También se justifica lo anterior por el componente aleatorio y estocástico de la metaheurística.

En conclusión, para el makespan como para el número de trabajos tardíos, el algoritmo tiende a estabilizarse para 2000 iteraciones, no ocurre así para la tardanza total, mostrando que este criterio tiene un bajo desempeño también en cuanto a estabilidad y convergencia de las soluciones obtenidas. Sin embargo, por lo observado en las gráficas podría decirse tanto para el makespan como para el número de trabajos tardíos, que a medida que el número de iteraciones crece, se tiende a estabilizar más el algoritmo; lo que eventualmente podría conducir a un número de iteraciones mayor a 2000 en donde este convergería.

Por otra parte, los anteriores resultados también confirman la estabilidad del número de trabajos tardíos -que también se había mostrado previamente- debido a que 2000 iteraciones fue el número establecido para efectuar las pruebas computacionales comparativas con LEKIN® y Lower-Bound; del mismo modo, se confirma además que el algoritmo tiene su mejor desempeño para los criterios del makespan y el número de trabajos tardíos simultáneamente, alcanzando inclusive el valor óptimo para el primero (se desconoce para esta instancia el valor óptimo para la tardanza total y para el número de trabajos tardíos) en la instancia considerada, la cual es de dos estaciones y 16 trabajos. Para mayores detalles, remitirse al Anexo 1 para observar la estructura y composición de esta instancia.

Con base en los resultados anteriores, para la instancia considerada, un número conveniente de iteraciones sería de 2000 o más (este parámetro depende de la instancia que se está evaluando por lo que no se puede establecer un valor único para todos los casos), puesto que si bien un número de iteraciones menor eventualmente podría llegar a generar buenas - o inclusive mejores- soluciones que para 2000 iteraciones, lo que se busca también con la

herramienta computacional propuesta aparte de generar buenos programas de producción, es proporcionarle al tomador de decisiones mayor confiabilidad a la hora de programar las operaciones, por lo que se debe validar no sólo la calidad de las soluciones encontradas sino también la estabilidad de las mismas, de modo que quien toma decisiones de manera cotidiana tenga una mayor garantía de que lo que está programando, eventualmente conducirá a buenos resultados en los diferentes escenarios en que se utilice esta herramienta.

5.3. ANALISIS COMPARATIVO FF-MUCH VS. LEKIN

Una vez establecidos los parámetros de la metaheurística que proporcionan un mejor desempeño al algoritmo, se procede a realizar el análisis comparativo. Como se mencionó previamente, y debido a las características particulares del problema bajo estudio en la presente investigación, no hay un conjunto de datos para problemas del flowshop flexible con solución óptima para los tres criterios simultáneamente que se están considerando y que corresponden al makespan, tardanza total y el número de trabajos tardíos. Por lo tanto, los conjuntos de datos o instancias seleccionadas se probaron a través del software LEKIN®, los cuales son evaluados para cada objetivo por separado y luego se compararon estos valores con los obtenidos por el algoritmo propuesto en este trabajo de investigación. El análisis comparativo se realizó con base al problema planteado, hallando el error relativo (ER) entre los resultados obtenidos para el makespan, la tardanza total y el número de trabajos tardíos entre los dos métodos, como se muestra en la ecuación (9). Para las soluciones obtenidas en LEKIN®, esta herramienta permite la opción de realizar la programación mediante heurísticas conocidas o a través de reglas de despacho. Para este caso, se utilizó la heurística de secuenciación por medio de la estación cuello de botella (General SB routine). Esta heurística se ejecutó tres veces (una para cada objetivo por separado) y se comparó con la herramienta multiobjetivo propuesta.

$$ER = \frac{Valor_{FF-MUCH} - Valor_{lekin-GSB}}{Valor_{lekin-GSB}} * 100 \quad (9)$$

Por otra parte, debido a que la heurística GSB de LEKIN® no realiza la programación de instancias de 50 y 100 trabajos, se necesita de otra métrica para medir el desempeño del algoritmo propuesto para estos casos. Por lo tanto, se usó la heurística para el cálculo de la cota inferior para el makespan propuesta por Santos, et. al (1995), la cual provee un método sencillo para calcular la cota inferior (Lower Bound) del makespan en un flowshop flexible, basado en el tiempo de terminación de todos los trabajos en la estación cuello de botella. Por lo tanto, el cálculo de la cota inferior viene dado por:

$$LB(j) = \frac{1}{M(j)} \left(\sum_{y=1}^{M(j)} LSA(y, j) + \sum_{y=1}^N P(i, j) + \sum_{y=1}^{M(j)} RSA(y, j) \right) \quad (10)$$

$$LBMAX = \max_j \{LB(0), LB(j)\}$$

Donde $M(j)$ es el número de máquinas en la estación j , los términos que incluyen LSA y RSA corresponden a la suma de los $M(j)$ tiempos más cortos de procesamiento en la estación $j-1$ y $j+1$ respectivamente, y el término que contiene $P(i, j)$ es el tiempo total de procesamiento en la estación j .

En ese sentido, para el caso de 50 y 100 trabajos, el análisis comparativo se realiza con respecto al makespan únicamente, puesto que se desconocen los valores óptimos para la tardanza y el número de trabajos tardíos. Para el caso del makespan, si bien no se conocen los valores óptimos para las instancias analizadas, se realiza la comparación con respecto a la heurística para el cálculo de la cota inferior. En este caso también se calcula el error relativo porcentual para conocer la desviación del algoritmo propuesto con respecto a la

cota inferior hallada para las diferentes instancias. Por lo tanto, la expresión para la cota inferior (LB), en estos escenarios viene dada por:

$$ER = \frac{Valor_{FF-MUCH} - Valor_{LB}}{Valor_{LB}} * 100 \quad (11)$$

Selección de instancias

Las instancias seleccionadas para este análisis experimental se tomaron de OR-Library en la sección: “Multiprocessor task scheduling in multistage hybrid flowshops” (Beasley, 2010) y están disponibles en: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/multiflowinfo.htm>. Se realizaron pruebas para instancias de 20, 50 y 100 trabajos con 2, 5 y 8 estaciones. El número de máquinas por estación varía entre 1 y 5. Para cada combinación de trabajos y estaciones utilizada se probaron 10 instancias de esta librería para un total de 90 y se ejecutaron 10 réplicas del algoritmo de las cuales se obtuvieron las soluciones no dominadas o conjunto Pareto para cada una de las instancias.

Puesto que para calcular tanto la tardanza como el número de trabajos tardíos, se requiere de las fechas de entrega de los trabajos y dado que las instancias utilizadas no proveen dicha información, se utilizó la metodología propuesta por Lei & Wu (2006) para la generación de las fechas de entrega. Para problemas con 10 trabajos, las fechas de entrega de los trabajos 2 y 3, corresponden a 1.5 veces el tiempo total de procesamiento del trabajo correspondiente; la fecha de entrega del trabajo 10 es igual a su tiempo total de procesamiento y para el resto de trabajos, la fecha de entrega es 2 veces sus tiempos de procesamiento. Para instancias de 20 trabajos, la fecha de entrega para los trabajos 2, 3 y 11 son 1.5 veces el correspondiente tiempo de procesamiento; la fecha de entrega del trabajo 20 es igual a su total tiempo de procesamiento y para los otros trabajos, la fecha de entrega

corresponde a dos veces el total de sus tiempos de procesamiento. Esta metodología se extendió para instancias para 50 y 100 trabajos.

El algoritmo propuesto fue programado en Visual Basic 6.0 ® y las pruebas fueron realizadas en un computador con procesador Intel Pentium (R) Dual Core de 2.40 GHz y 4.0 GB de memoria RAM.

En las tablas que se muestran a continuación se presentan los datos generados en las corridas del algoritmo FF-MUCH para cada una de las instancias en los tres escenarios estudiados (20, 50 y 100 trabajos con 2, 5 y 8 estaciones), adicionalmente se muestra el valor mínimo, máximo y el promedio para cada una estas instancias, así como también los resultados obtenidos por General SB routine de LEKIN® (para 20 trabajos) y Lower Bound (para 50 y 100 trabajos) junto con la diferencia o error relativo porcentual del algoritmo FF-MUCH con respecto a estas heurísticas; del mismo modo, en la última columna de cada tabla se muestra el conjunto Pareto de soluciones no dominadas, las cuales aparecen sombreadas y en negrilla, para cada una de las instancias evaluadas.

INSTANCIA	F. OBJETIVO	REPLICA 1	REPLICA 2	REPLICA 3	REPLICA 4	REPLICA 5	REPLICA 6	REPLICA 7	REPLICA 8	REPLICA 9	REPLICA 10	MIN.	MAX.	PROM.	VALOR LEKIN HEURISTICA SB	ERROR REL. FF-MUCH VS. LEKIN HEURISTICA SB	SOLUCIONES NO DOMINADAS
P20S2T01	Cmax	445	452	424	426	482	408	424	452	424	424	408	482	437	413	-1%	REPLICA 6, REPLICA 7
	ΣT_j	2837	2825	2729	2776	2909	2729	2721	2825	2729	2729	2721	2909	2781	2756	-1%	
	ΣU_j	13	12	13	13	14	12	12	12	13	13	12	14	13	17	-29%	
P20S2T02	Cmax	359	353	330	353	351	325	325	325	362	353	325	362	344	358	-9%	REPLICA 5, REPLICA 7
	ΣT_j	1067	1046	1029	982	982	1046	1050	1050	1023	1046	982	1067	1033	933	5%	
	ΣU_j	10	11	11	11	10	10	10	11	10	11	10	11	11	9	11%	
P20S2T03	Cmax	987	1000	1000	980	942	985	1000	980	996	1000	942	1000	987	1066	-12%	REPLICA 3, REPLICA 5
	ΣT_j	8515	8549	8409	8435	8435	8884	8550	8435	8489	8549	8409	8884	8525	4901	72%	
	ΣU_j	17	17	16	17	16	17	17	17	17	17	16	17	17	17	-6%	
P20S2T04	Cmax	394	381	391	385	388	387	394	388	390	393	381	394	390	431	-12%	REPLICA 2, REPLICA 3
	ΣT_j	2239	2233	2206	2219	2221	2222	2230	2230	2247	2244	2206	2247	2230	1112	98%	
	ΣU_j	12	12	12	13	12	13	13	13	12	12	12	13	13	12	0%	
P20S2T05	Cmax	879	863	868	865	873	868	864	860	873	872	860	879	869	768	12%	REPLICA 2, REPLICA 8
	ΣT_j	6814	6753	6670	6753	6670	6670	6670	6670	6670	6670	6670	6814	6701	4510	48%	
	ΣU_j	17	16	17	17	17	16	17	16	17	17	16	17	17	16	0%	
P20S2T06	Cmax	393	382	393	395	394	382	392	381	392	386	381	395	389	449	-15%	REPLICA 8, REPLICA 9
	ΣT_j	2128	2144	2117	2131	2111	2134	2113	2117	2104	2155	2104	2155	2126	2102	0%	
	ΣU_j	14	13	14	14	14	13	13	13	13	13	13	14	14	20	-35%	
P20S2T07	Cmax	553	550	555	557	484	553	550	544	553	557	484	557	546	628	-23%	REPLICA 5, REPLICA 6
	ΣT_j	2965	2972	2980	2958	3082	2953	2972	2980	2960	2958	2953	3082	2978	3586	-18%	
	ΣU_j	13	12	13	12	12	12	12	12	13	12	12	13	13	20	-40%	
P20S2T08	Cmax	415	406	398	400	403	398	405	409	415	406	398	415	406	468	-15%	REPLICA 3
	ΣT_j	2140	2165	2064	2064	2099	2087	2100	2102	2140	2165	2064	2165	2113	2759	-25%	
	ΣU_j	12	12	12	13	12	13	12	12	13	12	12	13	13	20	-40%	
P20S2T09	Cmax	1368	1351	1371	1391	1378	1359	1350	1396	1361	1356	1350	1396	1369	1409	-4%	REPLICA 7
	ΣT_j	8187	8187	8319	8187	8187	8319	8187	8319	8319	8187	8187	8319	8240	10128	-19%	
	ΣU_j	14	14	16	14	14	16	14	16	16	14	14	16	15	20	-30%	
P20S2T10	Cmax	749	776	750	749	755	776	749	743	754	749	743	776	755	697	7%	REPLICA 3, REPLICA 8
	ΣT_j	4703	4805	4690	4703	4744	4805	4703	4790	4710	4703	4690	4805	4736	4631	1%	
	ΣU_j	17	16	16	18	16	17	18	16	17	16	16	18	17	20	-20%	

Tabla 1. Diferencia porcentual entre LEKIN® y FF-MUCH para 10 instancias de 20 trabajos y 2 estaciones. Fuente: El autor.

INSTANCIA	F. OBJETIVO	REPLICA 1	REPLICA 2	REPLICA 3	REPLICA 4	REPLICA 5	REPLICA 6	REPLICA 7	REPLICA 8	REPLICA 9	REPLICA 10	MIN.	MAX.	PROM.	VALOR LEKIN HEURISTICA SB	ERROR REL. FF-MUCH VS. LEKIN HEURISTICA SB	SOLUCIONES NO DOMINADAS
P20S5T01	Cmax	1644	1654	1658	1646	1658	1612	1666	1664	1644	1643	1612	1666	1649	1206	34%	REPLICA 6, REPLICA 8
	ΣTj	13147	13145	13275	13160	13275	13271	13217	13127	13133	13147	13127	13275	13190	7380	78%	
	ΣUj	16	17	16	17	17	16	16	16	16	17	16	17	17	17	-6%	
P20S5T02	Cmax	1486	1477	1460	1460	1460	1477	1483	1483	1477	1465	1460	1486	1473	1093	34%	REPLICA 5
	ΣTj	9478	9503	9508	9508	9340	9503	9340	9340	9503	9525	9340	9525	9455	5851	60%	
	ΣUj	15	14	15	14	14	14	15	14	15	14	14	15	15	18	-22%	
P20S5T03	Cmax	1296	1289	1232	1296	1289	1250	1289	1252	1291	1289	1232	1296	1278	721	71%	REPLICA 3, REPLICA 7
	ΣTj	8390	8363	8391	8390	8390	8363	8338	8363	8396	8391	8338	8396	8378	4645	80%	
	ΣUj	14	14	13	14	14	14	13	14	13	14	13	14	14	13	0%	
P20S5T04	Cmax	839	845	854	847	845	855	845	836	845	836	836	855	845	888	-6%	REPLICA 8
	ΣTj	4169	4240	4240	4311	4311	4311	4240	4069	4240	4069	4069	4311	4220	4240	-4%	
	ΣUj	14	14	14	14	13	13	14	13	14	14	13	14	14	15	-13%	
P20S5T05	Cmax	1229	1357	1333	1347	1367	1344	1381	1345	1366	1355	1229	1381	1343	1120	10%	REPLICA 1, REPLICA 5
	ΣTj	8661	8709	8708	8672	8661	8689	8705	8706	8661	8700	8661	8709	8688	7186	21%	
	ΣUj	14	13	14	13	13	13	14	13	13	13	13	14	14	12	8%	
P20S5T06	Cmax	1847	1724	1785	1728	1800	1707	1707	1776	1750	1756	1707	1847	1758	1321	29%	REPLICA 5, REPLICA 6
	ΣTj	12692	12671	12684	12707	12653	12692	12728	12740	12741	12724	12653	12741	12704	8907	42%	
	ΣUj	16	17	16	17	16	16	17	17	17	17	16	17	17	20	-20%	
P20S5T07	Cmax	1694	1671	1693	1649	1601	1690	1636	1658	1626	1884	1601	1884	1681	1215	32%	REPLICA 3, REPLICA 5
	ΣTj	14340	14161	14119	14200	14309	14213	14211	14190	14332	14309	14119	14340	14239	7429	90%	
	ΣUj	17	16	16	16	16	16	17	16	17	16	16	17	17	19	-16%	
P20S5T08	Cmax	1196	1143	1194	1153	1143	1200	1143	1204	1194	1194	1143	1204	1177	899	27%	REPLICA 2, REPLICA 3
	ΣTj	7390	7057	7050	7057	7133	7112	7057	7120	7123	7050	7050	7390	7115	4416	60%	
	ΣUj	14	13	13	14	13	14	13	14	13	13	13	14	14	17	-24%	
P20S5T09	Cmax	1347	1341	1319	1400	1391	1318	1345	1384	1356	1356	1318	1400	1356	1531	-14%	REPLICA 6
	ΣTj	8187	8187	8435	8319	8199	8187	8453	8187	8319	8200	8187	8453	8268	11162	-27%	
	ΣUj	14	14	16	16	14	14	14	14	16	14	14	16	15	19	-26%	
P20S5T10	Cmax	1486	1483	1483	1490	1485	1486	1486	1483	1484	1485	1483	1490	1486	1185	25%	REPLICA 2, REPLICA 7, REPLICA 8
	ΣTj	9478	9340	9350	9340	9347	9478	9312	9340	9350	9347	9312	9478	9369	6784	37%	
	ΣUj	14	13	14	13	14	14	13	13	14	14	13	14	14	20	-35%	

Tabla 2. Diferencia porcentual entre LEKIN® y FF-MUCH para 10 instancias de 20 trabajos y 5 estaciones. Fuente: El autor.

INSTANCIA	F. OBJETIVO	REPLICA 1	REPLICA 2	REPLICA 3	REPLICA 4	REPLICA 5	REPLICA 6	REPLICA 7	REPLICA 8	REPLICA 9	REPLICA 10	MIN.	MAX.	PROM.	VALOR LEKIN HEURISTICA SB	ERROR REL. FF-MUCH VS. LEKIN HEURISTICA SB	SOLUCIONES NO DOMINADAS
P20S8T01	Cmax	2105	2185	2111	2185	2311	2141	2114	2148	2141	2105	2105	2311	2155	1405	50%	REPLICA 2, REPLICA 10
	ΣT_j	28311	21845	21941	21845	21941	21935	21929	22031	21935	28311	21845	28311	23203	12288	78%	
	ΣU_j	17	16	17	17	17	17	17	17	17	16	16	17	17	17	-6%	
P20S8T02	Cmax	2355	2371	2374	2345	2378	2362	2342	2333	2365	2351	2333	2378	2358	1493	56%	REPLICA 1, REPLICA 8
	ΣT_j	17256	17282	17716	17733	17778	17765	17716	17733	17282	17282	17256	17778	17555	12117	42%	
	ΣU_j	16	17	17	18	16	17	17	16	18	17	16	18	17	12	33%	
P20S8T03	Cmax	1887	1877	1895	1898	1896	1899	1884	1895	1889	1890	1877	1899	1891	1166	61%	REPLICA 2, REPLICA 10
	ΣT_j	12600	12642	12635	12642	12635	12596	12645	12642	12642	12596	12596	12645	12628	2951	327%	
	ΣU_j	16	15	15	16	15	16	16	15	15	15	15	16	16	13	15%	
P20S8T04	Cmax	1869	1754	1875	1831	1797	1745	1726	1900	1895	1713	1713	1900	1811	866	98%	REPLICA 2, REPLICA 10
	ΣT_j	9721	9712	9783	9804	9715	9736	9777	9808	9789	9730	9712	9808	9758	6721	45%	
	ΣU_j	13	13	14	13	15	13	14	13	14	13	13	15	14	7	86%	
P20S8T05	Cmax	2041	2087	2186	2040	2123	2175	2164	2135	2056	2156	2040	2186	2117	1174	74%	REPLICA 4
	ΣT_j	16671	16547	16721	16498	16621	16547	16721	16731	16671	16728	16498	16731	16646	10123	63%	
	ΣU_j	16	15	15	15	16	15	16	15	16	15	15	16	16	10	50%	
P20S8T06	Cmax	3141	3111	3148	3111	3114	3185	3141	3148	3114	3111	3111	3185	3133	2320	34%	REPLICA 2, REPLICA 6, REPLICA 10
	ΣT_j	21935	21845	22031	21941	22064	21845	21935	22031	21935	21941	21845	22064	21951	22169	-1%	
	ΣU_j	17	17	17	17	17	15	17	17	17	15	15	17	17	20	-25%	
P20S8T07	Cmax	1834	1838	1801	1814	1866	1834	1830	1801	1866	1794	1794	1866	1828	1642	9%	REPLICA 7, REPLICA 10
	ΣT_j	14448	14500	14661	14671	14675	14448	14429	14661	14675	14671	14429	14675	14584	14799	-3%	
	ΣU_j	15	16	15	15	19	19	15	15	15	15	15	19	16	20	-25%	
P20S8T08	Cmax	2592	2740	2592	2494	2749	2598	2767	2702	2740	2767	2494	2767	2675	2227	12%	REPLICA 4
	ΣT_j	19773	19814	19773	19715	19733	19801	19816	19801	19814	19816	19715	19816	19786	18921	4%	
	ΣU_j	17	16	16	16	16	17	16	18	16	16	16	18	17	20	-20%	
P20S8T09	Cmax	2307	2463	2544	2561	2451	2441	2551	2543	2502	2467	2307	2561	2483	2325	-1%	REPLICA 1, REPLICA 5
	ΣT_j	26232	26271	25564	26207	25526	26271	26232	26207	25564	25526	25526	26271	25960	21445	19%	
	ΣU_j	17	19	18	19	17	19	17	19	19	19	19	19	19	20	-5%	
P20S8T10	Cmax	2090	2081	2084	2090	2099	2099	2084	2081	2099	2099	2081	2099	2091	2082	0%	REPLICA 2, REPLICA 4
	ΣT_j	12599	12648	12635	12596	12642	12646	12635	12648	12642	12596	12596	12648	12629	16363	-23%	
	ΣU_j	16	15	15	15	16	15	16	15	16	16	15	16	16	20	-25%	

Tabla 3. Diferencia porcentual entre LEKIN® y FF-MUCH para 10 instancias de 20 trabajos y 8 estaciones. Fuente: El autor.

INSTANCIA	F. OBJETIVO	REPLICA 1	REPLICA 2	REPLICA 3	REPLICA 4	REPLICA 5	REPLICA 6	REPLICA 7	REPLICA 8	REPLICA 9	REPLICA 10	MIN.	MAX.	PROM.	VALOR LOWER BOUND	ERROR REL. FF-MUCH VS. LOWER BOUND	SOLUCIONES NO DOMINADAS
P50S2T01	Cmax	1037	1046	1097	1123	1067	1016	1065	1057	1037	1067	1016	1123	1070	894	14%	REPLICA 2,
	ΣT_j	21356	21093	21086	21407	21240	21504	21310	21418	21356	21240	21086	21504	21295			REPLICA 3,
	ΣU_j	38	38	38	39	38	38	38	38	38	38	38	39	39			REPLICA 6
P50S2T02	Cmax	2491	2483	2497	2496	2502	2490	2534	2495	2500	2494	2497	2534	2516	2483	1%	REPLICA 1,
	ΣT_j	50753	50841	51609	51100	51139	51097	51677	51585	51348	51520	50753	51677	51215			REPLICA 2,
	ΣU_j	42	42	42	42	42	42	42	42	42	42	42	42	42			REPLICA 6
P50S2T03	Cmax	1000	1029	985	1050	860	971	1022	1004	1004	985	860	1050	955	660	30%	REPLICA 5,
	ΣT_j	17098	16725	16641	16769	16611	16735	16581	16606	16606	16641	16581	17098	16840			REPLICA 7
	ΣU_j	17	37	37	37	37	37	37	37	37	37	17	37	27			
P50S2T04	Cmax	1016	985	991	1049	1045	982	1037	1001	985	1016	982	1049	1016	964	2%	REPLICA 2,
	ΣT_j	17857	17853	17908	17942	17807	17889	17883	17940	17853	17857	17807	17942	17875			REPLICA 5,
	ΣU_j	37	36	37	37	36	36	37	37	36	37	36	37	37			REPLICA 6
P50S2T05	Cmax	1063	1073	1077	1064	1064	1074	1063	1073	1064	1077	1063	1077	1070	1046	2%	REPLICA 1,
	ΣT_j	17493	18395	18478	17493	18399	18601	17493	18395	18399	18478	17493	18601	18047			REPLICA 7
	ΣU_j	38	39	39	38	39	39	38	39	39	39	38	39	39			
P50S2T06	Cmax	3282	3251	3251	3247	3259	3286	3275	3251	3282	3251	3247	3286	3267	3240	0%	REPLICA 1,
	ΣT_j	71763	72951	73055	72777	72871	72970	72994	72955	71763	72951	71763	73055	72409			REPLICA 4
	ΣU_j	47	47	47	47	47	47	47	47	47	47	47	47	47			
P50S2T07	Cmax	863	938	852	863	938	871	860	938	852	871	852	938	895	846	1%	REPLICA 2,
	ΣT_j	13563	13477	13529	13563	13477	13506	13625	13477	13529	13506	13477	13625	13551			REPLICA 3
	ΣU_j	43	43	43	44	43	44	44	43	43	44	43	44	44			
P50S2T08	Cmax	2900	2985	2985	2905	2941	2897	2985	2900	2909	2985	2897	2985	2941	2887	0%	REPLICA 1,
	ΣT_j	62116	60333	62361	62276	62379	62417	62370	62116	62530	60333	60333	62530	61432			REPLICA 2,
	ΣU_j	48	48	48	48	48	48	48	48	48	48	48	48	48			REPLICA 6
P50S2T09	Cmax	2138	2138	2133	2139	2139	2138	2154	2138	2138	2139	2133	2154	2144	2130	0%	REPLICA 2,
	ΣT_j	75236	74920	75470	75715	74976	76187	75928	75236	76187	74976	74920	76187	75554			REPLICA 3
	ΣU_j	48	48	48	48	48	48	48	48	48	48	48	48	48			
P50S2T10	Cmax	1189	1192	1191	1242	1201	1212	1212	1189	1192	1201	1189	1242	1216	1181	1%	REPLICA 1,
	ΣT_j	23018	23003	23133	23143	23237	23053	23053	23018	23003	23237	23003	23237	23120			REPLICA 2
	ΣU_j	46	44	45	46	46	46	46	46	44	46	44	46	45			

Tabla 4. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 50 trabajos y 2 estaciones. Fuente: El autor.

INSTANCIA	F. OBJETIVO	REPLICA 1	REPLICA 2	REPLICA 3	REPLICA 4	REPLICA 5	REPLICA 6	REPLICA 7	REPLICA 8	REPLICA 9	REPLICA 10	MIN.	MAX.	PROM.	VALOR LOWER BOUND	ERROR REL. FF-MUCH VS. LOWER BOUND	SOLUCIONES NO DOMINADAS
P 5055T01	C_{max}	3598	3564	3554	3587	3552	3562	3550	3540	3545	3558	3540	3598	3561	2635	34%	REPLICA 8
	ΣT_j	80868	81216	81857	81390	81325	80130	81212	80130	81857	81216	80130	81857	81121			
	EUj	41	42	42	41	42	41	41	41	42	42	41	42	42			
P 5055T02	C_{max}	2475	2440	2457	2448	2435	2444	2434	2484	2450	2435	2434	2484	2451	2420	0,57%	REPLICA 1, REPLICA 7
	ΣT_j	49729	50734	50098	50734	51364	49729	50879	51147	50098	50533	49729	51364	50505			
	EUj	42	43	43	43	43	43	42	43	43	43	43	42	43	43		
P 5055T03	C_{max}	4469	4494	4475	4470	4432	4474	4465	4476	4475	4472	4432	4494	4471	2604	70%	REPLICA 5, REPLICA 7
	ΣT_j	111232	118359	109220	109317	108016	109744	107301	109872	108963	109066	107301	118359	110109			
	EUj	43	42	44	40	40	43	40	42	41	41	40	44	42			
P 5055T04	C_{max}	2291	2292	2297	2288	2280	2300	2293	2280	2285	2301	2280	2301	2291	2017	13%	REPLICA 8
	ΣT_j	49203	49221	49173	49211	49218	49187	49212	49171	49186	62624	49171	62624	50541			
	EUj	40	41	42	42	41	40	42	40	40	40	40	42	41			
P 5055T05	C_{max}	3858	4023	3952	3851	3704	4011	4035	4097	4041	4035	3704	4097	3961	2755	34%	REPLICA 4, REPLICA 5
	ΣT_j	88810	88930	88787	88348	88358	88439	88730	88095	88779	88708	88095	88930	88599			
	EUj	41	42	41	41	41	42	41	42	41	41	41	42	42	42		
P 5055T06	C_{max}	4355	4003	4159	4187	4168	4163	4180	4170	4183	4185	4003	4355	4176	2638	52%	REPLICA 2
	ΣT_j	95777	95407	95759	95740	95792	95471	95777	95471	95580	95704	95407	95792	95648			
	EUj	41	41	42	42	42	42	42	41	41	41	41	42	42			
P 5055T07	C_{max}	4722	4651	4543	4737	4781	4451	4474	4721	4783	4785	4451	4785	4665	2728	63%	REPLICA 7, REPLICA 10
	ΣT_j	105528	105540	105515	104962	105417	105540	105105	105417	105459	104852	104852	105540	105334			
	EUj	43	43	43	43	42	41	41	42	42	41	41	43	43			
P 5055T08	C_{max}	4286	4232	4052	4218	4058	4270	4296	4256	4254	4327	4052	4327	4225	2820	44%	REPLICA 3, REPLICA 5
	ΣT_j	92565	92987	92861	92579	91682	91893	91912	91955	92913	93005	91682	93005	92436			
	EUj	42	42	41	42	41	42	43	42	43	42	41	43	42			
P 5055T09	C_{max}	1653	1731	1663	1692	1657	1706	1673	1645	1667	1636	1636	1731	1673	1394	14,79%	REPLICA 3, REPLICA 10
	ΣT_j	31521	31712	30232	31687	31614	31772	31518	31785	31503	31580	30232	31785	31493			
	EUj	41	42	40	42	42	41	41	42	41	40	40	42	42			
P 5055T10	C_{max}	2805	2733	2863	2728	2600	2694	2742	2895	2783	2714	2600	2895	2756	1512	72%	REPLICA 5, REPLICA 6
	ΣT_j	54077	54682	54597	54752	54797	53992	54064	54752	54535	54429	53992	54797	54468			
	EUj	45	46	47	46	44	44	46	45	45	45	44	47	46			

Tabla 5. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 50 trabajos y 5 estaciones. Fuente: El autor.

INSTANCIA	F. OBJETIVO	REPLICA 1	REPLICA 2	REPLICA 3	REPLICA 4	REPLICA 5	REPLICA 6	REPLICA 7	REPLICA 8	REPLICA 9	REPLICA 10	MIN.	MAX.	PROM.	VALOR LEKIN REGLA EDD	ERROR REL. FF-MUCH VS. LOWER BOUND	SOLUCIONES NO DOMINADAS
P50S8T01	Cmax	5833	6174	6009	6139	6176	6172	6134	5850	5903	6189	5833	6189	6058	2326	151%	REPLICA 1, REPLICA 8
	ΣT_j	132414	132497	132774	132716	133600	133392	133561	131465	133285	133282	131465	133600	132899			
	ΣU_j	40	41	40	41	40	41	41	40	40	40	40	41	41			
P50S8T02	Cmax	5964	6035	6039	6046	6042	6048	6044	6052	6046	6039	5964	6052	6036	2764	116%	REPLICA 1
	ΣT_j	141435	142590	142286	142248	142254	141781	141781	142903	142296	142271	141435	142903	142185			
	ΣU_j	47	47	48	48	47	48	48	48	48	47	47	48	48			
P50S8T03	Cmax	5302	5151	5211	5320	5121	5316	5327	5325	5223	5320	5121	5327	5262	2670	92%	REPLICA 5, REPLICA 9
	ΣT_j	114232	114471	114702	114426	114873	114393	114595	114475	113904	114657	113904	114873	114473			
	ΣU_j	41	41	42	40	40	43	42	41	40	42	40	43	42			
P50S8T04	Cmax	4656	4571	4720	4618	4656	4650	4620	4648	4718	4625	4571	4720	4649	1522	200%	REPLICA 2, REPLICA 4
	ΣT_j	97268	97407	97315	96525	97409	97453	97810	96682	96525	97183	96525	97810	97158			
	ΣU_j	37	36	37	36	37	36	37	37	36	37	36	37	37			
P50S8T05	Cmax	6374	6368	6433	6391	6375	6393	6375	6392	6372	6445	6368	6445	6392	2638	141%	REPLICA 2, REPLICA 9
	ΣT_j	140702	142213	141109	142170	142566	142050	140803	140679	139938	142131	139938	142566	141437			
	ΣU_j	41	40	42	41	40	42	41	40	40	42	40	42	41			
P50S8T06	Cmax	5326	5526	5457	5539	5523	5621	5543	5572	5564	5566	5326	5621	5524	2560	108%	REPLICA 1, REPLICA 2
	ΣT_j	111086	110543	111933	111202	111399	111457	111113	111955	115211	111107	110543	115211	111701			
	ΣU_j	41	41	42	42	41	42	42	43	41	41	41	43	42			
P50S8T07	Cmax	5887	5880	5898	5877	5893	5838	5864	5879	5871	5847	5838	5898	5874	2615	123%	REPLICA 6, REPLICA 7
	ΣT_j	125771	125933	125989	126037	126212	126219	125739	126033	125937	126112	125739	126219	125999			
	ΣU_j	43	41	43	43	43	41	41	43	43	41	41	43	43			
P50S8T08	Cmax	5908	5896	5903	5910	5901	5894	5898	5899	5904	5895	5894	5910	5901	2712	117%	REPLICA 6, REPLICA 7, REPLICA 8
	ΣT_j	128678	128716	129699	128857	129475	129034	128471	128471	128774	129464	128471	129699	128964			
	ΣU_j	41	41	42	42	43	40	40	40	41	42	40	43	42			
P50S8T10	Cmax	8470	8486	8406	8450	8497	8450	8420	8416	8414	8408	8406	8497	8442	2793	201%	REPLICA 3, REPLICA 8
	ΣT_j	204364	204159	204807	204745	204865	204567	204715	204053	204821	204141	204053	204865	204524			
	ΣU_j	42	43	42	42	42	43	42	42	43	43	42	43	43			

Tabla 6. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 50 trabajos y 8 estaciones. Fuente: El autor.

INSTANCIA	F. OBJETIVO	REPLICA 1	REPLICA 2	REPLICA 3	REPLICA 4	REPLICA 5	REPLICA 6	REPLICA 7	REPLICA 8	REPLICA 9	REPLICA 10	MIN.	MAX.	PROM.	VALOR MAKESPAN LOWER BOUND	ERROR REL. MIN. VALOR FF- MUCH VS.	SOLUCIONES NO DOMINADAS
PH1S2T1	Cmax	1909	1910	1886	1870	1875	1867	1875	1898	1867	1933	1867	1933	1900	1346	38,71%	REPLICA 6,
	ΣTj	75480	75560	75059	75446	75208	75017	75208	74824	75059	75457	74824	75560	75192			REPLICA 8,
	ΣUj	80	79	80	80	80	79	80	79	79	80	79	80	80			REPLICA 9
PH1S2T2	Cmax	5035	4969	4976	4971	4976	5035	4969	4984	4981	5035	4969	5035	5002	4914	1,12%	REPLICA 3,
	ΣTj	239759	239853	238781	240396	238781	239759	239853	241079	240743	239759	238781	241079	239930			REPLICA 7
	ΣUj	84	84	83	84	83	84	83	84	84	84	83	84	84			
PH1S2T3	Cmax	2294	2274	2298	2322	2274	2291	2221	2280	2322	2298	2221	2322	2272	1723	28,90%	REPLICA 6,
	ΣTj	91117	90989	90989	91096	90964	90928	91091	91015	90992	90989	90928	91117	91023			REPLICA 7
	ΣUj	80	80	79	80	80	79	79	81	79	79	79	81	80			
PH1S2T4	Cmax	4954	4965	4959	4942	4973	4954	4973	4965	4974	4973	4942	4974	4958	4929	0,26%	REPLICA 3,
	ΣTj	247462	247434	245889	246720	246925	247462	246925	247434	247743	246925	245889	247743	246816			REPLICA 4
	ΣUj	84	84	83	83	84	85	84	86	84	84	83	86	85			
PH1S2T5	Cmax	6371	6361	6366	6352	6357	6342	6357	6355	6361	6342	6342	6371	6357	5094	24,50%	REPLICA 4,
	ΣTj	197917	200659	200618	197917	197917	197917	200139	198892	198497	200201	197917	200659	199288			REPLICA 5,
	ΣUj	84	84	84	83	83	83	84	84	84	84	83	84	84			REPLICA 6
PH1S2T6	Cmax	5371	5356	5327	5371	5375	5367	5371	5367	5345	5345	5327	5375	5351	5265	1,18%	REPLICA 3,
	ΣTj	213851	214667	214653	213851	213851	214415	213851	214415	214636	214636	213851	214667	214259			REPLICA 4
	ΣUj	85	86	83	83	84	83	84	84	83	83	83	86	85			
PH1S2T7	Cmax	1856	1869	1883	1856	1864	1883	1856	1883	1945	1864	1856	1945	1901	1730	7,28%	REPLICA 3
	ΣTj	81060	81610	81533	81060	81713	81533	81060	81533	81575	81713	81060	81713	81387			
	ΣUj	78	78	78	78	78	78	78	78	78	78	78	78	78			
PH1S2T8	Cmax	5035	5035	4976	4971	4969	4971	4969	4976	4981	5035	4969	5035	5002	4914	1,12%	REPLICA 3,
	ΣTj	229119	229119	228184	229728	229728	229728	229728	230122	2300554	229119	228184	2300554	1264369			REPLICA 5
	ΣUj	84	84	83	83	83	84	84	83	84	84	83	84	84			
PH1S2T9	Cmax	2864	2933	2959	2970	2894	2824	2945	2864	2897	2864	2824	2970	2897	2568	9,97%	REPLICA 6,
	ΣTj	133317	134857	134657	134657	135849	134118	135328	133317	135317	133317	133317	135849	134583			REPLICA 8
	ΣUj	81	81	82	82	83	81	83	81	81	81	81	83	82			
PH1S2T10	Cmax	3123	3187	3109	3186	3156	3116	3170	3186	3116	3120	3109	3187	3148	2668	16,53%	REPLICA 3,
	ΣTj	110963	110047	111204	109813	109741	111386	109744	109813	111123	110966	109741	111386	110564			REPLICA 5
	ΣUj	83	82	82	83	82	83	83	83	83	83	82	83	83			

Tabla 7. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 100 trabajos y 2 estaciones. Fuente: El autor.

INSTANCIA	F. OBJETIVO	REPLICA 1	REPLICA 2	REPLICA 3	REPLICA 4	REPLICA 5	REPLICA 6	REPLICA 7	REPLICA 8	REPLICA 9	REPLICA 10	MIN.	MAX.	PROM.	VALOR MAKESPAN LOWER BOUND	ERROR REL. MIN. VALOR FF- MUCH VS.	SOLUCIONES NO DOMINADAS
PH1S5T1	Cmax	9401	9343	9413	9379	9460	9343	9377	9413	9392	9343	9343	9460	9402	5353	74,54%	REPLICA 10
	ΣTj	467162	466996	467002	467329	467296	466996	467365	467002	467567	466996	466996	467567	467282			
	ΣUj	84	84	83	84	83	84	84	83	84	83	84	83	84			
PH1S5T2	Cmax	7364	7364	7307	7404	7404	7305	7404	7364	7303	7333	7303	7404	7354	5513	32,47%	REPLICA 6, REPLICA 9
	ΣTj	345328	345328	344969	344986	344986	344806	344986	345328	345038	345347	344806	345347	345077			
	ΣUj	84	84	84	84	84	83	84	84	83	84	83	84	84			
PH1S5T3	Cmax	5631	5602	5594	5599	5602	5566	5599	5631	5599	5588	5566	5631	5599	5370	3,65%	REPLICA 6
	ΣTj	248332	248449	248867	248614	248449	248332	248614	248332	248614	248762	248332	248867	248600			
	ΣUj	81	81	82	80	83	80	80	81	80	82	80	83	82			
PH1S5T4	Cmax	5570	5581	5580	5526	5581	5614	5579	5580	5583	5570	5526	5614	5570	4520	22,26%	REPLICA 4
	ΣTj	249594	249594	250219	249265	249781	249484	250395	250219	250446	249594	249265	250446	249856			
	ΣUj	82	82	82	82	82	82	82	82	82	82	82	82	82			
PH1S5T5	Cmax	7523	7523	7631	7523	7523	7631	7523	7599	7631	7488	7488	7631	7560	5280	41,82%	REPLICA 4, REPLICA 10
	ΣTj	352042	351902	352265	351902	352042	352316	352027	352158	352316	352330	351902	352330	352116			
	ΣUj	85	84	84	84	84	86	85	84	84	84	84	86	85			
PH1S5T6	Cmax	5833	5854	5977	5833	5871	5833	5928	5971	5854	5943	5833	5977	5905	5460	6,83%	REPLICA 2, REPLICA 9
	ΣTj	282069	281664	282249	282069	281771	282069	282339	282674	281664	282437	281664	282674	282169			
	ΣUj	83	81	82	83	82	83	82	82	81	81	81	83	82			
PH1S5T7	Cmax	7570	7711	7589	7587	7746	7649	7565	7586	7611	7627	7565	7746	7656	5001	51,27%	REPLICA 4, REPLICA 7
	ΣTj	379312	378371	377598	376884	379068	379156	377141	377071	378719	378513	376884	379312	378098			
	ΣUj	85	85	84	83	85	85	83	84	84	84	83	85	84			
PH1S5T8	Cmax	9455	9455	9379	9413	9455	9401	9343	9455	9402	9343	9343	9455	9399	5070	84,28%	REPLICA 5, REPLICA 10
	ΣTj	466945	466945	467329	467002	466945	467335	466996	466945	467469	466996	466945	467469	467207			
	ΣUj	83	83	84	84	83	84	83	83	84	83	83	84	84			
PH1S5T9	Cmax	10319	10269	10311	10333	10266	10278	10262	10367	10278	10262	10262	10367	10315	5511	86,21%	REPLICA 4, REPLICA 7
	ΣTj	513469	513879	513673	513034	513072	513351	513072	513351	513351	513072	513034	513879	513457			
	ΣUj	83	84	84	83	84	84	83	84	84	84	83	84	84			
PH1S5T10	Cmax	5231	5206	5196	5259	5231	5205	5196	5231	5210	5181	5181	5259	5220	4749	9,10%	REPLICA 2, REPLICA 10
	ΣTj	237574	237360	237833	238566	237574	238001	237833	237574	238566	237833	237360	238566	237963			
	ΣUj	82	82	83	82	84	84	83	84	82	82	82	84	83			

Tabla 8. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 100 trabajos y 5 estaciones. Fuente: El autor.

INSTANCIA	F. OBJETIVO	REPLICA 1	REPLICA 2	REPLICA 3	REPLICA 4	REPLICA 5	REPLICA 6	REPLICA 7	REPLICA 8	REPLICA 9	REPLICA 10	MIN.	MAX.	PROM.	VALOR MAKESPAN LOWER BOUND	ERROR REL. MIN. VALOR FF- MUCH VS.	SOLUCIONES NO DOMINADAS
PH1S8T1	Cmax	9287	9355	9349	9355	9347	9352	9349	9347	9349	9355	9287	9355	9321	4970	86,86%	REPLICA 1, REPLICA 4
	ΣTj	426315	425458	426375	424216	426375	425515	426465	425515	425515	426540	424216	426540	425378			
	ΣUj	82	83	83	82	82	84	83	83	83	84	82	84	83			
PH1S8T2	Cmax	15570	15445	15331	15452	15473	15536	15496	15510	15470	15464	15331	15570	15451	5746	166,81%	REPLICA 3, REPLICA 9
	ΣTj	737452	737107	737807	737600	737207	738307	738380	736851	735575	737989	735575	738380	736978			
	ΣUj	87	85	84	85	85	86	85	86	84	85	84	87	86			
PH1S8T3	Cmax	12566	12525	12418	12525	12621	12516	12621	12621	12583	12446	12418	12621	12520	5612	121,28%	REPLICA 3, REPLICA 6
	ΣTj	598778	598392	598392	597323	598392	590128	597323	598128	598128	599694	590128	599694	594911			
	ΣUj	83	87	82	83	83	82	84	83	83	83	82	87	85			
PH1S8T4	Cmax	11791	11791	11680	11811	11794	11712	11713	11752	11717	11811	11680	11811	11746	5818	100,76%	REPLICA 3, REPLICA 6
	ΣTj	549990	549385	549385	551583	550737	548487	550262	550206	550206	551218	548487	551583	550035			
	ΣUj	84	83	82	83	85	82	86	84	83	85	82	86	84			
PH1S8T5	Cmax	11264	11268	11186	11287	11291	11271	11236	11249	11216	11219	11186	11291	11239	5196	115,28%	REPLICA 3
	ΣTj	528663	528409	528219	528706	528921	528791	529039	528655	528976	528915	528219	529039	528629			
	ΣUj	84	83	83	85	84	83	85	83	83	83	83	85	84			
PH1S8T6	Cmax	16760	16700	16679	16674	16679	16679	16756	16727	16415	16665	16415	16760	16588	6194	165,01%	REPLICA 9, REPLICA 10
	ΣTj	806708	806214	805429	806256	806754	805598	806329	805671	805190	804937	804937	806754	805846			
	ΣUj	85	84	85	84	85	84	85	85	83	83	83	85	84			
PH1S8T7	Cmax	14649	14637	14551	14565	14658	14658	14649	14551	14658	14669	14551	14669	14610	5896	146,79%	REPLICA 1, REPLICA 3
	ΣTj	717639	719013	718030	718543	718881	718126	718126	719013	718543	718126	717639	719013	718326			
	ΣUj	83	84	83	87	85	84	84	85	83	83	83	87	85			
PH1S8T8	Cmax	9180	9153	9147	9177	9146	9162	9162	9177	9177	9128	9128	9180	9154	2882	216,72%	REPLICA 3, REPLICA 5, REPLICA 10
	ΣTj	413494	413325	412809	413590	413050	413265	413078	413078	413059	413059	412809	413590	413200			
	ΣUj	81	83	81	83	81	82	82	81	83	81	81	83	82			
PH1S8T9	Cmax	12527	12457	12531	12481	12467	12428	12473	12520	12520	12531	12428	12531	12480	5023	147,42%	REPLICA 6
	ΣTj	584485	584269	584460	584335	584448	582115	584423	584460	584335	584335	582115	584485	583300			
	ΣUj	83	82	83	84	82	82	83	83	84	83	82	84	83			
PH1S8T10	Cmax	12252	12173	12069	12127	12232	12268	12281	12245	12188	12116	12069	12281	12175	4847	149,00%	REPLICA 3, REPLICA 7
	ΣTj	592929	660723	593677	593299	592661	592949	592110	593272	661232	593996	592110	661232	626671			
	ΣUj	84	83	83	85	84	83	83	84	85	83	83	85	84			

Tabla 9. Diferencia porcentual entre LOWER-BOUND y FF-MUCH para 10 instancias de 100 trabajos y 8 estaciones. Fuente: El autor.

Análisis de los resultados obtenidos

Los resultados de las tablas 1 al 9 donde se observa el desempeño del algoritmo FF-MUCH para los diferentes escenarios se observa que el algoritmo propuesto tiene muy buen desempeño para dos estaciones en los tres conjuntos de trabajos analizados. Para el escenario de 20 trabajos y dos estaciones, el algoritmo FF-MUCH tiene su mejor desempeño alcanzando inclusive a superar a LEKIN® en los tres criterios evaluados con respecto a la heurística General SB routine. En los casos en los que FF-MUCH supera a LEKIN® en dos de los tres criterios considerados, los que tienen predominancia son el makespan y el número de trabajos tardíos, especialmente este último siendo en general el criterio que mejor desempeño presenta en términos de variabilidad y desviación con respecto a los parámetros establecidos, la cual en la mayoría de los casos es inferior al 10%. Para el caso de la tardanza total, esta medida de desempeño no tiene un buen comportamiento, lo cual es consistente con la teoría de optimización multiobjetivo en el sentido de que si se mejora un criterio, se tiende a desmejorar el resto; esto se ve en tres instancias particulares donde la diferencia porcentual de este criterio está por encima del 50% (pero inferior a 100%) con relación a LEKIN®.

Con respecto al desempeño del algoritmo para 20 trabajos y 5 estaciones, el comportamiento de FF-MUCH es bastante aceptable, teniendo en cuenta que la herramienta propuesta optimiza los tres criterios simultáneamente, aunque desmejora con respecto al escenario de dos estaciones. Para algunos casos de 5 estaciones no es tan evidente una superioridad de FF-MUCH con respecto a LEKIN®, sin embargo hay instancias en las que sí se muestra esta superioridad en los tres criterios y otras en las que lo supera en dos criterios, que son el makespan y el número de trabajos tardíos. Nuevamente para estos casos el criterio que tiene el mejor desempeño es el número de trabajos tardíos, donde en la mayoría de escenarios evaluados supera a LEKIN®, mientras que el peor es la tardanza

total, donde esta diferencia con respecto a los valores de LEKIN® se hace más notoria conforme aumenta el número de estaciones.

Para el caso de 20 trabajos y 8 estaciones, el desempeño del algoritmo disminuye; sin embargo, en casi la mitad de las instancias FF-MUCH supera a LEKIN® en dos de los tres objetivos simultáneamente, makespan y número de trabajos tardíos, y en otras lo supera en el número de trabajos tardíos, confirmando que es la medida de desempeño que mejor se comporta; lo mismo sucede para la tardanza total, que es la que peor se comporta. Por lo tanto, se podría decir para el caso de 20 trabajos que el algoritmo FF-MUCH propuesto tiende a mejorar en primera medida el número de trabajos tardíos y luego el makespan, pero en el proceso se causa un detrimento que en algunos casos es bastante considerable en la tardanza total, por encima del 100%, con respecto al patrón de comparación establecido.

Para el escenario compuesto por 50 trabajos y teniendo en cuenta que sólo se compara en este caso el makespan, la herramienta FF-MUCH también presenta un buen desempeño con respecto a la cota inferior o Lower-Bound para dos estaciones, donde la primera supera a la segunda en aproximadamente la mitad de los escenarios evaluados; adicionalmente, cuando FF-MUCH es superado por Lower-Bound, las diferencias porcentuales no son muy altas, por debajo del 20%, salvo en un caso particular en el que esta diferencia es del 30%, teniendo en cuenta que las cotas propuestas por Santos et. al (1995), son muy cercanas al óptimo. Del mismo modo que para el caso de 20 trabajos y 5 estaciones, para 50 trabajo y 5 estaciones el desempeño del algoritmo FF-MUCH propuesto tiende a desmejorar y los casos en donde este supera a LEKIN®, en este escenario son menores que para 50 trabajos y 2 estaciones. Adicionalmente, las diferencias porcentuales con respecto a la cota inferior son mayores en los casos en que LEKIN® supera a FF-MUCH, en donde en un caso sobrepasa el 60% inclusive.

Con respecto al escenario de 50 trabajos y 8 estaciones, el algoritmo presenta un pobre desempeño, en donde el algoritmo propuesto no se acerca a la cota inferior en ninguna de las instancias consideradas; además, las diferencias porcentuales son considerablemente mayores que para el caso de 2 y 5 estaciones (por encima del 100%); sin embargo, hay que aclarar nuevamente el hecho de que para el escenario de 50 trabajos (y para el de 100), no se cuentan con las medidas correspondientes a la tardanza total ni al número de trabajos tardíos, sino sólo la del makespan, aunque se presume que el comportamiento sería similar al escenario de 20 trabajos, donde el criterio de mejor desempeño es el número de trabajos tardíos y el de peor desempeño es la tardanza total, pero esto habría que comprobarlo experimentalmente.

Por último, para el escenario de 100 trabajos, se presenta un comportamiento similar que en los de 20 y 50 trabajos, en el sentido de que el algoritmo FF-MUCH tiene un buen desempeño para 2 estaciones, pero este va disminuyendo cuando aumenta el número de estaciones, especialmente para 8 estaciones. Para el caso de 100 trabajos y 2 estaciones, el algoritmo propuesto tiene un muy buen desempeño en el makespan, logrando tener diferencias inclusive por debajo del 1% con la cota inferior de Santos et. al (1995). En los casos en que Lower-Bound supera a FF-MUCH las diferencias porcentuales no son muy acentuadas, salvo sólo en el primer caso, por lo que los valores obtenidos con FF-MUCH son muy cercanos a los valores de esta cota.

Por otro lado, para el escenario de 5 estaciones, el desempeño del algoritmo propuesto (al igual que en las instancias de 20 y 50 trabajos) tiende a desmejorar, así como también aumenta la diferencia porcentual de FF-MUCH con respecto a Lower-Bound, en donde en tres de las diez instancias, esta diferencia es relativamente pequeña. Para el caso de 8 estaciones, se ve más claramente la diferencia porcentual entre FF-MUCH y Lower-Bound,

donde esta es muy grande en la mayoría de los casos, lo que confirma que a medida que se aumenta el número de estaciones, el desempeño del algoritmo desmejora.

Por lo tanto, se puede concluir, teniendo en cuenta los escenarios de 20, 50 y 100 trabajos analizados, que conforme aumenta el número de estaciones, el desempeño del algoritmo tiende a desmejorar con respecto a las medidas de desempeño evaluadas; adicionalmente, cuando el número de trabajos aumenta, la diferencia porcentual entre FF-MUCH y Lower-Bound tiende a aumentar significativamente. No se evidencia tanto una disminución del desempeño algoritmo propuesto cuando aumenta el número de trabajos (aunque hay una ligera influencia), como cuando aumenta el número de estaciones, donde se observó en los escenarios evaluados que para dos estaciones, los resultados son prácticamente los mismos para 20, 50 y 100 trabajos, teniendo en cuenta que en estos dos últimos casos sólo se evaluó uno de los tres criterios considerados, además de que el criterio de comparación varía de 20 con respecto a 50 y 100 trabajos. Del mismo modo, lo que sucede para 5 y 8 estaciones se presenta de manera similar en los tres escenarios de 20, 50 y 100 trabajos. Sin embargo, esto habría que comprobarlo experimentalmente.

Una razón de que el algoritmo disminuya su desempeño a medida que aumenta el número de estaciones para los escenarios analizados, se debe probablemente a que a medida que el tamaño del problema crece, la información heurística compartida que maneja el algoritmo es mayor, lo mismo que el comportamiento aleatorio del mismo, por lo que aumenta el riesgo de que las decisiones tomadas por la regla de transición de estados pueda ocasionar cambios bruscos en dicha información que evita que la metaheurística converja satisfactoriamente o se obtengan buenas soluciones. En ese sentido, se podría evaluar el desempeño del algoritmo colonia de hormigas junto con otra metodología, de modo que se pueda superar esa dificultad. Del mismo modo, como se mostró en la sección anterior, la estructura del algoritmo propuesto resuelve el problema bajo estudio estación por estación, lo que refuerza la anterior explicación, debido a que las decisiones tomadas por la regla de

transición de estados, se modifican a medida que se va de una estación a otra por la naturaleza aleatoria de la metaheurística, por lo que si sólo se trabajan un número pequeño de estaciones (por ejemplo 2), la información heurística está casi inalterada (o levemente modificada) conservando más su estructura inicial, a diferencia de que si son 5, 8 o más estaciones.

6. CONCLUSIONES Y FUTUROS TRABAJOS

En este trabajo de grado se utilizó la metaheurística de colonia de hormigas para resolver un problema multi-objetivo de programación de la producción, cuyos objetivos a optimizar fueron el makespan, la tardanza total y el número de trabajos tardíos, a través de una herramienta computacional denominada FF-MUCH. La optimización por colonia de hormigas ha mostrado ser eficiente para resolver problemas de scheduling con un solo objetivo y por lo tanto se puede extender su utilización para problemas multi-objetivos.

Como primera medida, se compararon las soluciones obtenidas por FF-MUCH con el software LEKIN ® para instancias de 20 trabajos utilizando la heurística cuello de botella General SB routine. El algoritmo FF-MUCH presenta muy buenas soluciones frente a LEKIN ® para dos y cinco estaciones en las instancias analizadas, en donde FF-MUCH domina en varios casos fuertemente a los resultados producidos por la herramienta LEKIN ® para los tres criterios simultáneamente, especialmente para dos estaciones. Para el caso de 50 y 100 trabajos, se comparó el algoritmo propuesto con la heurística para calcular la cota inferior propuesta por (Santos et. al, 1995). Para estos casos, el algoritmo FF-MUCH presenta valores muy cercanos al valor de la cota inferior para el makespan para 2 estaciones, pero aumenta la diferencia porcentual conforme aumenta el número de trabajos.

Por otro lado, la calidad de las soluciones disminuye conforme aumenta el número de estaciones, aunque no sucede esto si aumenta el número de trabajos. Para el escenario con instancias de cinco estaciones, el algoritmo presenta un desempeño aceptable, donde en la mitad de los casos se obtuvieron muy buenas soluciones con relación a LEKIN ® para 20 trabajos, especialmente para dos de los tres criterios considerados (makespan y el número de trabajos tardíos) y para 50 y 100 trabajos, también aproximadamente en la mitad de las instancias evaluadas, se alcanzaron valores muy cercanos a la cota inferior del makespan.

Para ocho estaciones, el desempeño del algoritmo no es bueno, tanto para 20, 50 y 100 trabajos en las instancias consideradas.

Lo anterior puede estar sucediendo tal vez porque el algoritmo para instancias moderadamente grandes (en este caso para el número de estaciones), la información heurística compartida que maneja el algoritmo es mayor, lo mismo que el comportamiento aleatorio del mismo, por lo que aumenta el riesgo de que las decisiones tomadas por la regla de transición de estados ocasionen cambios bruscos en dicha información que evita que la metaheurística obtenga soluciones de buena calidad, por lo que sería conveniente evaluar una técnica complementaria que permita mejorar las soluciones encontradas en estos escenarios. Una alternativa puede ser la búsqueda local o la búsqueda tabú, con las cuales algunos estudios muestran que el algoritmo colonia de hormigas tiene su mejor desempeño, especialmente la búsqueda local (Yagmahan y Mutlu-Yenesey, 2010). Aun así, los resultados obtenidos se consideran aceptables teniendo en cuenta la complejidad del problema y el número de pruebas realizadas para probar el desempeño del algoritmo.

En el caso de los escenarios donde se compararon los tres criterios se observó que el criterio que mejor desempeño tiene es el número de trabajos tardíos, donde para todos los escenarios donde se comparó con LEKIN® para 20 trabajos con 2,5 y 8 estaciones obtuvo muy buenos valores salvo en casos muy puntuales. Del mismo modo, el makespan presenta muy buenas soluciones para dos estaciones y en varios casos para cinco estaciones pero se disipa su desempeño para instancias de ocho estaciones. La tardanza total es también el criterio que peor desempeño tiene en la mayoría de los escenarios evaluados con respecto a LEKIN®, aunque para las instancias de dos estaciones en varios casos lo supera, pero para cinco y ocho estaciones, su desempeño desmejora considerablemente, y su diferencia porcentual con respecto a este último es bastante amplia, especialmente para 8 estaciones.

El hecho anterior se justifica con el argumento de que en las soluciones de un problema con objetivos múltiples se tienen en cuenta todos los objetivos al mismo tiempo, por lo cual, ningún objetivo puede mejorarse sin degradar a los demás. Luego se establece como para afirmar que una situación o solución es mejor que otra, el que en ella no se disminuya a nadie, pero se mejore a alguno; es decir que una situación será mejor que otra sólo si en la nueva es posible compensar las pérdidas de todos los perjudicados y aún queda un sobrante (Solano-Charris, 2008). En este caso, el objetivo que el algoritmo propuesto tiende a mejorar, es el número de trabajos tardíos, sacrificando un poco al makespan y en mayor medida a la tardanza total.

Como enfoque alternativo para los escenarios en los que el algoritmo FF-MUCH tuvo un pobre desempeño, especialmente para las instancias con ocho estaciones, podría evaluarse la utilización de una técnica de hibridización con otras metaheurísticas de modo que se pueda mejorar la calidad de las soluciones encontradas como. En tal sentido, podría utilizarse Colonia de Hormigas con GRASP, Algoritmos Genéticos o Recocido Simulado, además de las técnicas de búsqueda mencionadas anteriormente. Sin embargo, se debe evaluar el desempeño del algoritmo propuesto en términos de la calidad de las soluciones generadas y de la eficiencia computacional, con una y otra metodología. Del mismo modo, podría evaluarse el uso de reglas de despacho como información heurística, ya sea SPT, LPT o SRPT, entre otras, o inclusive otras heurísticas para el makespan, como el método de Palmer o la heurística NEH. Sin embargo, para estos dos escenarios habría que hacer un diseño experimental de modo que se puedan establecer diferencias significativas entre los diferentes enfoques alternativos de hibridización para colonia de hormigas, así como también del uso de diferentes reglas de despacho y otras heurísticas como información inicial.

Como futuros trabajos se puede extender este enfoque a problemas multi-objetivos en donde los tiempos de procesamiento no sean determinísticos, sino probabilísticos, lo cual se ajusta más al contexto real. Del mismo modo, se podrían tener en cuenta los tiempos de alistamiento y/o mantenimiento de las máquinas como dependientes de la secuencia de producción, entre otras consideraciones tales como tener máquinas no idénticas o de distintas velocidades en cada estación y para cada trabajo, e inclusive se podría extender este enfoque a configuraciones o talleres de producción un poco más complejos, como el caso del jopshop flexible, el openshop o aún en sistemas de producción reconfigurables.

Adicionalmente a lo que se mencionó anteriormente sobre la utilización de Colonia de hormigas junto con otras metaheurísticas, se podrían evaluar enfoques híbridos alternativos con otras metodologías tales como: redes neuronales, lógica difusa o redes de petri, que aunque estas pertenecen a diferentes campos, también son herramientas que permiten resolver problemas muy complejos sin el grado de dificultad y laboriosidad que en general presentan las metaheurísticas y permiten combinarse con estas. Por ejemplo, existen aplicaciones en electrónica de las redes neuronales o sistemas fuzzy junto con algoritmos genéticos y recocido simulado, pero muy pocas (o ninguna) en las que se utilizan colonias de hormigas.

7. REFERENCIAS BIBLIOGRAFICAS

Alfieri A. Workload simulation and optimisation in multi-criteria hybrid flowshop scheduling: a case study *International Journal of Production Research*. Vol. 47, No. 18, 15 September 2009, 5129–5145.

Allaoui Hami, Lamouri Samir, Lebbar Maria. A robustness framework for a stochastic hybrid flow shop to minimize the makespan. 2006. IEEE.

Allaoui Hami, Artiba Abdelhakim. Scheduling two-stage hybrid flow shop with availability constraints. *Computers & Operations Research* 33 (2006) 1399–1419.

Allaoui Hami, Artiba A. Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. *Computers & Industrial Engineering* 47 (2004) 431–450. Science Direct.

Amin-Naseri Mohammad Reza, Beheshti-Nia Mohammad Ali. Hybrid flow shop scheduling with parallel batching. *International Journal of Production Economics* 117 (2009) 185–196. Science Direct.

Barichard V., Ehrgott M., Gandibleux X., T'Kindt V . *Multiobjective Programming and Goal Programming. Theoretical Results and Practical Applications*. Springer. 2009.

Bagchi, T. (1999). *Multiobjective scheduling by genetic algorithms*. Kluwer Academic Publishers.

Brah, S. A., & Hunsucker, J. L. (1991). Branch and bound algorithm for the flowshop with multiple processors. *European Journal of Operation Research*, 51, 88–99.

Behnamian J., Ghomi Fatemi, M. Zandieh. A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. *Expert Systems with Applications*. Science Direct.

Behnamian J., Ghomi Fatemi, M. Zandieh. Hybrid flowshop scheduling with machine and resource-dependent processing times. *Applied Mathematical Modelling*. Science Direct 17 pags.

Blum, C., Sampels, M., 2002. Ant colony optimization for FOP shop scheduling: a case study on different pheromone representations. *Proceedings of the 2002 Congress on*

Evolutionary Computation (CEC'02), vol.2. IEEE Computer Society Press, Los Alamitos, pp.1558–1563.

Botta-Genoulaz Valérie. Hybrid Flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *International Journal of Production Economics* 64 (2000) 101-111. Science Direct.

Botta-Genoulaz Valérie. Considering Bills of material in Hybrid Flow shop Scheduling Problems. *Proceedings of the 1997 IEEE International Symposium on Assembly and Task Planning Marina del Rey, CA - August 1997.*

Caricato P., Grieco A., Serino D. Tsp-based scheduling in a batch-wise hybrid flow-shop. *Robotics and Computer-Integrated Manufacturing* 23 (2007) 234–241. Science Direct.

Caricato Pierpaolo, Grieco Antonio, Nucci Francesco. Simulation and mathematical programming for a multi-objective configuration problem in a Hybrid Flow Shop. *Proceedings of the 2008 Winter Simulation Conference* S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

Chang Junlin, Yan Weiwu. Shao Huihe. Scheduling a two-stage no-wait hybrid flowshop with separated setup and removal times. *Proceeding of the 2004 American Control Conference Boston, Massachusetts June 30. July 2, 2004.*

Chang Junlin, Ma Guangfu, Ma Xiaoping. A New Heuristic for Minimal Makespan in No-Wait Hybrid Flowshops. *Proceedings of the 25th Chinese Control Conference.* 7-11 August, 2006, Harbin, Heilongjiang.

Chen, B. (1995). Analysis of classes of heuristics for scheduling a two-stage flowshop with parallel machines at one stage. *Journal of the Operational Research Society*, 46, 234–244.

Chen, B. (1994). Scheduling multiprocessor flowshops. In *Advances in optimization and approximation.* Dordrecht: Kluwer.

Choi, S.-W., Y.-D. Kim, and G.-C. Lee. 2005. Minimizing total tardiness of orders with reentrant lots in a hybrid flow shops. *International Journal of Production Research* 43:2149-2167

Choi Hyun-Seon, Lee Dong-Ho Scheduling algorithms to minimize the number of tardy jobs in two-stage hybrid flow shops. *Computers & Industrial Engineering* 56 (2009) 113–120. Science Direct.

Choi Hyun-Seon, Kim Ji-Su, Lee Dong-Ho. Real-time scheduling for reentrant hybrid flow shops: A decision tree based mechanism and its application to a TFT-LCD line. *Expert Systems with Applications*. 2010. Science Direct.

Coello-Coello, C.A., Lamont G. B., & Van-Veldhuizen D. A. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Second Edition. Springer. 2007.

Davis, Lawrence (Editor). "Handbook of Genetic Algorithms", Van-Nostrand Reinhold, 1991. 385 p.

Dauzère-Pérès, S., 1994. A connected neighborhood structure for the multiprocessor job-shop scheduling problem. *Management Report Series*, vol. 180. Erasmus University Rotterdam, Rotterdam School of Management, Rotterdam.

Dauzère-Pérès, S., Paulli, J., 1994. Solving the general job-shop scheduling problem. *Management Report Series*, vol. 182. Erasmus University Rotterdam, Rotterdam School of Management, Rotterdam.

Dauzère-Pérès, S., Paulli, J., 1997. An integrated approach for modelling and solving the general multiprocessor job-shop scheduling problem using tabu search. *Annals of Operations Research* 70, 281-306.

Dessouky, M., Dessouky, M., & Verma, S. (1998). Flowshop scheduling with identical jobs and uniform parallel machines. *European Journal of Operational Research*, 109, 620–631.

Díaz Adenso. *Optimización Heurística y Redes Neuronales*. Editorial Paraninfo. 1996.

Dugardin Frédéric, Yalaoui Farouk, Amodeo Lionel. New multi-objective method to solve reentrant hybrid flow shop scheduling problem. *European Journal of Operational Research* 203 (2010) 22–31. Science Direct.

Dauzère-Pérès, S., 1994. A connected neighborhood structure for the multiprocessor job-shop scheduling problem. *Management Report Series*, vol. 180. Erasmus University Rotterdam, Rotterdam School of Management, Rotterdam.

Dauzère-Pérès, S., Paulli, J., 1994. Solving the general job-shop scheduling problem. *Management Report Series*, vol. 182. Erasmus University Rotterdam, Rotterdam School of Management, Rotterdam.

Dauzère-Pérès, S., Paulli, J., 1997. An integrated approach for modelling and solving the general multiprocessor job-shop scheduling problem using tabu search. *Annals of Operations Research* 70, 281-306.

De la cruz, Jair J. and Carlos D. Paternina-Arboleda. An Ant Colony System approach for heterogeneous vehicle routing problems with time windows and multiple products. III Colombian congress and 1st international Andean conference on Operational Research. Cartagena, Colombia, 2004.

De la cruz, Jair J., Adriana Mendoza, Astrid Del Castillo y Carlos D. Paternina-Arboleda. Análisis comparativo de las aproximaciones heurísticas Ant-Q, recocido simulado y búsqueda tabu en la solución del problema del agente viajero. *Revista Ingeniería & Desarrollo*, (14) Universidad del Norte, 2004.

Dessouky, M., Dessouky, M., & Verma, S. (1998). Flowshop scheduling with identical jobs and uniform parallel machines. *European Journal of Operational Research*, 109, 620–631.

Dorigo, M. Optimization, learning and natural algorithms [in Italian]. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1992.

Dorigo, M. & Gambardella, L. M.. Ant colonies for the travelling salesman problem. *BioSystems*, 43(2):73–81, 1997.

Dorigo, M., Maniezzo, V., & Colomi, A. Positive feedback as a search strategy. Technical report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1991a.

Dorigo, M., Maniezzo, V., & Colomi, A. The ant system: An autocatalytic optimizing process. Technical report 91-016 revised, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1991b.

Dorigo, M., Maniezzo, V., & Colomi, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 26(1):29–41, 1996.

Dorigo, M., & Stützle, T. *Ant Colony Optimization*. Asco Typesetters, Hong Kong, 2004.

Dugardin Frédéric, Yalaoui Farouk, Amodeo Lionel. Multiobjective scheduling of a reentrant hybrid flowshop.

Feng Haodi, Lu Shengpeng, Li Xiuqian. Genetic Algorithm for Hybrid Flow-shop Scheduling with Parallel Batch Processors. 2009 WASE International Conference on Information Engineering.

Gao Cong, Tang Lixin. Scheduling Hybrid Flow Shop for Minimizing Total Weight Completion Time. Proceedings of the IEEE International Conference on Automation and Logistics August 18 - 21, 2007, Jinan, China.

Gholami M., Zandieh M., Alem-Tabriz A. Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. Springer-Verlag London Limited 2008.

Grangeon Nathalie, Tanguy Alain, Tchernev Nikolay. Generic Simulation Model for Hybrid Flow-Shop. Computers & Industrial Engineering 37 (1999) 207-210.

Guinet A., & Solomon M. M. (1996). Scheduling hybrid Flowshops to minimize maximum tardiness or maximum completion-times.

Gupta, J. (1988). Two stage hybrid flowshop scheduling problem. Journal of the Operations Research Society, 38, 359–364.

Gupta J., & Tunc E. A. (1994). Scheduling a 2-stage hybrid Flowshop with separable setup and removal times.

Gupta Jatinder N.D., Krüger Karin, Lauff Volker, Werner Frank, Sotskov Yuri N. Heuristics for hybrid flow shops with controllable processing times and assignable due dates. Computers & Operations Research 29 (2002) 1417-1439. Science Direct.

Haouari, M., & M'Hallah, R. (1997). Heuristic algorithms for the two-stage hybrid flowshop problem. Operations Research Letters, 21, 43–53.

Hentous H., Benhammedi F. Heuristics Resolution of a Constrained Hybrid Flow Shop Problem. 2006. IEEE.

Hmida Abir Ben, Huguet Marie-Jose, Lopez Pierre, Haouari Mohamed. Adaptation of Discrepancy-based Methods for Solving Hybrid Flow Shop Problems. 2006. IEEE.

Hoogeveen, H. (2005). Multicriteria scheduling. European Journal of Operational Research, 167, 592–623.

Hossein-Hojjati Seyed Mohammad, Sahraeyan Amin. Minimizing Makespan Subject to Budget Limitation in Hybrid Flow Shop. 2009. IEEE.

Hu Yanhai. Ma Dengzhe. Ye Feifan. Scheduling Hybrid Flow Shop Containing Multi-Functional Machines. 2006. IEEE.

Huang Rong-Hwa & Yang Chang-Lin. Ant colony system for job shop scheduling with time Windows. *International Journal Advanced Manufacturing Technology*. 39:151–157 DOI 10.1007/s00170-007-1203-9. Springer-Verlag London Limited 2007.

Janiak Adam, Kozan Erhan, Lichtenstein Maciej, Oguz Ceyda. Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion. *International Journal of Production Economics* 105 (2007) 407–424. Science Direct.

Johnson, D. S., Aragon, C. R., McGeoch, Lyle A., Schevon, C. Optimization by Simulated Annealing: An Experimental Evaluation. *Operations Research*; Nov/Dec 1989; 37, 6; ABI/INFORM Global pag. 865.

Jiang Shujun, Tang Lixin. Lagrangian Relaxation algorithms for re-entrant hybrid flowshop scheduling. 2008 *International Conference on Information Management, Innovation Management and Industrial Engineering*. IEEE.

Jing Xiao, Li Zheng. A MILP-based batch scheduling for two-stage hybrid flowshop with sequence-dependent setups in semiconductor assembly and test Manufacturing. 6th annual IEEE Conference on Automation Science and Engineering. Marriott Eaton Centre Hotel. Toronto, Ontario, Canada, August 21-24, 2010.

Jouglet Antoine, Oguz Ceyda, Sevaux Marc. Hybrid Flow-Shop: a Memetic Algorithm Using Constraint-Based Scheduling for Efficient Search. Springer Science + Business Media B.V. 2009.

Jungwattanakit Jitti, Reodecha Manop, Chaovalitwongse Paveena, Werner Frank. Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. 13 pags.

Jungwattanakit Jitti, Reodecha Manop, Chaovalitwongse Paveena, Werner Frank. Sequencing algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. 23 pags.

Jungwattanakit Jitti, Reodecha Manop, Chaovalitwongse Paveena, Werner Frank. An Evaluation of Sequencing Heuristics for Flexible Flowshop Scheduling Problems with Unrelated Parallel Machines and Dual Criteria. 23 pags.

Jungwattanakit Jitti, Reodecha Manop, Chaovalitwongse Paveena, Werner Frank. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *Computers & Operations Research* 36 (2009) 358 – 378. Science Direct.

Kahramana Cengiz, Engin Orhan, Kaya Ihsan, Öztürk R. Elif. Multiprocessor task scheduling in multistage hybrid flow-shops: A parallel greedy algorithm approach. *Applied Soft Computing* 10 (2010) 1293–1300. Science Direct.

Karimi N., Zandieh M., Karamooz H.R. Bi-objective group scheduling in hybrid flexible flowshop: A multi-phase approach. *Expert Systems with Applications* 37 (2010) 4024–4032. Science Direct.

Khalouli Safa, Ghedjati Fatima, Hamzaoui Abdelaziz. A meta-heuristic approach to solve a JIT scheduling problem in hybrid flow shop *Engineering Applications of Artificial Intelligence*. Science Direct.

Khalouli Safa, Ghedjati Fatima, Hamzaoui Abdelaziz. An integrated ant colony optimization algorithm for the hybrid flow shop scheduling problema. 2009. IEEE.

Khalouli Safa, Ghedjati Fatima, Hamzaoui Abdelaziz. Ant Colony Optimization for Solving a Bi-Criteria Hybrid Flow Shop Problem. 2008. IEEE.

Kirkpatrick, S., Gelatt, C.D., Vecchi P.M. Optimization by Simulated Annealing. *Science*, 220, 671-680. 1983.

Kulcsár Gyula, Erdélyi Ferenc and Hormyák Olivér. Multi-Objective Optimization and Heuristic Approaches for solving scheduling problems. University of Miskolc, Department of Information Engineering. Egyetem Road, Miskolc-Egyetemváros 3515, Hungary.

Kyparisis George J., Koulamas Christos. Flexible flow shop scheduling with uniform parallel machines. *European Journal of Operational Research* 168 (2006) 985–997. Science Direct.

Lee C. Y., & Vairaktarakis G. L. (1994). Minimizing makespan in hybrid Flowshops.

Lei, D. and Wu, Z., 2006, Crowding-measure-based multiobjective evolutionary algorithm for job shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 30 (1-2), 112-117.

Li Tieke, Li Yan. Constructive Backtracking Heuristic for Hybrid Flowshop Scheduling with Limited Waiting Times. 2007. IEEE.

Li Tieke, Xiao Yongjun, Wang Haifeng Dynamic Constraint Satisfaction Approach to Hybrid Flowshop Rescheduling. Proceedings of the IEEE International Conference on Automation and Logistics August 18 - 21, 2007, Jinan, China.

Lin Hung-Tso, Liao Ching-Jong. A case study in a two-stage hybrid flow shop with setup time and dedicated machines. *Int. J. Production Economics* 86 (2003) 133–143. Science Direct.

Li Lin, Wang Lele, Huo Jiazhen. Hybrid flowshop scheduling with setup times for cold treating process in Baoshan Iron & Steel Complex. 2010. IEEE.

Linn, R., & Zhang, W. (1999). Hybrid flowshop scheduling: A survey. *Computers and Industrial Engineering*, 37, 57–61.

Ling-li Zeng, Feng-xing Zou, Zheng Gao, Xiao-hong Xu. Dynamic Scheduling of Multi-task for Hybrid Flow-shop Based on Idle Time Windows. 2009 IEEE.

Ling-li Zeng, Feng-xing Zou, Zheng Gao, Xiao-hong Xu. Dynamic Scheduling of Multi-task for Hybrid Flow-shop Based on Energy Consumption. Proceedings of the 2009 IEEE International Conference on Information and Automation June 22-25, 2009, Zhuhai/Macau, China.

Liu Feng, Zhang Xiang-ping, Zou Feng-xing, Zeng Ling-li. Immune Clonal Selection Algorithm for Hybrid Flow-shop Scheduling Problem. 2009. IEEE.

Liu Siyi, Cui Jianshuang, Li Yan. Heuristic-Tabu Algorithm for Hybrid Flowshop Scheduling with Limited Waiting Time. 2008 International Symposium on Computational Intelligence and Design.

Liu Xiang, Zou Fengxing, Zhang Xiangping. Mathematical Model and Genetic Optimization for Hybrid Flow Shop Scheduling Problem Based on Energy Consumption. 2008. IEEE.

Logendran Rasaratnam, Carson Sara, Hanson Erik. Group scheduling in flexible flow shops. *International Journal of Production Economics* 96 (2005) 143–155. Science Direct.

Logendran Rasaratnam, DeSzoeka Paula, Barnard Faith. Sequence-dependent group scheduling problems in flexible flow shops. *International Journal of Production Economics* 102 (2006) 66–86. Science Direct.

Mahdavi I., Mojarad M. S., Javadi B., Tajdin A. A Genetic Approach for Solving a Hybrid Flow Shop Scheduling Problem. 2008. IEEE.

Montoya-Torres, J. R., & Vargas-Nieto, F. (2008). Production scheduling in the apparel industry using an evolutionary algorithm. Submitted.

Montoya-Torres, J. R. *Modelos y Métodos de Programación de Operaciones*. Editorial Lulu. 2010.

Montoya-Torres, J.R., Vargas-Nieto, F. (in press). Solving a bi-criteria hybrid flowshop scheduling problem occurring in apparel manufacturing. *International Journal of Information Systems and Supply Chain Management*.

Montoya-Torres, J. R., Aponte A., & Rosas P. Applying GRASP to solve the multi-item threeechelon uncapacitated facility location problem. *Journal of the Operational Research Society* (2010), 1–10.

Moreno J. Marcos, Moreno Jose A. *Heurísticas en Optimización*. Dirección General de Universidades. Gobierno de Canarias. 1999.

Moscato, P., Cotta C. Una introducción a los algoritmos meméticos. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*. No.19 (2003), pp. 131-148.

Negenman, Ebbe G. Local search algorithms for the multiprocessor Flow shop scheduling problem. *European Journal of Operational Research* 128 (2001) 147-158.

Nowicki, E., Smutnicki, C., 1996b. A fast tabu search algorithm for the permutation Flow-shop problem. *European Journal of Operational Research* 91, 160-175.

Oguz Ceyda, Ercan M. Fikret. A Genetic Algorithm for Hybrid Flow-Shop Scheduling With Multiprocessor Tasks. *Journal of Scheduling* 8: 323–351, 2005. Springer Science + Business Media, Inc. Manufactured in the Netherlands.

Oguz Ceyda, Ercan M. Fikret, Cheng T.C. Edwin, Fung Y.F. Heuristic algorithms for multiprocessor task scheduling in a two-stage hybrid flow-shop. *European Journal of Operational Research* 149 (2003) 390–403. Science Direct.

Oguz Ceyda, Zinder Yakov, Do Van Ha, Janiak Adam, Lichtenstein Maciej. Hybrid flow-shop scheduling problems with multiprocessor task systems. *European Journal of Operational Research* 152 (2004) 115–131. Science Direct.

Orhan Engin, Alper Döyen. A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future Generation Computer Systems* 20 (2004) 1083–1095. Science Direct.

Osman, I.H. & Christofides N. Capacitated clustering problems by hybrid simulated annealing and tabu search, *International Transactions in Operational Research*, 1, 317-336.

Pacheco-Espejel, A. y Cruz-Estrada M. C. Metodología crítica de la investigación. CECSA. 2006.

Paternina-Arboleda, Carlos D. y Tapas K. Das. A Multi-Agent Reinforcement Learning Approach to Obtaining Dynamic Control Policies for Stochastic Lot Scheduling Problem. In press, *Simulation Modelling Practice and Theory*. 2005.

Paternina-Arboleda, Carlos D., Montoya-Torres, J. R., Herrera-Hernández, M. C., & Acero-Domínguez, M. (2008). Scheduling jobs on a k-stage flexible flowshop. *Annals of Operations Research*, 164(1), 29–40.

Pinedo, M. *Scheduling: Theory, algorithms, and systems*. Springer. 2008. Reeves, C.R. *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publications. 1993.

Reza Hejazi, S., S. Saghafian (2005) Flowshop-scheduling problems with makespan criterion: a review. *International Journal of Production Research*, 43(14), 2895-2929.

Ribas Imma, Rainer Leisten, Framiñan Jose M. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research* 37 (2010) 1439–1454. Science Direct.

Ruíz Ruben, Vazquez-Rodriguez Jose Antonio. The hybrid flowshop scheduling problem. *European Journal of Operational Research* 205 (2010) 1–18. Science Direct.

Santos, D. L., Hunsucker, J. L., & Deal, D. E. (1995). Global lower bounds for flowshops with multiple processors. *European Journal of Operational Research*, 80, 112–120.

Shiau Der-Fang, Cheng Shu-Chen, Huang Yueh-Min. Proportionate flexible flow shop scheduling via a hybrid constructive genetic algorithm. *Expert Systems with Applications* 34 (2008) 1133–1143. Science Direct.

Sipper, D. y Bulfin, R. *Production, Planning, Control and Integration*. Mc Graw Hill (1997).

Solano-Charris, E. L. (2008). Tesis: Programación de Talleres de Producción Seriales Híbridos (Flexibles) con múltiples objetivos mediante el uso de Metaheurísticas: Colonias de Hormigas. Universidad del Norte, Barranquilla. Maestría en Ingeniería Industrial.

Solano-Charris, E. L., Montoya-Torres, J. M., & Paternina-Arboleda, C. D. (2009). Multi-criteria Optimization Evolving Artificial Ants as a Computational Intelligence Technique. *Proceedings of the IEEE Conference on Computational Intelligence and Games*. 2009.

Solano-Charris, E. L., Montoya-Torres, J. M., & Paternina-Arboleda, C. D. (2009). Ant colony optimization algorithm for a Bi-criteria 2-stage hybrid flowshop scheduling problem. *Journal of Intelligent Manufacturing*. (DOI: 10.1007/s10845-009-0370-y).

Su Zhixiong, Li Tieke. Genetic Algorithm for Minimizing the Makespan in Hybrid Flow Shop Scheduling Problem. This work was supported by the National Natural Science Foundation of China under Grant Nos. 70371057 and 70771008.

Tang Lixin, Liu Wenxin, Liu Jiyin. A neural network model and algorithm for the hybrid flow shop scheduling problema in a dynamic environment. *Journal of Intelligent Manufacturing*, 16, 361–370, 2005 Springer Science Business Media, Inc. Manufactured in The Netherlands.

T'kindt, V., & Billaut, J. C. (2006). *Multicriteria scheduling: Theory, models and algorithms*. Berlin: Springer.

T'kindt, V., Gupta, J. N. D., & Billaut, J. C. (2003). Two-machine flowshop-scheduling with a secondary criterion. *Computers and Operations Research*, 30, 505–526.

Tzung-Pei Hong, Pei-Ying Huang, Gwoboa Horng and Chan-Lon Wang. Three Algorithms for Flexible Flow-shop Scheduling. *American Journal of Applied Sciences* 4 (11): 889-896, 2007. ISSN 1546-9239. 2007 Science Publications.

Uetake Toshifumi, Tsubone Hitoshi, Ohba Masaaki. A production scheduling system in a hybrid flow shop. *International Journal of Production Economics* 41 (1995) 395-398.

Urlings Thijs, Ruiz Rubén, Sivrikaya Funda. Genetic algorithms for complex hybrid flexible flowline problems. March 7, 2007. 32 pags.

Urlings Thijs, Ruiz Rubén. Genetic algorithms with different representation schemes for complex hybrid flexible flow line problems. *Int. J. Metaheuristics, Vol. 1, No. 1, 2010*. Pags 30-54.

Urlings Thijs, Ruiz Rubén, Stutzle Thomas. Shifting representation search for hybrid flexible flowline problems. *European Journal of Operational Research* 207 (2010) 1086–1095. Science Direct.

Vob Stefan, Witt Andreas. Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application. *Int. J. Production Economics* 105 (2007) 445–458. Science Direct.

Wang K. & Choi S.H.. A Decomposition-Based Algorithm for Flexible Flow Shop Scheduling with Stochastic Processing Times. *Proceedings of the World Congress on Engineering and Computer Science 2009 Vol II. WCECS 2009, October 20-22, 2009, San Francisco, USA.*

Widmer, M., Hertz A. A new heuristic method for the flow shop sequencing problem. *European Journal of Operational Research*, 41, 186-193, 1989.

Wojciech Bozejko Jarosław Pemperaz, and Adam Smutnicki. Parallelization of the Tabu Search Algorithm for the Hybrid FlowShop Problem. *Para 2010 – State of the Art in Scientific and Parallel Computing – extended abstract no. 95. University of Iceland, Reykjavik, June 6–9 2010. <http://vefir.hi.is/para10/extab/para10-paper-95.pdf>*

Xiao Wendong, Hao Peifeng, Zhang Sen, Xu Xinhe. Hybrid Flow Shop Scheduling Using Genetic Algorithms. *Proceedings of the 3rd World Congress on Intelligent Control and Automation June 28-Jul. 2, 2000, Hefei, P.R. China.*

Xiaofeng Li Hai Zhao. Greedy Algorithm Solution of Flexible Flow Shop Scheduling Problem. *IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.11, November 2009.*

Xie Xie, Tang Lixin. Analysis of a Heuristics for Scheduling Two-Stage Hybrid Flow Shop. *2009 International Joint Conference on Computational Sciences and Optimization.*

Yagmahan, Betul, Mutlu Yenisey, Mehmet. A multi-objective ant colony system algorithm for flow shop scheduling problem. *Expert Systems with Applications*. 2010. Science Direct.

Yang Jaehwan. Minimizing total completion time in two-stage hybrid flow shop with dedicated machines. *Computers & Operations Research*. 2010. Science Direct.

Yalaoui Naim, Amodeo Lionel, Yalaoui Farouk, Mahdi Halim. Particle Swarm Optimization under Fuzzy Logic Controller for solving a Hybrid Reentrant Flow Shop problem. 2010. IEEE.

Yaurima Víctor, Burtseva Larisa, Tchernykh Andrei. Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. *Computers & Industrial Engineering* 56 (2009) 1452–1463. Science Direct.

Yaurima Víctor, Burtseva Larisa, Tchernykh Andrei. Hybrid Flowshop with Unrelated Machines, Sequence Dependent Setup Time and Availability Constraints: An Enhanced Crossover Operator for a Genetic Algorithm. R. Wyrzykowski et al., (Eds.): PPAM 2007, LNCS 4967, pp. 608–617, 2008. Springer-Verlag Berlin Heidelberg 2008.

Yokohama Masao. Hybrid Flow-shop scheduling with assembly operations. *International Journal of Production Economics* 73 (2001) 103-116. Science Direct.

Yuan Kun, Sauer Nathalie, Sauvey Christophe. Application of EM algorithm to hybrid flow shop scheduling problems with a special blocking. 2009. IEEE.

Zandieh M., Mozaffari E., Gholami M.. A robust genetic algorithm for scheduling realistic hybrid flexible flow line problems. Springer Science+Business Media, LLC 2009. *J Intell Manuf*. DOI 10.1007/s10845-009-0250-5

Zandieh M., S.M.T. Ghomi Fatemi, Hussein S.M. Moattar. An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematics and Computation* 180 (2006) 111–127. Science Direct.

Zandieh, M. Dorri Behrouz, Khamseh A. R.. Robust metaheuristics for group scheduling with sequence-dependent setup times in hybrid flexible flow shops. Springer-Verlag London Limited 2008.

Zhihong Jing, Zan Yanga, Takahiro Ito. Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem. *Int. J. Production Economics* 100 (2006) 322–334. Science Direct.

Zhan Yong, Qiu Changhua. Genetic Algorithm Application to the Hybrid Flow Shop Scheduling Problem. Proceedings of 2008 IEEE International Conference on Mechatronics and Automation.

Zhonghua Han, Haibo Shi, Chang Liu. Differential Evolution Algorithm for the Earliness/Tardiness Hybrid Flow-shop Scheduling Problem. 2009 Third International Symposium on Intelligent Information Technology Application.

8. ANEXOS

ANEXO 1. DISEÑO DE EXPERIMENTOS REALIZADO

Se realizó un total de 4 diseños experimentales para realizar la parametrización del algoritmo propuesto. Los tres primeros se realizaron con los datos de las instancias P20S2T01, P50S2T01 y PH1S2T1, los cuales se detallan a continuación:

INSTANCIA P20S2T01

No. de Estaciones	2																			
No. de Trabajos	20																			
No. de máquinas en la estación k	3	3																		
Tiempo de procesamiento del trabajo i en la en la máquina j de la estación 1	74	53	57	68	17	61	7	83	40	38	65	26	86	69	6	41	38	25	19	10
Tiempo de procesamiento del trabajo i en la en la máquina 1 de la estación 2	71	58	16	79	35	82	10	31	82	46	73	50	26	4	59	98	31	77	27	64

Tabla 10. Escenario del flowshop flexible estudiado para la instancia P20S2T01. Fuente: El autor.

Análogamente, las fechas de entrega de los trabajos se muestran en la **Tabla 11**, las cuales se generaron a través de la metodología expuesta para generar fechas de entrega:

Trabajos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Fecha de entrega	145	167	110	294	104	286	34	228	244	168	207	152	224	146	130	278	138	204	92	74

Tabla 11. Fechas de entrega para los trabajos de la instancia P20S2T01. Fuente: El autor.

Del mismo modo, se muestran los datos para las otras instancias. Por lo tanto:

INSTANCIA P50S2T01

No. de Estaciones	2																			
No. de Trabajos	50																			
No. de máquinas en la estación k	4	3																		
Tiempo de procesamiento del trabajo i en la en la máquina j de la estación 1	99	98	87	82	27	100	17	18	58	85	30	69	58	19	91	50	17	66	33	21
	19	63	78	51	4	45	19	37	40	13	67	49	24	29	27	64	15	15	94	1
	28	89	27	11	29	86	9	35	12	3										
Tiempo de procesamiento del trabajo i en la en la máquina 1 de la estación 2	70	23	97	22	54	59	53	95	70	89	87	11	46	52	60	38	15	61	97	79
	18	72	50	54	42	6	44	71	34	56	84	71	79	29	20	24	86	90	42	39
	42	10	60	44	85	47	16	69	16	94										

Tabla 12. Escenario del flowshop flexible estudiado para la instancia P50S2T01. Fuente: El autor.

Fecha de entrega del trabajo i	169	182	276	208	162	318	140	226	256	348	234	160	208	142	302	176	64	254	260	200
	74	270	256	210	92	77	126	216	148	138	302	240	206	116	94	176	202	210	272	80
	140	74	298	110	228	266	50	208	56	97										

Tabla 13. Fechas de entrega para los trabajos de la instancia P50S2T01. Fuente: El autor.

Por último, para completar los datos de esta primera ronda de análisis, se colocan los correspondientes al escenario de 100 trabajos, a saber:

INSTANCIA PH1S2T1

No. de Estaciones	2																			
No. de Trabajos	100																			
No. de máquinas en la estación k	4	4																		
Tiempo de procesamiento del trabajo i en la en la máquina k de la estación 1	65	48	87	1	86	17	39	37	66	29	58	41	28	51	58	20	87	2	82	39
	39	59	8	24	7	31	18	95	16	1	67	20	81	8	83	40	73	89	66	57
	8	25	52	90	65	7	54	18	29	99	31	75	59	3	96	98	31	83	57	89
	37	55	89	62	7	61	55	90	31	98	13	77	23	70	82	66	63	69	84	74
	92	96	27	83	92	70	3	18	23	70	93	26	80	79	14	60	27	33	9	54
Tiempo de procesamiento del trabajo i en la en la máquina 1 de la estación 2	95	65	79	28	53	28	90	97	34	25	7	71	84	82	83	17	17	97	32	30
	12	31	25	7	84	27	89	87	9	86	30	3	74	91	48	10	35	52	10	4
	40	98	60	92	81	12	58	36	25	62	84	79	50	41	13	44	60	94	97	9
	98	61	74	55	84	86	62	40	84	78	8	63	9	50	97	10	84	90	94	85
	51	51	43	62	85	42	26	81	35	9	26	48	75	23	69	49	65	73	40	49

Tabla 14. Escenario del flowshop flexible estudiado para la instancia PH1S2T1. Fuente: El autor.

Fecha de entrega del trabajo i	160	170	249	58	278	90	258	268	200	108	130	224	224	266	282	74	208	198	228	138
	102	180	66	62	182	116	214	364	50	174	194	46	310	198	262	100	216	282	152	122
	96	246	224	364	292	38	224	108	108	322	173	308	218	88	218	284	182	354	308	196
	270	232	326	234	182	294	234	260	230	352	42	280	64	240	358	152	294	318	356	318
	286	294	140	290	354	224	58	198	116	158	238	148	310	204	166	218	184	212	98	103

Tabla 15. Fechas de entrega para los trabajos de la instancia PH1S2T1. Fuente: El autor.

Con los datos anteriores, se realizaron tres diseños experimentales para realizar la parametrización del algoritmo propuesto; sin embargo, debido a la extensión de estos análisis, se presentan únicamente los resultados del diseño de experimentos para la instancia de 16 trabajos que se muestra a continuación y cuyos resultados, son similares a los realizados para las instancias mostradas anteriormente. Esta instancia se tomó de la tesis

de maestría titulada, “Programación de talleres de producción seriales híbridos (flexibles) con múltiples objetivos mediante metaheurísticas: colonias de hormigas”, desarrollada por Elyn Lizeth Solano Charris, en el año 2008. Dicha instancia se considera representativa, debido a que el problema allí planteado es un caso particular del problema bajo estudio en la presente investigación. Los datos de la instancia en mención se muestran a continuación:

No. de Estaciones	2																
No. de Trabajos	16																
No. de máquinas en la estación k	2	1															
Tiempo de procesamiento del trabajo i en la en la máquina j de la estación 1	5	9	3	5	3	2	9	3	7	5	3	2	4	4	9	7	
Tiempo de procesamiento del trabajo i en la en la máquina 1 de la estación 2	9	2	7	8	5	3	5	5	3	3	4	9	7	5	8	2	

Tabla 16. Escenario del flowshop flexible estudiado para la instancia seleccionada en el diseño de experimentos realizado. Fuente: El autor.

Análogamente, las fechas de entrega de los trabajos se muestran en la **Tabla 11**, las cuales se generaron a través de la metodología expuesta para generar fechas de entrega:

Trabajos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Fecha de entrega	14	17	15	26	16	10	28	16	15	16	14	22	22	18	34	9

Tabla 17. Fechas de entrega para los trabajos de la instancia seleccionada para el diseño de experimentos realizado. Fuente: El autor.

A continuación se muestran los datos obtenidos para esta instancia teniendo en cuenta el número de corridas establecido:

Datos obtenidos del diseño de experimentos realizado:

CORRIDA	ALFA	BETA	FEROMONAS	q ₀	RHO	MAKESPAN	TARDANZA TOTAL	NUM. TRAB. TARD.
1	1	1	1	1	1	91	557	16
2	1	1	1	1	0	88	554	13
3	1	1	1	1	-1	97	568	16
4	1	1	1	-1	1	92	569	11
5	1	1	1	-1	0	91	547	15
6	1	1	1	-1	-1	113	569	12
7	1	1	-1	1	1	91	565	12
8	1	1	-1	1	0	91	551	11
9	1	1	-1	1	-1	91	569	12
10	1	1	-1	-1	1	87	574	16
11	1	1	-1	-1	0	82	557	16
12	1	1	-1	-1	-1	83	602	14
13	1	-1	1	1	1	103	600	15
14	1	-1	1	1	0	109	577	12
15	1	-1	1	1	-1	100	554	16
16	1	-1	1	-1	1	96	581	11
17	1	-1	1	-1	0	84	560	16
18	1	-1	1	-1	-1	105	586	12
19	1	-1	-1	1	1	106	581	16
20	1	-1	-1	1	0	90	556	13
21	1	-1	-1	1	-1	96	553	10
22	1	-1	-1	-1	1	92	593	12
23	1	-1	-1	-1	0	87	550	14
24	1	-1	-1	-1	-1	98	548	16
25	-1	1	1	1	1	102	635	12
26	-1	1	1	1	0	99	612	13
27	-1	1	1	1	-1	103	617	10
28	-1	1	1	-1	1	121	668	16
29	-1	1	1	-1	0	103	622	16
30	-1	1	1	-1	-1	103	628	16
31	-1	1	-1	1	1	105	719	13
32	-1	1	-1	1	0	96	612	16
33	-1	1	-1	1	-1	99	613	16
34	-1	1	-1	-1	1	97	667	15
35	-1	1	-1	-1	0	99	613	12
36	-1	1	-1	-1	-1	112	617	11
37	-1	-1	1	1	1	108	642	16
38	-1	-1	1	1	0	91	627	13

CORRIDA	ALFA	BETA	FEROMONAS	q ₀	RHO	MAKESPAN	TARDANZA TOTAL	NUM. TRAB. TARD.
39	-1	-1	1	1	-1	103	615	13
40	-1	-1	1	-1	1	99	667	14
41	-1	-1	1	-1	0	103	611	13
42	-1	-1	1	-1	-1	103	653	12
43	-1	-1	-1	1	1	124	645	12
44	-1	-1	-1	1	0	119	665	16
45	-1	-1	-1	1	-1	103	618	12
46	-1	-1	-1	-1	1	111	687	15
47	-1	-1	-1	-1	0	103	612	13
48	-1	-1	-1	-1	-1	103	608	16
49	1	1	1	1	1	106	635	16
50	1	1	1	1	0	90	631	12
51	1	1	1	1	-1	96	630	13
52	1	1	1	-1	1	99	617	16
53	1	1	1	-1	0	99	625	16
54	1	1	1	-1	-1	99	617	15
55	1	1	-1	1	1	99	612	16
56	1	1	-1	1	0	103	607	14
57	1	1	-1	1	-1	118	614	15
58	1	1	-1	-1	1	86	553	16
59	1	1	-1	-1	0	117	576	14
60	1	1	-1	-1	-1	89	578	16
61	1	-1	1	1	1	85	554	11
62	1	-1	1	1	0	99	568	16
63	1	-1	1	1	-1	96	560	10
64	1	-1	1	-1	1	88	550	16
65	1	-1	1	-1	0	94	586	13
66	1	-1	1	-1	-1	91	597	11
67	1	-1	-1	1	1	85	555	16
68	1	-1	-1	1	0	115	573	16
69	1	-1	-1	1	-1	94	592	16
70	1	-1	-1	-1	1	117	608	16
71	1	-1	-1	-1	0	103	556	13
72	1	-1	-1	-1	-1	92	628	15
73	-1	1	1	1	1	84	558	13
74	-1	1	1	1	0	105	580	13
75	-1	1	1	1	-1	101	582	11
76	-1	1	1	-1	1	90	561	16
77	-1	1	1	-1	0	84	559	16

CORRIDA	ALFA	BETA	FEROMONAS	q ₀	RHO	MAKESPAN	TARDANZA TOTAL	NUM. TRAB. TARD.
78	-1	1	1	-1	-1	99	556	15
79	-1	1	-1	1	1	87	597	11
80	-1	1	-1	1	0	116	550	13
81	-1	1	-1	1	-1	98	642	14
82	-1	1	-1	-1	1	99	615	14
83	-1	1	-1	-1	0	112	619	13
84	-1	1	-1	-1	-1	124	656	14
85	-1	-1	1	1	1	96	611	16
86	-1	-1	1	1	0	99	625	14
87	-1	-1	1	1	-1	105	660	16
88	-1	-1	1	-1	1	96	610	10
89	-1	-1	1	-1	0	112	610	16
90	-1	-1	1	-1	-1	96	636	16
91	-1	-1	-1	1	1	91	642	15
92	-1	-1	-1	1	0	103	616	16
93	-1	-1	-1	1	-1	103	621	14
94	-1	-1	-1	-1	1	100	611	15
95	-1	-1	-1	-1	0	103	609	15
96	-1	-1	-1	-1	-1	124	619	11
97	1	1	1	1	1	118	614	15
98	1	1	1	1	0	86	553	16
99	1	1	1	1	-1	117	576	14
100	1	1	1	-1	1	89	578	16
101	1	1	1	-1	0	85	554	11
102	1	1	1	-1	-1	99	568	16
103	1	1	-1	1	1	96	560	10
104	1	1	-1	1	0	88	550	16
105	1	1	-1	1	-1	94	586	13
106	1	1	-1	-1	1	91	597	11
107	1	1	-1	-1	0	85	555	16
108	1	1	-1	-1	-1	115	573	16
109	1	-1	1	1	1	94	592	16
110	1	-1	1	1	0	117	608	16
111	1	-1	1	1	-1	103	556	13
112	1	-1	1	-1	1	92	628	15
113	1	-1	1	-1	0	84	558	13
114	1	-1	1	-1	-1	105	580	13
115	1	-1	-1	1	1	101	582	11
116	1	-1	-1	1	0	90	561	16

CORRIDA	ALFA	BETA	FEROMONAS	q ₀	RHO	MAKESPAN	TARDANZA TOTAL	NUM. TRAB. TARD.
117	1	-1	-1	1	-1	84	559	16
118	1	-1	-1	-1	1	99	556	15
119	1	-1	-1	-1	0	87	597	11
120	1	-1	-1	-1	-1	116	550	13
121	-1	1	1	1	1	98	642	14
122	-1	1	1	1	0	99	615	14
123	-1	1	1	1	-1	112	619	13
124	-1	1	1	-1	1	124	656	14
125	-1	1	1	-1	0	96	611	16
126	-1	1	1	-1	-1	99	625	14
127	-1	1	-1	1	1	105	660	16
128	-1	1	-1	1	0	96	610	10
129	-1	1	-1	1	-1	112	610	16
130	-1	1	-1	-1	1	96	636	16
131	-1	1	-1	-1	0	91	642	15
132	-1	1	-1	-1	-1	103	616	16
133	-1	-1	1	1	1	103	621	14
134	-1	-1	1	1	0	100	611	15
135	-1	-1	1	1	-1	103	609	15
136	-1	-1	1	-1	1	124	619	11
137	-1	-1	1	-1	0	96	610	16
138	-1	-1	1	-1	-1	112	621	13
139	-1	-1	-1	1	1	99	641	16
140	-1	-1	-1	1	0	100	661	16
141	-1	-1	-1	1	-1	105	643	14
142	-1	-1	-1	-1	1	114	638	15
143	-1	-1	-1	-1	0	99	626	13
144	-1	-1	-1	-1	-1	103	611	12
145	1	1	1	1	1	92	650	16
146	1	1	1	1	0	88	706	15
147	1	1	1	1	-1	87	752	15
148	1	1	1	-1	1	93	793	16
149	1	1	1	-1	0	88	624	15
150	1	1	1	-1	-1	92	654	16
151	1	1	-1	1	1	88	643	15
152	1	1	-1	1	0	92	566	16
153	1	1	-1	1	-1	88	638	15
154	1	1	-1	-1	1	84	841	14
155	1	1	-1	-1	0	92	521	16

CORRIDA	ALFA	BETA	FEROMONAS	q ₀	RHO	MAKESPAN	TARDANZA TOTAL	NUM. TRAB. TARD.
156	1	1	-1	-1	-1	88	666	15
157	1	-1	1	1	1	88	899	15
158	1	-1	1	1	0	88	815	15
159	1	-1	1	1	-1	88	617	15
160	1	-1	1	-1	1	92	859	16
161	1	-1	1	-1	0	93	722	16
162	1	-1	1	-1	-1	92	551	16
163	1	-1	-1	1	1	87	689	15
164	1	-1	-1	1	0	88	620	15
165	1	-1	-1	1	-1	92	547	16
166	1	-1	-1	-1	1	92	580	16
167	1	-1	-1	-1	0	92	540	16
168	1	-1	-1	-1	-1	92	614	16
169	1	1	1	1	1	87	866	15
170	1	1	1	1	0	87	866	15
171	1	1	1	1	-1	92	895	16
172	1	1	1	-1	1	84	960	14
173	1	1	1	-1	0	93	747	16
174	1	1	1	-1	-1	92	773	16
175	1	1	-1	1	1	88	632	15
176	1	1	-1	1	0	88	645	15
177	1	1	-1	1	-1	92	561	16
178	1	1	-1	-1	1	92	644	16
179	1	1	-1	-1	0	88	650	15
180	1	1	-1	-1	-1	88	792	15
181	1	-1	1	1	1	93	750	16
182	1	-1	1	1	0	92	852	16
183	1	-1	1	1	-1	93	761	16
184	1	-1	1	-1	1	87	751	15
185	1	-1	1	-1	0	88	683	15
186	1	-1	1	-1	-1	88	720	15
187	1	-1	-1	1	1	87	698	15
188	1	-1	-1	1	0	92	566	16
189	1	-1	-1	1	-1	87	699	15
190	1	-1	-1	-1	1	88	736	15
191	1	-1	-1	-1	0	88	606	15
192	1	-1	-1	-1	-1	98	730	17

Tabla 168. Matriz de datos del Diseño Experimental realizado. Fuente: El autor.

Inicialmente, se construyó la tabla ANOVA (ver Tabla 19) para la variable de respuesta makespan con el fin de analizar qué factores y en qué medida afectan a esta variable. Se observa que el nivel de significancia del modelo corregido es inferior al p-value de 0.05, por lo que se rechaza la hipótesis nula. Esto quiere decir que existe por lo menos un factor que causa un efecto en la variable de respuesta. También indica que las medias de los tratamientos no son iguales, aunque no dice específicamente cuales difieren entre sí. Sin embargo, esto se puede ver más claramente con una prueba de comparaciones múltiples de Tukey (**Tabla 20**) donde se muestran las medias de los grupos de subconjuntos homogéneos, basadas en las medias observadas. La prueba de Tukey muestra que para el único factor de tres niveles considerado, rho, hay una diferencia significativa en las medias de los niveles 0 y 1 y 0 y -1 para el nivel de significancia del 95%, en donde el mejor valor de rho, o el que proporciona mejores valores del makespan es rho=0.01 (nivel 0). Las medias de los niveles 1 y -1 no presentan diferencias estadísticamente significativas. No se realizaron pruebas de comparaciones múltiples para alfa, debido a que sólo tiene dos niveles o grupos, por lo que se deduce que las medias de estos no son iguales ya que este factor genera un efecto en la variable de respuesta. Adicionalmente, En la tabla ANOVA se observa que para la variable makespan, los factores Alfa y Rho generan un efecto, debido a que sus p-values son menores al nivel de significancia establecido del 0.05. Esto quiere decir que la elección más conveniente del valor de Alfa, depende del nivel en que esté Rho y viceversa. Del mismo modo, la interacción de segundo orden entre los factores Feromonas y Rho también presenta una influencia para la variable de respuesta considerada, lo mismo que la interacción de tercer orden Alfa, q_0 y Rho. Lo anterior se puede observar mejor en las **Figuras 10 a) y b)** donde se muestran las gráficas de efectos principales y de las interacciones entre los factores considerados sobre la variable de respuesta estudiada.

Pruebas de los efectos inter-sujetos

Variable dependiente: MAKESPAN

Origen	Suma de cuadrados tipo III ^a	gl	Media cuadrática	F	Sig.
Modelo corregido		47	139,989	2,350	,000
Intersección	1423050,174	1	1423050,174	23886,143	,000
ALFA	1350,563	1	1350,563	22,669	,000
BETA	9,507	1	9,507	,160	,690
FEROMONAS	2,007	1	2,007	,034	,855
K	101,674	1	101,674	1,707	,195
RHO	1574,681	2	787,340	13,216	,000
ALFA * BETA	12,840	1	12,840	,216	,644
ALFA * FEROMONAS	,340	1	,340	,006	,940
ALFA * K	8,507	1	8,507	,143	,706
ALFA * RHO	211,542	2	105,771	1,775	,175
BETA * FEROMONAS	166,840	1	166,840	2,800	,097
BETA * K	14,063	1	14,063	,236	,628
BETA * RHO	223,097	2	111,549	1,872	,159
FEROMONAS * K	5,062	1	5,062	,085	,771
FEROMONAS * RHO	453,347	2	226,674	3,805	,026
K * RHO	106,264	2	53,132	,892	,413
ALFA * BETA * FEROMONAS	158,340	1	158,340	2,658	,106
ALFA * BETA * K	82,507	1	82,507	1,385	,242
ALFA * BETA * RHO	49,681	2	24,840	,417	,660
ALFA * FEROMONAS * K	207,840	1	207,840	3,489	,065
ALFA * FEROMONAS * RHO	284,597	2	142,299	2,389	,097
ALFA * K * RHO	458,514	2	229,257	3,848	,025
BETA * FEROMONAS * K	105,063	1	105,063	1,763	,187
BETA * FEROMONAS * RHO	126,764	2	63,382	1,064	,349
BETA * K * RHO	86,292	2	43,146	,724	,487
FEROMONAS * K * RHO	195,542	2	97,771	1,641	,199
ALFA * BETA * FEROMONAS * K	,340	1	,340	,006	,940
ALFA * BETA * FEROMONAS * RHO	99,847	2	49,924	,838	,436
ALFA * BETA * K * RHO	4,597	2	2,299	,039	,962
ALFA * FEROMONAS * K * RHO	125,514	2	62,757	1,053	,353
BETA * FEROMONAS * K * RHO	38,792	2	19,396	,326	,723
ALFA * BETA * FEROMONAS * K * RHO	314,931	2	157,465	2,643	,076
Error	5719,333	96	59,576		
Total	1435349,000	144			
Total corregida	12298,826	143			

R cuadrado = ,535 (R cuadrado corregida = ,307)

Tabla 17. Tabla ANOVA para la variable de respuesta MAKESPAN. Fuente: El autor.

MAKE SPAN

DHS de Tukey

RHO	N	Subconjunto	
		1	2
,00	48	94,7500	
1,00	48		101,3958
-1,00	48		102,0833
Sig.		1,000	,900

Tabla 20. Prueba de comparaciones múltiples de Tukey en donde se observa que la diferencia de medias en los niveles para el factor rho, se debe al nivel 0, correspondiente a un valor de 0.01. Fuente: El autor.

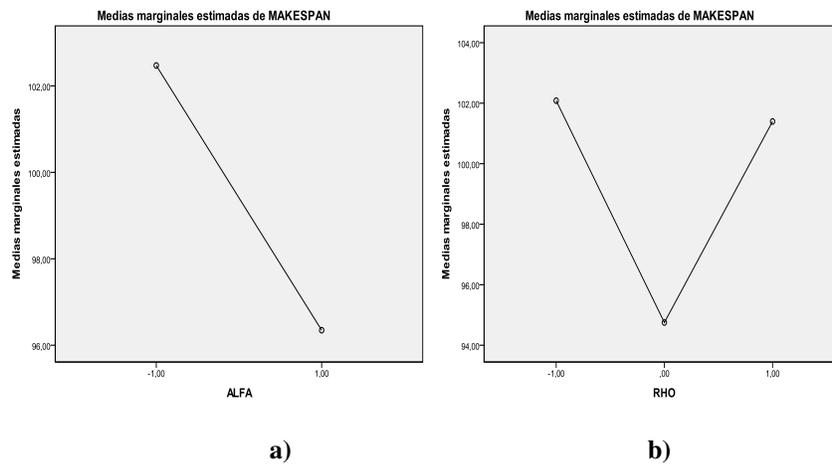


Figura 10. En a) se presenta el efecto del factor Alfa sobre la variable de respuesta Makespan y en b) se presenta el efecto del factor Rho sobre dicha variable. Fuente: El autor.

Estas gráficas muestran que para el caso del factor alfa, el valor de 2 (nivel 1) proporciona de manera muy significativa mejores valores para el makespan que el valor de 1 (nivel -1). Del mismo modo, para el factor rho, valor de 0.01 (nivel 0) presenta mejores resultados que los otros valores de rho correspondientes a los niveles de 1 y -1 respectivamente. Sin embargo, para llegar a conclusiones más contundentes es necesario estudiar el efecto de las interacciones de los factores sobre la variable de respuesta de modo que se pueda elegir la

mejor combinación de parámetros que proporcionen mejores valores para dicha variable. A continuación se muestra la interacción de segundo orden encontrada en el ANOVA de la Tabla 1. En la **Figura 11a)** se observa la interacción entre los factores Feromonas y Rho en donde se evidencia que cuando el valor de la cantidad feromonas es de 0.001 (nivel -1), hay variaciones considerables del valor de la variable makespan para los diferentes niveles del factor Rho, en donde los mejores (menores) valores para el makespan se obtiene para Rho 0.01 (nivel 0). Lo mismo sucede cuando el valor de la cantidad de feromonas es de 0.01 (nivel 1). Para esta interacción se muestra claramente que el makespan tiene mejores valores cuando la cantidad de feromonas es de 0.1 y el parámetro rho tiene un valor de 0.01.

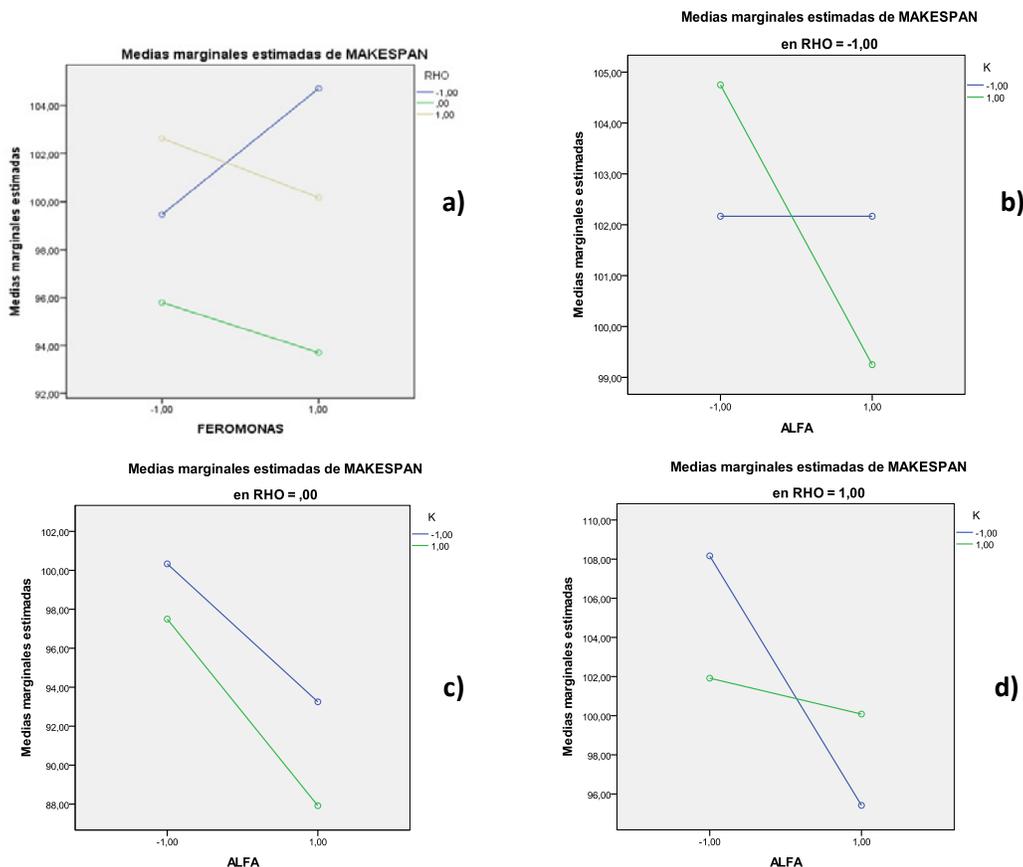


Figura 11. En a) se presenta la interacción de segundo orden entre los factores feromonas y rho y su efecto sobre el Makespan. En c) y d) se presenta la interacción de tercer orden y su efecto sobre el makespan, entre los factores Alfa, q_0 y Rho para los diferentes niveles de este último. Fuente: El autor.

Por otra parte, en las **Figuras 11 b), c) y d)** en donde se muestran las gráficas correspondientes a la interacción de tercer orden entre los factores alfa, q_0 , y rho, se observa un fenómeno interesante: En a), cuando rho tiene el valor de 0.001 (nivel -1) el valor del makespan es mayor o empeora para un valor de alfa de 1 (nivel -1) y un valor de q_0 de 0.8 (nivel 1), mientras que para un valor de alfa de 2 (nivel 1) y el correspondiente valor de q_0 , el makespan mejora significativamente o disminuye. Además, cuando el valor de q_0 es $\log(\text{iter}/\text{num_iter})$ (nivel -1), el makespan tiene un comportamiento constante, para los valores de alfa considerados. Para este caso se obtiene un mejor valor del makespan para $\text{alfa}=2$ y $q_0=0.8$, los cuales proporcionan un valor del makespan de 99 para la instancia que se está considerando.

Por otro lado, en c) se muestra que para un valor de rho de 0.01 (nivel 0), el valor del makespan es mejor cuando el valor de q_0 es de 0.8 (nivel 1), independientemente del valor de alfa. Sin embargo, se obtienen mejores valores del makespan cuando alfa es igual a 2 (nivel 1). Por lo tanto, para este caso ($\text{rho}=0.01$) también la mejor combinación de parámetros es $\text{alfa}=2$ y $q_0=0.8$, que proporciona un valor del makespan de 88 (inferior al del inciso b).

Por último, en d) se observa que cuando rho es 0.1 (nivel 1), el valor del makespan es mejor cuando q_0 es $\log(\text{iter}/\text{num_iter})$ (nivel -1) y alfa es 2 (nivel 1) obteniendo un valor del makespan de 96. Sin embargo, este es mayor al que se obtuvo en el inciso c. Por lo tanto, por lo observado en las gráficas anteriores, se concluye que para la variable de respuesta analizada (makespan) la combinación de valores de los parámetros o factores que proporciona un mejor valor, se da para $\text{alfa}=2$, $\text{rho}=0.01$, $\text{feromonas}=0.1$ y $q_0=0.8$, aunque para la cantidad de feromonas de 0.01 se obtuvieron también buenos valores para el makespan. Estos resultados, son consecuentes con las conclusiones obtenidas a partir del análisis de las gráficas de efectos principales para alfa y rho (figura 8 a y b) y de la prueba de comparaciones múltiples de tukey para rho. Adicionalmente, se concluye que el valor

de beta poco incide en el makespan y que por lo tanto para esta variable se podría utilizar cualquier valor; es decir 2 o 3. Sin embargo, como se están analizando varios objetivos o variables de respuesta simultáneamente junto con el makespan, no se puede llegar a una conclusión definitiva hasta que se analicen las otras dos variables de respuesta consideradas (tardanza total y número de trabajos tardíos). Inclusive, la mejor elección de los parámetros puede depender también de la ponderación o importancia relativa que se le otorgue a los objetivos en cuestión, a la hora de ejecutar la herramienta computacional que resuelve el problema bajo estudio en la presente investigación.

Comprobación de los supuestos del diseño experimental

A continuación se corroborarán los supuestos del diseño experimental realizado los cuales proporcionan validez al análisis descrito y soportan las conclusiones obtenidas en el mismo. Como se dijo previamente, se deben verificar los supuestos del diseño experimental para soportar la validez del modelo. A continuación se presentan dichos supuestos.

- Normalidad de residuos tipificados
- Homogeneidad de varianzas
- Correlación en el tiempo

Normalidad: Mediante la prueba de Kolmogorov-Smirnov se realiza la siguiente prueba de hipótesis para comprobar la normalidad de los residuos:

H_0 : Los residuos tipificados de la muestra provienen de una población con distribución normal.

H_1 : Los residuos tipificados de la muestra no provienen de una población con distribución normal.

Si significancia (valor p) < 0.05: Rechazo H_0

Si significancia (valor p) > 0.05 No Rechazo H_0

Prueba de Kolmogorov-Smirnov para una muestra

			Residuo estandarizado para MAKESPAN
N			144
	a,b	Media	,0000
		Desviación típica	,81935
Diferencias más extremas		Absoluta	,083
		Positiva	,083
		Negativa	-,059
Z de Kolmogorov-Smirnov			,998
Sig. asintót. (bilateral)			,272

La distribución de contraste es la Normal.

Se han calculado a partir de los datos.

Tabla 218. Prueba normalidad de Kolmorov-Smirnov. Fuente: El autor.

Al realizar esta prueba se tiene en la tabla 3 un p-value de 0.272 el cual es mayor al nivel de significancia del 0.05, se concluye que no se rechaza la hipótesis nula, lo que quiere decir que los residuos tienen un comportamiento normal con media relativamente cercana a cero y varianza constante.

Esto quiere decir que el ANOVA es robusto, ya que tener residuos normales indica que lo que queda después del modelo es ruido blanco, y que la variabilidad que se presenta en el experimento es debido al error aleatorio inherente al sistema, el cual no se puede eliminar completamente, pero que no influye de manera significativa en el resultado del experimento. Se realiza a continuación la prueba de correlación con el tiempo:

Prueba de correlación: mediante la prueba de correlación se pueden comprobar las siguientes hipótesis:

H_0 : Los residuales no están correlacionados con el tiempo

H_1 : Los residuales están asociados con el tiempo.

Si significancia (valor p) < 0.05: Rechazo H_0

Si significancia (valor p) > 0.05: No Rechazo H_0

Correlaciones

		Residuo estandarizado para MAKESPAN	ID
Residuo estandarizado para MAKESPAN	Correlación de Pearson	1	,066
	Sig. (bilateral)		,431
	N	144	144
ID	Correlación de Pearson	,066	1
	Sig. (bilateral)	,431	
	N	144	144

Tabla 19. Prueba de correlación de los residuos tipificados para la variable makepan. Fuente: El autor.

Esta prueba arroja un p-value de 0.431; como se tiene una significancia mayor que 0.05, entonces no se rechaza H_0 . Por lo tanto, se concluye que no hay suficiente evidencia estadística para aseverar de que el tiempo u orden de la corrida experimental es un factor importante en la ejecución del experimento; en otras palabras, no existe una correlación de los errores en el tiempo.

Homogeneidad de varianzas: Por último, se realiza una prueba de Levene para verificar si las varianzas de los grupos (tratamientos) son iguales; por lo tanto:

H_0 : Las varianzas de los grupos son iguales

H_1 : Las varianzas de los grupos son iguales

Si significancia (valor p) < 0.05 Rechazo H_0

Si significancia (valor p) > 0.05: No Rechazo H_0

**Contraste de Levene sobre la igualdad de las
varianzas error^a**

Variable dependiente: MAKESPAN

F	gl1	gl2	Sig.
3,617	47	96	,000

Contrasta la hipótesis nula de que la varianza
error de la variable dependiente es igual a lo
largo de todos los grupos.

Tabla 23. Prueba de Levene de igualdad de varianzas para el makespan. Fuente: El autor.

Al realizar la prueba de Levene el nivel de significancia obtenido es menor que 0.05 (relativamente cercano a cero), por lo que se rechaza la hipótesis nula y se concluye que las varianzas de los grupos son diferentes; es decir, existe una diferencia estadísticamente significativa entre los grupos. Sin embargo, el incumplimiento de este supuesto no representa un problema grave, debido a que los tamaños de muestra son iguales para las observaciones de los factores con el mismo número de niveles.

Por lo tanto, debido a que se cumplieron los supuestos más importantes del diseño experimental, se puede dar fe estadística de que las conclusiones obtenidas para la variable de respuesta makepan son válidas, lo cual proporciona un buen soporte estadístico al análisis realizado. A continuación se realiza el estudio correspondiente a la variable tardanza. Inicialmente se muestra la tabla ANOVA:

Pruebas de los efectos inter-sujetos

Variable dependiente: TARDANZA

Origen	Suma de cuadrados tipo III	gl	Media cuadrática	F	Sig.
Modelo corregido	^a	47	3429,870	10,005	,000
Intersección		1		152546,608	,000
ALFA	123201,000	1	123201,000	359,368	,000
BETA	342,250	1	342,250	,998	,320
FEROMONAS	78,028	1	78,028	,228	,634
K	10,028	1	10,028	,029	,865
RHO	20204,514	2	10102,257	29,468	,000
ALFA * BETA	462,250	1	462,250	1,348	,248
ALFA * FEROMONAS	354,694	1	354,694	1,035	,312
ALFA * K	633,361	1	633,361	1,847	,177
ALFA * RHO	981,792	2	490,896	1,432	,244
BETA * FEROMONAS	312,111	1	312,111	,910	,342
BETA * K	1089,000	1	1089,000	3,177	,078
BETA * RHO	1416,542	2	708,271	2,066	,132
FEROMONAS * K	1708,444	1	1708,444	4,983	,028
FEROMONAS * RHO	881,931	2	440,965	1,286	,281
K * RHO	2695,264	2	1347,632	3,931	,023
ALFA * BETA * FEROMONAS	348,444	1	348,444	1,016	,316
ALFA * BETA * K	136,111	1	136,111	,397	,530
ALFA * BETA * RHO	369,542	2	184,771	,539	,585
ALFA * FEROMONAS * K	1225,000	1	1225,000	3,573	,062
ALFA * FEROMONAS * RHO	942,931	2	471,465	1,375	,258
ALFA * K * RHO	797,931	2	398,965	1,164	,317
BETA * FEROMONAS * K	96,694	1	96,694	,282	,597
BETA * FEROMONAS * RHO	111,847	2	55,924	,163	,850
BETA * K * RHO	205,125	2	102,563	,299	,742
FEROMONAS * K * RHO	135,847	2	67,924	,198	,821
ALFA * BETA * FEROMONAS * K	30,250	1	30,250	,088	,767
ALFA * BETA * FEROMONAS * RHO	1289,347	2	644,674	1,880	,158
ALFA * BETA * K * RHO	65,181	2	32,590	,095	,909
ALFA * FEROMONAS * K * RHO	16,125	2	8,062	,024	,977
BETA * FEROMONAS * K * RHO	414,264	2	207,132	,604	,549
ALFA * BETA * FEROMONAS * K * RHO	648,042	2	324,021	,945	,392
Error	32911,333	96	342,826		
Total		144			
Total corregida	194115,222	143			

R cuadrado = ,830 (R cuadrado corregida = ,747)

Tabla 24. Tabla ANOVA para la variable de respuesta Tardanza Total. Fuente: El autor.

Para este caso, la tabla ANOVA muestra que el modelo corregido tiene una significancia inferior al p-value de 0.05, por lo que se concluye que al menos un factor causa un efecto en la variable tardanza total. Del mismo modo que en el caso del makespan, también se infiere que las diferencias poblacionales de los grupos de tratamientos no son iguales, aunque no dice explícitamente cuales son diferentes entre sí. Para eso, se utiliza una prueba de comparaciones múltiples de tukey (**Tabla 15**), en donde se muestra que no hay diferencia estadísticamente significativa para los valores de $\rho=0.001$ y $\rho=0.01$, los cuales proporcionan los mejores valores para la tardanza para un nivel de significancia del 95%. Por otro lado, se muestra en la tabla ANOVA, que los factores alfa y rho generan un efecto en la variable de respuesta considerada, lo mismo que las interacciones de segundo orden feromonas- q_0 y q_0 -rho. Esto se muestra mejor en la figura 9, que muestra las gráficas de efectos principales y de interacciones para los factores mencionados y la variable de respuesta analizada.

TARDANZA

DHS de Tukey

RHO	N	Subconjunto	
		1	2
,00	48	593,6458	
-1,00	48	594,8958	
1,00	48		619,3750
Sig.		,942	1,000

Tabla 20. Prueba de comparaciones múltiples de Tukey en donde se observa que la diferencia de medias en los niveles para el factor rho, se debe al nivel 0, correspondiente a un valor de 0.01. Fuente: El autor.

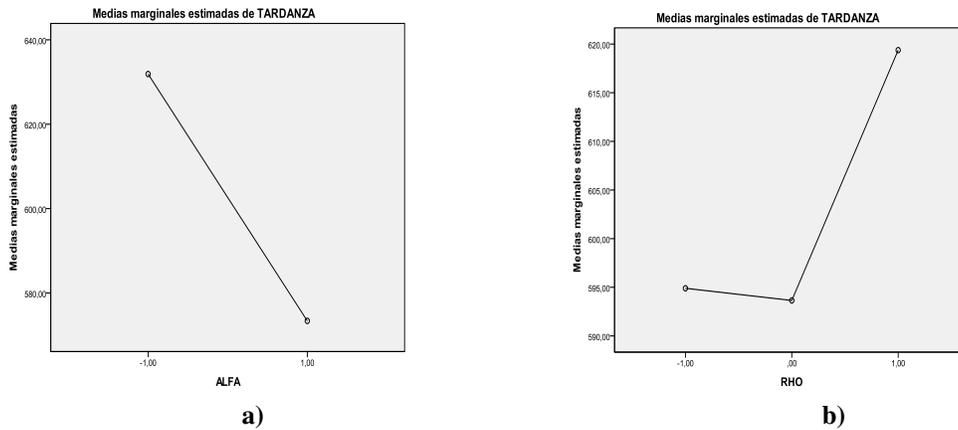


Figura 12. En a) se presenta el efecto del factor Alfa sobre la variable de respuesta Tardanza y en b) se presenta el efecto del factor Rho sobre dicha variable. Fuente: El autor.

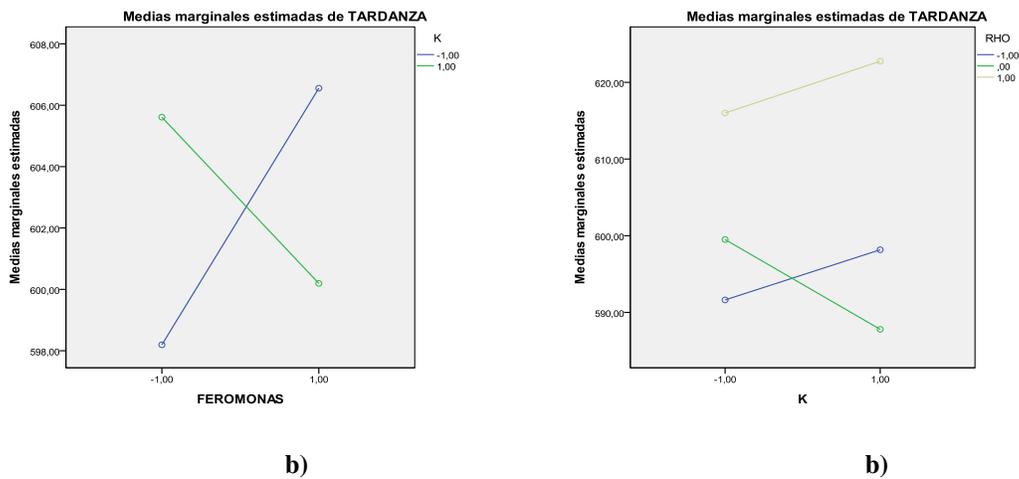


Figura 13. En a) se presenta la interacción de segundo orden entre los factores feromonas y q_0 y su efecto sobre la Tardanza total. En b) se presenta la interacción de segundo orden entre q_0 y rho y su efecto sobre la Tardanza total. Fuente: El autor.

La **Figura 12 a)** se observa que el valor de $\alpha=2$ (nivel 1), proporciona los mejores valores para la tardanza total. Análogamente, en b) se muestra que el valor de $\rho=0.01$ (nivel 0), presenta los mejores valores para la tardanza total. Sin embargo, el valor de $\rho=0.001$ (nivel -1) tiene valores de la tardanza total muy cercanos a aquellos generados

con $\rho=0.01$. Del mismo modo que con el makespan, se deben tener en cuenta las interacciones para poder establecer conclusiones más precisas y fundamentadas que permitan determinar la combinación de valores de los factores estudiados que optimicen la variable de respuesta analizada; esto se observa en la **figura 13 a)** donde se muestra la interacción entre la cantidad de feromonas y q_0 . Aquí se observa que el mejor valor para la tardanza total se obtiene con un valor de feromonas de 0.001 (nivel -1) y un valor de q_0 de $\log(\text{iter}/\text{num_iter})$ (nivel -1). Por otro lado, en b) se analiza el efecto de la interacción entre q_0 y ρ , donde se observa nuevamente que la tardanza total disminuye con un valor de ρ de 0.01 (nivel 0) -que corrobora lo observado en la gráfica de efectos principales (figura 13 b) y en la prueba de comparaciones múltiples de tukey- y un valor de q_0 de 0.8 (nivel 1). Hasta este punto, se podría decir que la mejor elección de ρ está determinada por el valor de 0.01, puesto que es la que mejor desempeño proporciona en dos de las tres variables de respuesta consideradas. También se podría decir lo mismo para alfa, que para el valor de 2 presenta los mejores valores para las dos variables de respuesta analizadas y para el caso de la tardanza total no presenta interacciones significativas con los otros factores. Del mismo modo que para el makespan, el valor de beta parece ser irrelevante hasta este punto, de entre los valores considerados. Sin embargo, para el factor feromonas y q_0 la conclusión no es tan obvia, puesto que para el makespan los valores de estos factores que presentan un mejor desempeño son 0.1 y 0.8, mientras que para la tardanza total gráfica correspondiente a la interacción entre estos factores muestra precisamente que un valor de 0.01 de feromonas y de $\log(\text{iter}/\text{num_iter})$ disminuyen la tardanza total obteniendo un valor para esta última de 598. Sin embargo, la interacción q_0 y ρ muestra que la tardanza total se disminuye para un valor de q_0 de 0.8, obteniendo un valor de la tardanza por debajo de 590, los cuales son inferiores a los de $q_0 = \log(\text{iter}/\text{num_iter})$. Por lo tanto, en apariencia, este último resultado podría inclinar la balanza a favor de $q_0=0.8$, pero faltaría definir el valor para las feromonas. En el inciso anterior se observó para el makespan que un valor de 0.1 proporcionaba las mejores soluciones, pero también se obtenían buenos resultados para un valor de 0.01. Para el caso de la tardanza total, se obtienen mejores resultados para la

cantidad de feromonas de 0.01 que para 0.1. Hasta este punto, se podría decir que un valor de 0.01 para el nivel de feromonas sería el más conveniente, porque tiene muy buenos resultados para los dos objetivos considerados y sus interacciones. Sin embargo, es preferible realizar el análisis para la última variable de respuesta considerada (número de trabajos tardíos) antes de establecer una conclusión definitiva para este valor, lo mismo que para el de q_0 .

A continuación se realiza la comprobación de los supuestos del diseño experimental para el validar el modelo planteado. Dicha comprobación se realiza de la misma manera que para el makespan. Por lo tanto:

Normalidad:

H_0 : Los residuos tipificados de la muestra provienen de una población con distribución normal.

H_1 : Los residuos tipificados de la muestra no provienen de una población con distribución normal.

Si significancia (valor p) < 0.05: Rechazo H_0

Si significancia (valor p) > 0.05 No Rechazo H_0

Prueba de Kolmogorov-Smirnov para una muestra

		Residuo estandarizado para TARDANZA
N		144
	a,b	
	Media	,0000
	Desviación típica	,81935
Diferencias más extremas	Absoluta	,090
	Positiva	,090
	Negativa	-,048
Z de Kolmogorov-Smirnov		1,079
Sig. asintót. (bilateral)		,194

La distribución de contraste es la Normal.

Se han calculado a partir de los datos.

Tabla 21. Prueba de Kolmogorv-Smirnov para normalidad de residuos tipificados de la variable Tardanza Total. Fuente: El autor.

Al realizar esta prueba se tiene un p-value de 0.194 el cual es mayor al nivel de significancia del 0.05, se concluye que no se rechaza la hipótesis nula, lo que quiere decir que los residuos tienen un comportamiento normal con media relativamente cercana a cero y varianza constante.

Correlación:

H_0 : Los residuales no están correlacionados con el tiempo

H_1 : Los residuales están asociados con el tiempo.

Si significancia (valor p) < 0.05: Rechazo H_0

Si significancia (valor p) > 0.05: No Rechazo H_0

Correlaciones

		Residuo estandarizado para TARDANZA	ID
Residuo estandarizado para TARDANZA	Correlación de Pearson	1	-,027
	Sig. (bilateral)		,752
	N	144	144
ID	Correlación de Pearson	-,027	1
	Sig. (bilateral)	,752	
	N	144	144

Tabla 27. Prueba de Correlación de los residuales de la variable Tardanza Total. Fuente: El autor.

Esta prueba arroja un p-value de 0.752; como se tiene una significancia mayor que 0.05, entonces no se rechaza H_0 . Por lo tanto, se concluye que no hay suficiente evidencia estadística para aseverar de que el tiempo u orden de la corrida experimental es un factor importante en la ejecución del experimento; en otras palabras, no existe una correlación de los errores en el tiempo.

Igualdad de varianzas:

H_0 : Las varianzas de los grupos son iguales

H_1 : Las varianzas de los grupos son iguales

Si significancia (valor p) < 0.05 Rechazo H_0

Si significancia (valor p) > 0.05: No Rechazo H_0

**Contraste de Levene sobre la igualdad de las
varianzas error^a**

Variable dependiente: TARDANZA

F	gl1	gl2	Sig.
3,950	47	96	,000

Contrasta la hipótesis nula de que la varianza error de la variable dependiente es igual a lo largo de todos los grupos.

Tabla 28. Prueba de Levene para igualdad de varianzas de la variable Tardanza total. Fuente: El autor.

Al realizar la prueba de Levene el nivel de significancia obtenido es menor que 0.05 (relativamente cercano a cero), por lo que se rechaza la hipótesis nula y se concluye que las varianzas de los grupos son diferentes; es decir, existe una diferencia estadísticamente significativa entre los grupos. Sin embargo, como en el caso del makespan, el incumplimiento de este supuesto no representa una violación grave, puesto que el ANOVA es robusto.

Por lo tanto, como se comprobaron los supuestos del diseño experimental, se corrobora la validez del diseño experimental realizado. Del mismo modo, así como para las dos variables de respuesta estudiadas anteriormente, a continuación se muestra el análisis de varianza para el número de trabajos tardíos:

Pruebas de los efectos inter-sujetos

Variable dependiente: TRAB_TARD

Origen	Suma de cuadrados tipo III	gl	Media cuadrática	F	Sig.
Modelo corregido ^a		47	3,631	,961	,551
Intersección	27944,694	1	27944,694	7397,125	,000
ALFA	3,361	1	3,361	,890	,348
BETA	1,000	1	1,000	,265	,608
FEROMONAS	1,778	1	1,778	,471	,494
K	1,778	1	1,778	,471	,494
RHO	4,222	2	2,111	,559	,574
ALFA * BETA	2,778	1	2,778	,735	,393
ALFA * FEROMONAS	9,000	1	9,000	2,382	,126
ALFA * K	9,000	1	9,000	2,382	,126
ALFA * RHO	,222	2	,111	,029	,971
BETA * FEROMONAS	,028	1	,028	,007	,932
BETA * K	,028	1	,028	,007	,932
BETA * RHO	27,167	2	13,583	3,596	,031
FEROMONAS * K	3,361	1	3,361	,890	,348
FEROMONAS * RHO	,389	2	,194	,051	,950
K * RHO	5,056	2	2,528	,669	,515
ALFA * BETA * FEROMONAS	12,250	1	12,250	3,243	,075
ALFA * BETA * K	,250	1	,250	,066	,798
ALFA * BETA * RHO	,389	2	,194	,051	,950
ALFA * FEROMONAS * K	38,028	1	38,028	10,066	,002
ALFA * FEROMONAS * RHO	3,167	2	1,583	,419	,659
ALFA * K * RHO	9,500	2	4,750	1,257	,289
BETA * FEROMONAS * K	4,000	1	4,000	1,059	,306
BETA * FEROMONAS * RHO	,889	2	,444	,118	,889
BETA * K * RHO	3,389	2	1,694	,449	,640
FEROMONAS * K * RHO	7,056	2	3,528	,934	,397
ALFA * BETA * FEROMONAS * K	1,778	1	1,778	,471	,494
ALFA * BETA * FEROMONAS * RHO	2,000	2	1,000	,265	,768
ALFA * BETA * K * RHO	1,167	2	,583	,154	,857
ALFA * FEROMONAS * K * RHO	5,389	2	2,694	,713	,493
BETA * FEROMONAS * K * RHO	10,167	2	5,083	1,346	,265
ALFA * BETA * FEROMONAS * K * RHO	2,056	2	1,028	,272	,762
Error	362,667	96	3,778		
Total	28478,000	144			
Total corregida	533,306	143			

R cuadrado = ,320 (R cuadrado corregida = -,013)

Tabla 29. Tabla ANOVA para la variable de respuesta número de trabajos tardíos. Fuente: El autor.

La tabla ANOVA para el número de trabajos tardíos muestra que el nivel de significancia del modelo corregido es 0.551, el cual es mayor al p-value de 0.05 por lo que no se rechaza la hipótesis nula. Esto quiere decir que no hay factores que a nivel individual causen un efecto en la variable de respuesta. Los correspondientes p-values para los factores analizados, corroboran esta afirmación, puesto que todos son mayores a 0.05. Esto también quiere decir que las medias de los tratamientos son iguales, lo cual se observa en la figura 12 de comparaciones múltiples de tukey donde para la variable de respuesta rho, se observa que no hay diferencias estadísticamente significativas en los grupos considerados. Sin embargo, el análisis de varianza muestra que hay interacciones estadísticamente significativas, una de segundo orden entre los factores beta y rho, y otra de tercer orden entre los factores alfa, feromonas y q_0 , las cuales se pueden ver en en la figura 12 a), b) y c).

TRAB_TARD

DHS de Tukey

RHO	N	Subconjunto
		1
-1,00	48	13,7083
,00	48	13,9583
1,00	48	14,1250
Sig.		,547

Tabla 30. Prueba de comparaciones múltiples de Tukey para Rho. Se observa que las medias de los grupos no presentan diferencias estadísticamente significativas. Fuente: El autor.

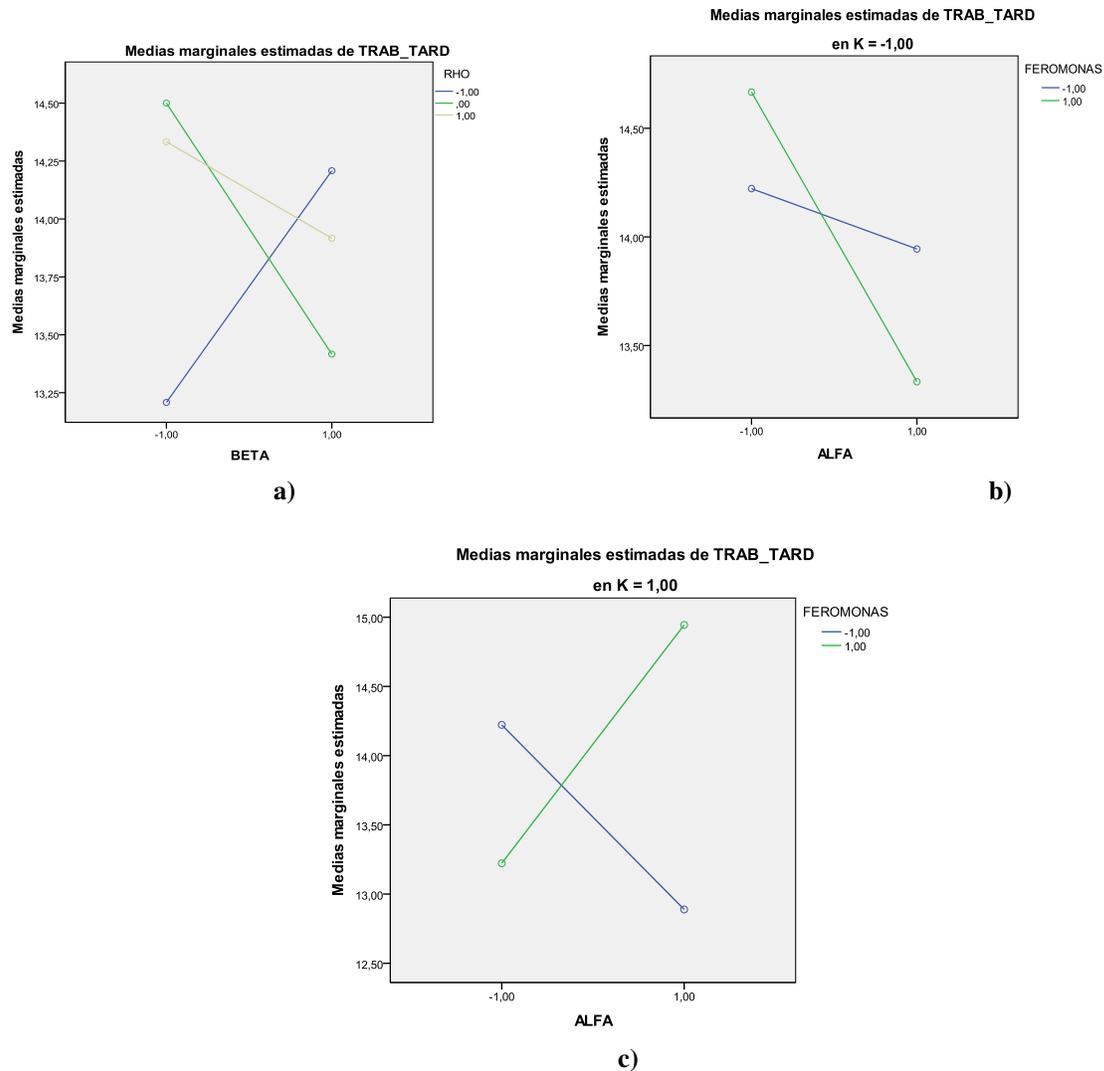


Figura 14. Gráfico de interacciones para Beta y Rho y para Alfa, Feromonas y q_0 . Fuente: El autor.

La **Figura 14** muestra en a) el efecto de la interacción entre los factores beta y rho, donde se observa que el valor de beta de 2 (nivel -1), el número de trabajos tardíos es menor. Sin embargo, esto se da para un valor de rho de 0.001 (nivel -1) ya que para el valor de beta considerado, un valor de rho de 0.01 (nivel 0) que es el valor de rho que proporciona mejores resultados para las otras variables de respuesta analizadas aumenta el valor del número de trabajos tardíos. Sin embargo, para un valor de beta de 3 (nivel 1) y un valor de

rho de 0.01 (nivel 0) se obtiene un valor mucho mejor que con la combinación beta=2 y rho=0.01, aunque desmejora un poco con relación a la combinación de beta=2 y rho=0.001. Sin embargo, para ser consistente con los resultados previos, en donde el mejor valor para rho es de 0.01, es mejor utilizar un valor de beta de 3. Puesto que beta para las otras variables de respuesta no genera efecto, es decir que para esas variables no es relevante el valor de beta utilizado, se podría utilizar un valor de beta de 3 para las otras variables de respuesta. Con respecto a la elección de los valores de q_0 y feromonas, en la **Figura 11 b)** se observa que para un valor de q_0 de $\log(\text{iter}/\text{num_iter})$ (nivel -1) y de alfa de 2 (nivel 1), es más conveniente un valor de feromonas de 0.1, puesto que con esta combinación se disminuye el número de trabajos tardíos. Por otro lado, en c) se muestra que para un valor de q_0 igual a 0.8 (nivel 1) y de alfa 2 (nivel 1), es más conveniente un valor de feromonas de 0.01, debido a que esta combinación disminuye el valor de la variable de respuesta o el número de trabajos tardíos en una proporción mayor que para $q_0 = \log(\text{iter}/\text{num_iter})$, lo cual aparentemente contrasta o difiere un poco esto con los resultados obtenidos para la variable de respuesta tardanza total en donde se establecía que la mejor combinación de feromonas y q_0 eran 0.01 y $\log(\text{iter}/\text{num_iter})$ respectivamente; sin embargo, recordemos que el valor de feromonas de 0.01 mostraba buenos resultados para el makespan y mejores para la tardanza total con respecto a los otros niveles para este factor. Del mismo modo, un valor de q_0 de 0.8 proporciona mejores resultados para el número de trabajos tardíos que $\log(\text{iter}/\text{num_iter})$ para un alfa de 3 y un valor de feromonas de 0.01. Por lo tanto, y siendo consistentes con los resultados previos para las otras variables de respuesta, y debido a que se deben considerar los tres criterios simultáneamente, es evidente que la mejor combinación de valores para los diferentes parámetros son respectivamente: alfa=2, beta=3, rho=0.01, feromonas=0.01 y $q_0=0.8$. A continuación, y del mismo modo que para los dos casos anteriores, se comprobarán los supuestos del diseño experimental realizado con el fin de mostrar la validez del modelo realizado.

Prueba de Normalidad:

H_0 : Los residuos tipificados de la muestra provienen de una población con distribución normal.

H_1 : Los residuos tipificados de la muestra no provienen de una población con distribución normal.

Si significancia (valor p) < 0.05: Rechazo H_0

Si significancia (valor p) > 0.05 No Rechazo H_0

Prueba de Kolmogorov-Smirnov para una muestra

		Residuo estandarizado para TRAB_TARD
N		144
Parámetros normales ^{a,b}	Media	,0000
	Desviación típica	,81935
Diferencias más extremas	Absoluta	,070
	Positiva	,070
	Negativa	-,065
Z de Kolmogorov-Smirnov		,835
Sig. asintót. (bilateral)		,488

a. La distribución de contraste es la Normal.

b. Se han calculado a partir de los datos.

Tabla 31. Prueba de normalidad Kolmogorov-Smirnov para el número de trabajos tardíos. Fuente: El autor.

La prueba arroja un p-value de 0.488, por lo que no se rechaza H_0 . Esto quiere decir que la distribución de probabilidad de los residuales es normal con media cero.

Prueba de Correlación:

H_0 : Los residuales no están correlacionados con el tiempo

H_1 : Los residuales están asociados con el tiempo.

Si significancia (valor p) < 0.05: Rechazo H_0

Si significancia (valor p) > 0.05: No Rechazo H_0

Esta prueba arroja un p-vaue de 0.148 (Figura 24), por lo que no se rechaza H_0 . Esto quiere decir que el orden en que se ejecutaron las corridas no influye en el experimento.

		Residuo estandarizado para TRAB_TARD	ID
Residuo estandarizado para TRAB_TARD	Correlación de Pearson	1	,121
	Sig. (bilateral)		,148
	N	144	144
ID	Correlación de Pearson	,121	1
	Sig. (bilateral)	,148	
	N	144	144

Tabla 32. Prueba de correlación para el número de trabajos tardíos. Fuente: El autor.

Prueba de Levene para igualdad de varianzas:

H_0 : Las varianzas de los grupos son iguales

H_1 : Las varianzas de los grupos son iguales

Si significancia (valor p) < 0.05 Rechazo H_0

Si significancia (valor p) > 0.05: No Rechazo H_0

**Contraste de Levene sobre la igualdad de las
varianzas error^a**

Variable dependiente: TRAB_TARD

F	gl1	gl2	Sig.
1,929	47	96	,003

Contrasta la hipótesis nula de que la varianza error de la variable dependiente es igual a lo largo de todos los grupos.

Tabla 33. Prueba de Levene para igualdad de varianzas. Fuente: El autor.

Esta prueba arroja un p-value < 0.05 por lo que se rechaza la hipótesis nula. Esto quiere decir que la varianza de los grupos son diferentes. Sin embargo, la violación de este supuesto no constituye un hecho grave.

ANEXO 2. GUIA DE USUARIO DE LA HERRAMIENTA FF-MUCH

El algoritmo FF-MUCH fue desarrollado en Visual Basic ® 6.0. Esta herramienta por ser compatible con la plataforma Windows, permite al usuario trabajar de forma casi que intuitiva. Por lo tanto, como parte de los objetivos de este trabajo de grado era la consecución de una herramienta computacional, esta se desarrolló de manera que fuese amigable para el usuario y muy práctica. A continuación se esboza una pequeña guía para el manejo de la herramienta:

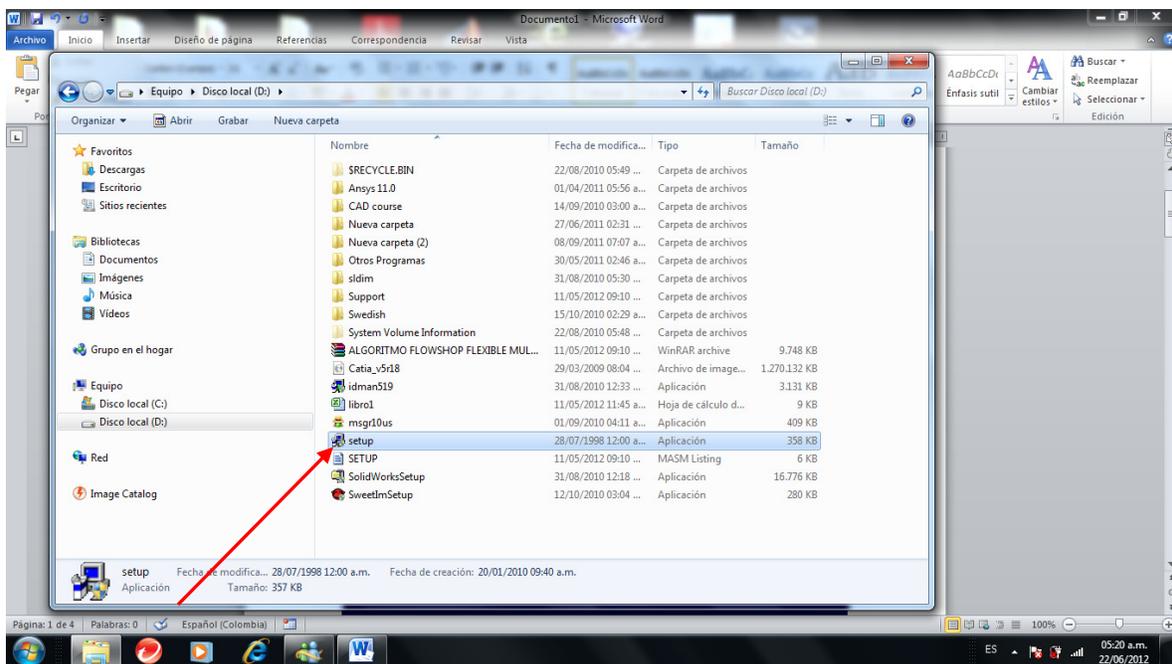


Figura 15. Paso 1 de instalación del programa FF-MUCH. Fuente: El autor.

En primera instancia, se debe instalar la aplicación, para eso hay que ejecutar el instalador dándole doble clic al archivo que se llama Setup.exe que viene en la carpeta de instalación y se despliega el siguiente pantallazo:

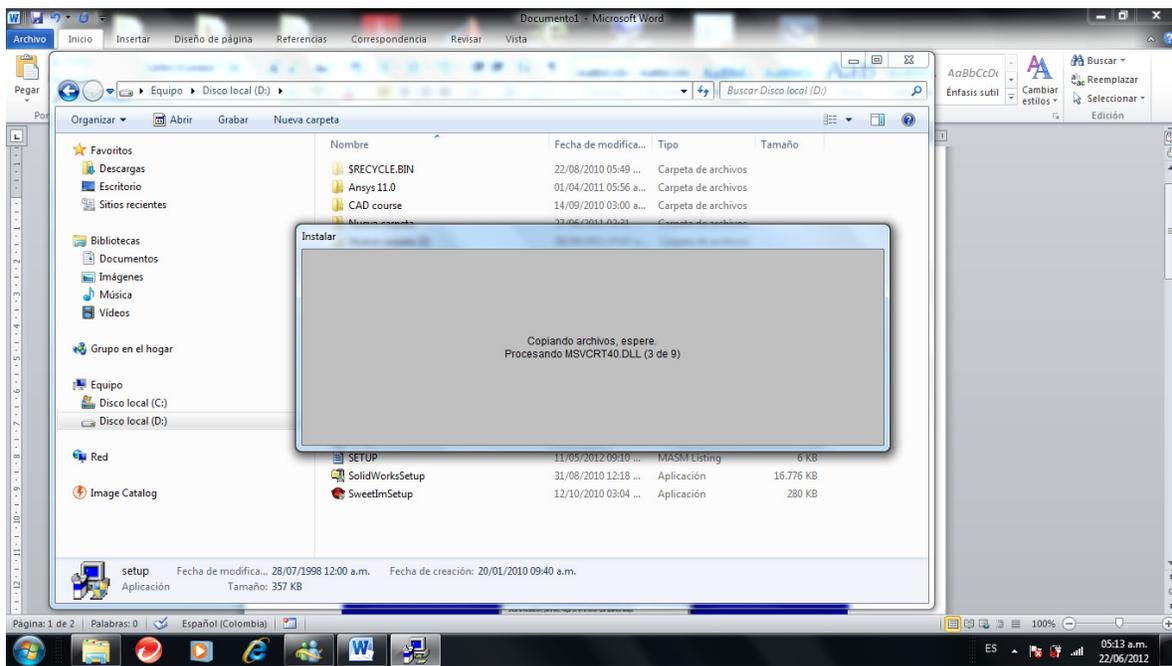


Figura 16. Paso 2 de instalación del programa FF-MUCH. Fuente: El autor.

Hay que esperar un momento a que terminen de cargar los archivos de instalación, antes de que se ejecute el asistente de instalación. Debe salir un pantallazo como este:

Instalación de ALGORITMO FLOWSHOP FLEXIBLE MULTI OBJETIVO ACO

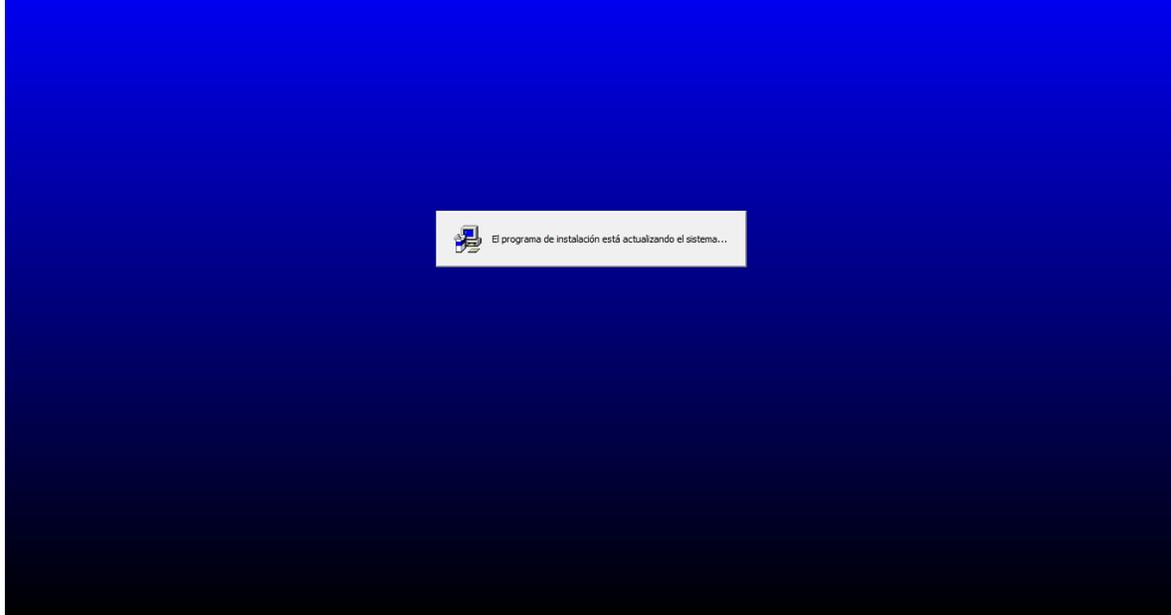


Figura 17. Paso 3 de instalación del programa FF-MUCH. Fuente: El autor.

Hay que esperar a que el asistente indique las instrucciones que se deben de seguir. Posteriormente aparece un pantallazo de la forma:

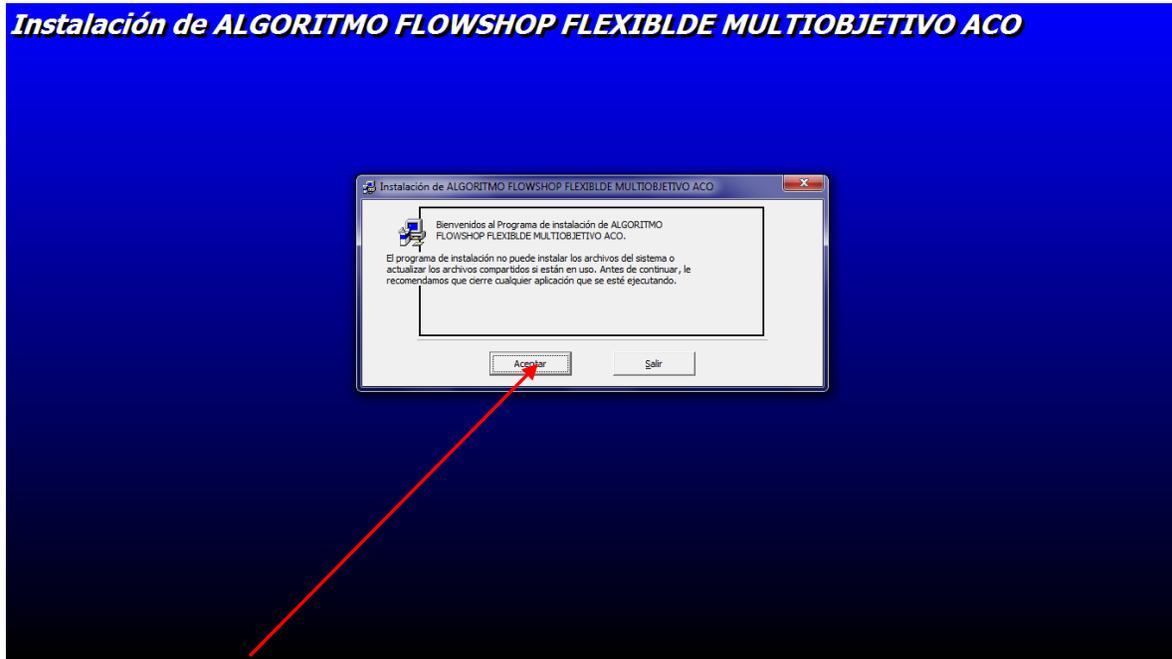


Figura 18. Paso 4 de instalación del programa FF-MUCH. Fuente: El autor.

Este es un mensaje de bienvenida al proceso de instalación. De clic en aceptar:

Posteriormente, le aparece un pantallazo donde le indica cómo dar inicio al programa de instalación y si desea instalar el programa en un directorio o carpeta específica:

Instalación de ALGORITMO FLOWSHOP FLEXIBLE MULTIOBJETIVO ACO

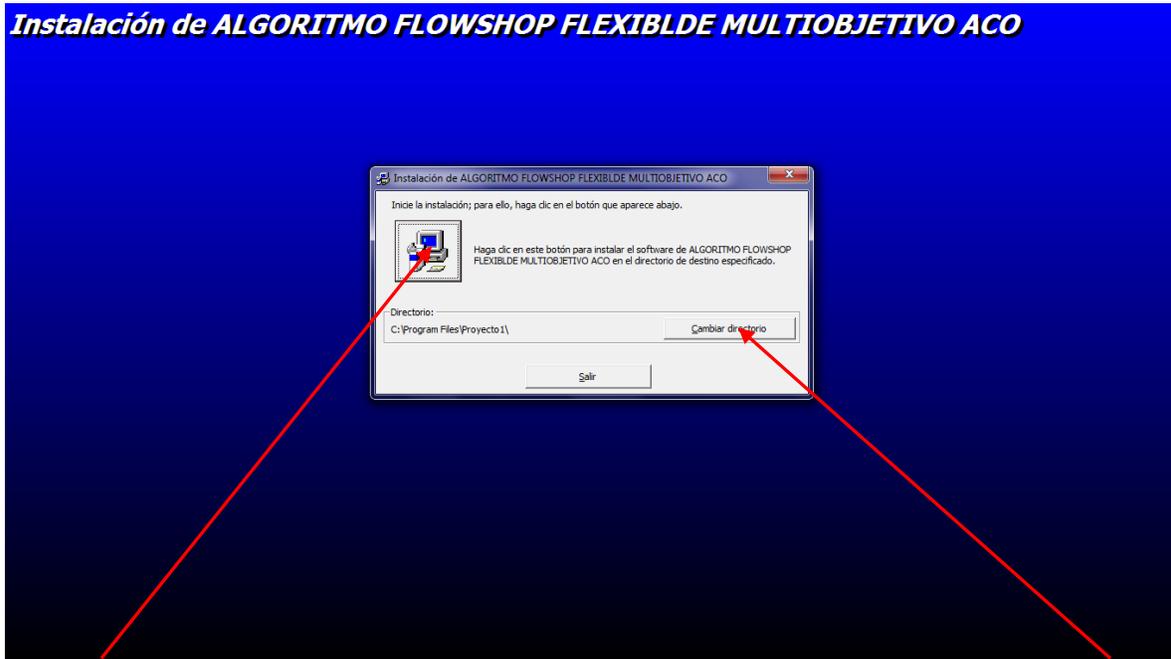


Figura 19. Paso 5 de instalación del programa FF- MUCH. Fuente: El autor.

Se debe dar clic en el botón señalado para poder comenzar con la instalación del programa: Si desea cambiar el directorio donde se guardará la aplicación, de clic cambiar directorio y escoja una ruta específica. Si no lo hace, el programa quedará instalado en la carpeta predeterminada que es C:\Archivos de programas. Después que haya dado clic en el botón para instalar, le aparecerá un mensaje diciendo que si quiere agregar unos componentes específicos:

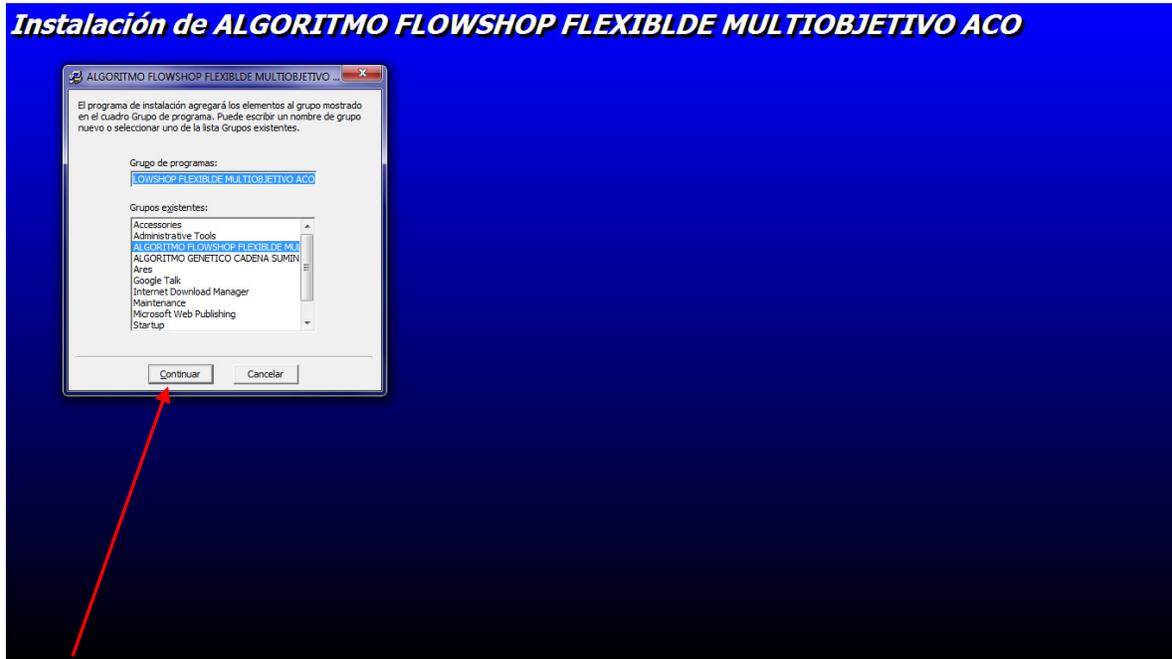


Figura 20. Paso 6 de instalación del programa FF-MUCH. Fuente: El autor.

Aquí simplemente de clic en continuar. Posteriormente le aparecerá un pantallazo con una barra de progreso, diciéndole que iniciará el proceso de instalación. En este paso, no debe realizar ninguna acción hasta que la barra de progreso llegue a 100% y le aparezca el siguiente mensaje:



Figura 21. Paso 7 de instalación del programa FF-MUCH. Fuente: El autor.

Por último, si el proceso se ejecuta correctamente, le debe aparecer un mensaje como este:

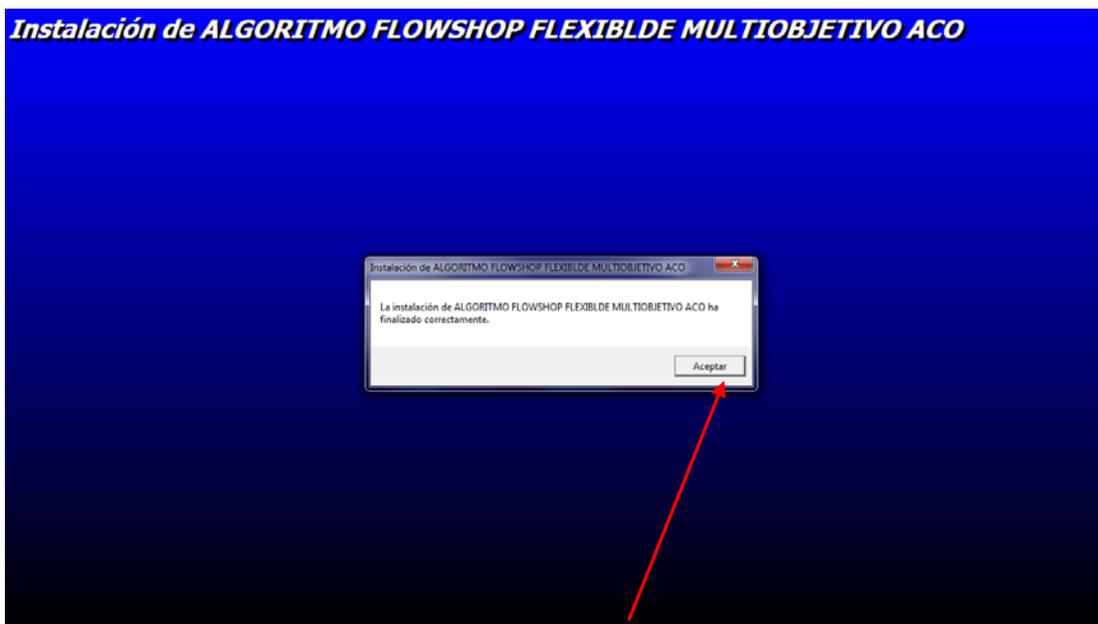


Figura 22. Paso 8 de instalación del programa FF-MUCH. Fuente: El autor.

Esto quiere decir que el programa se instaló sin problemas. Dele clic en aceptar y comience a usar FF-MUCH. Para comenzar a usar el programa, de clic en el menú inicio todos los programas y le aparecerán resaltados aquellos instalados recientemente. Busque la carpeta donde se instaló FF-MUCH, las cuales aparecen en orden alfabético:

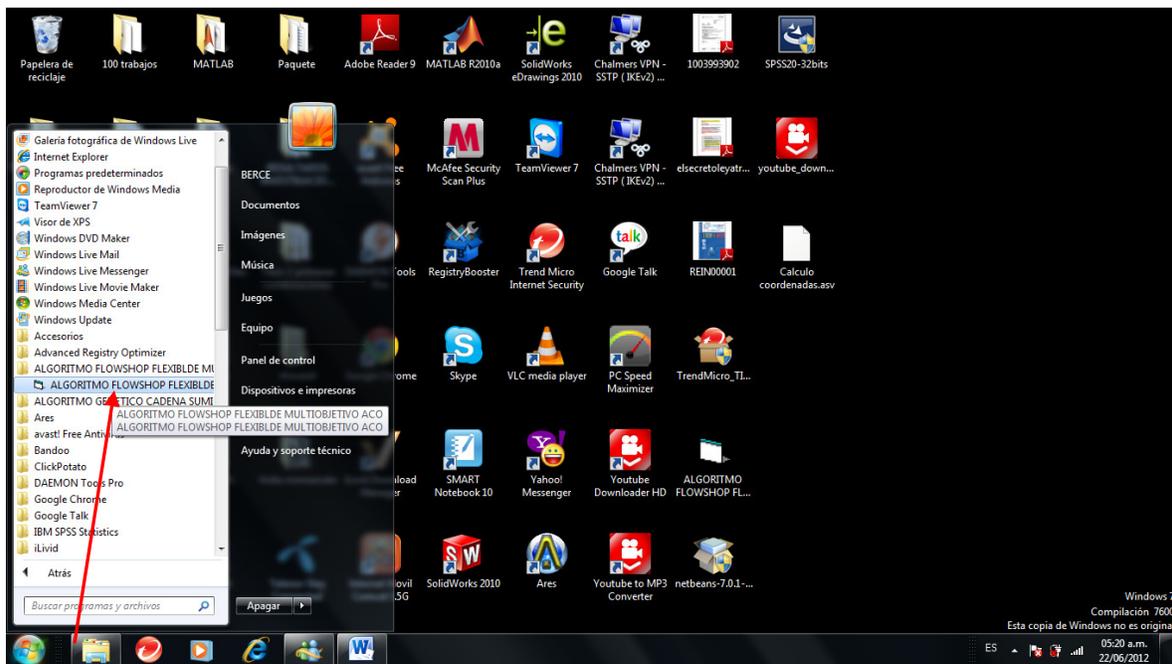


Figura 23. Paso 1 de utilización del programa FF-MUCH. Fuente: El autor.

Cuando haya encontrado programa instalado, el cual tiene un ícono de color verde con blanco (lo cual indica que es un archivo ejecutable) le da clic y el programa comienza a ejecutar. Inicialmente le preguntará si desea cargar los datos desde archivo, es decir si desea abrir un archivo existente o digitar uno nuevo manualmente. Si da clic en Si, entonces se abrirá un cuadro de diálogo preguntándole el nombre y la ubicación del archivo:

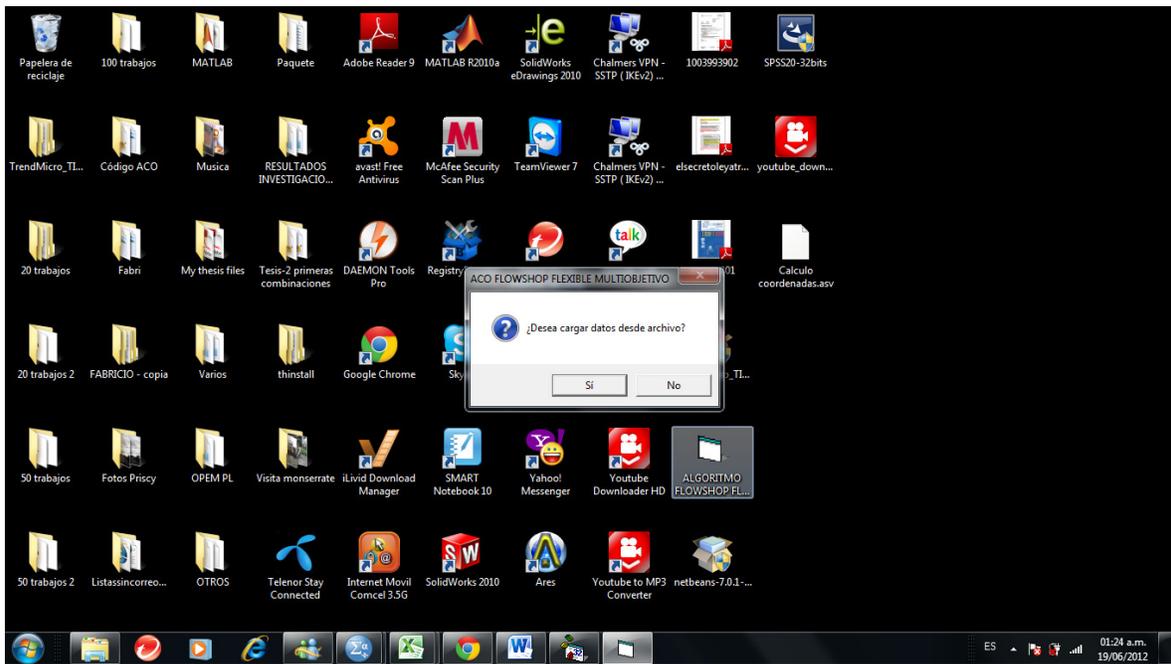


Figura 24. Paso 2 de utilización del programa FF-MUCH. Fuente: El autor.

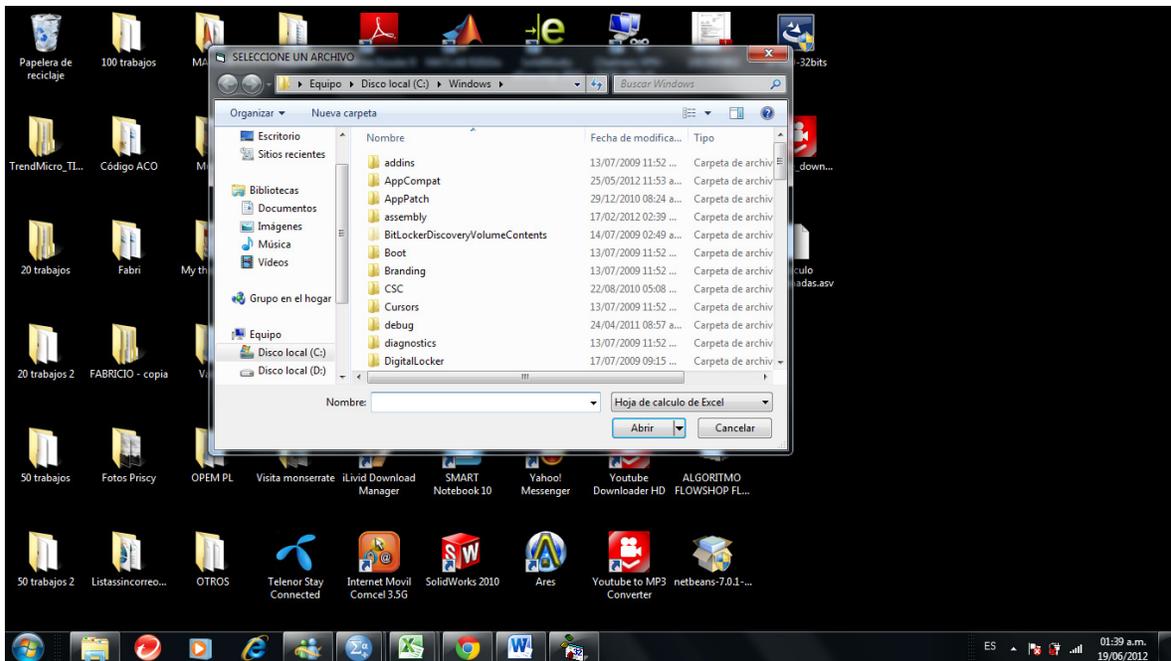


Figura 25. Paso 3 de utilización del programa FF-MUCH. Fuente: El autor.

En caso de que la respuesta a la pregunta sea No, es decir que no se desea abrir un conjunto de datos ya existente, se despliegan dos mensajes que solicitan el número de trabajos y estaciones respectivamente. Si no se le coloca un nombre válido o si no se encuentra el archivo, se desplegará este mensaje hasta que el nombre escrito sea válido. Esto se hace para evitar en lo posible que al programa ingresen datos incorrectos o inconsistentes. Una vez haya digitado el número de trabajos, le aparece otro cuadro para digitar el número de estaciones; se deben escribir el número de trabajos y de estaciones en el espacio habilitado para esto y después le da clic en aceptar; lo mismo se hace para el número de estaciones:

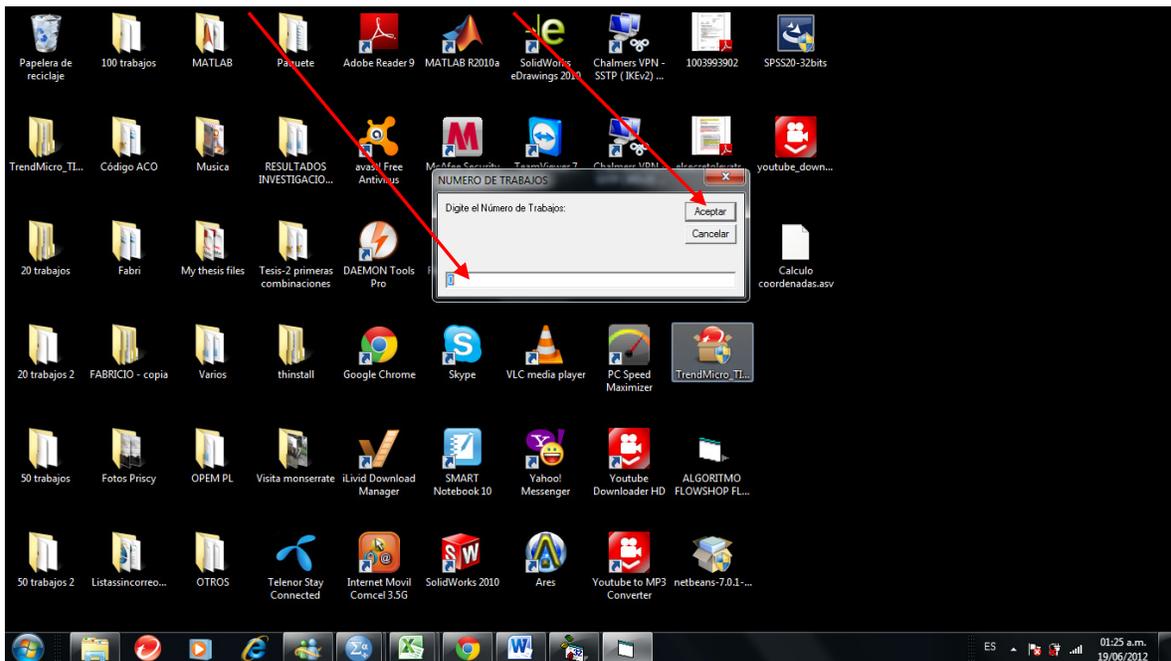


Figura 26. Paso 4 de utilización del programa FF-MUCH. Fuente: El autor.

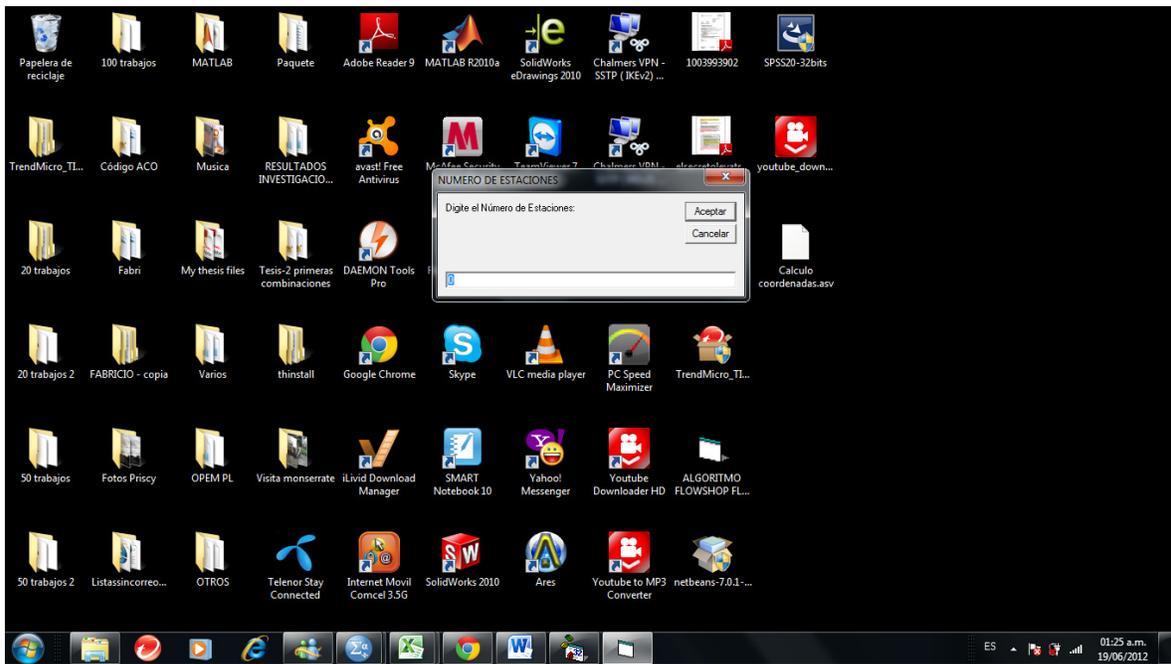


Figura 27. Paso 5 de utilización del programa FF-MUCH. Fuente: El autor.

Posteriormente, se despliega un pantallazo donde solicita el resto de la información o los parámetros del algoritmo colonia de hormigas:

Se debe ingresar un número entre cero y uno.

Se debe digitar un número positivo mayor o igual a 10

Se ingresan las fechas de entrega para cada trabajo

Se ingresa el número de máquinas por estación

Figura 28. Paso 6 de utilización del programa FF-MUCH. Fuente: El autor.

Se ingresa el tiempo de procesamiento de cada trabajo en cada estación

Se debe ingresar un número mayor que cero

Se debe ingresar un número mayor que cero

Se debe ingresar un número entre cero y uno.

Por último, donde dice Ponderación Makepan, Ponderación Tardanza y Ponderación Número de trabajos tardíos, en la mitad del formulario, se colocan las asignaciones dadas a los objetivos en cuestión. Después que se haya digitado toda la información puede realizar la evaluación de la metaheurística para la instancia considera o bien se pueden guardar los datos; Lo último se hace a través del siguiente cuadro de diálogo:

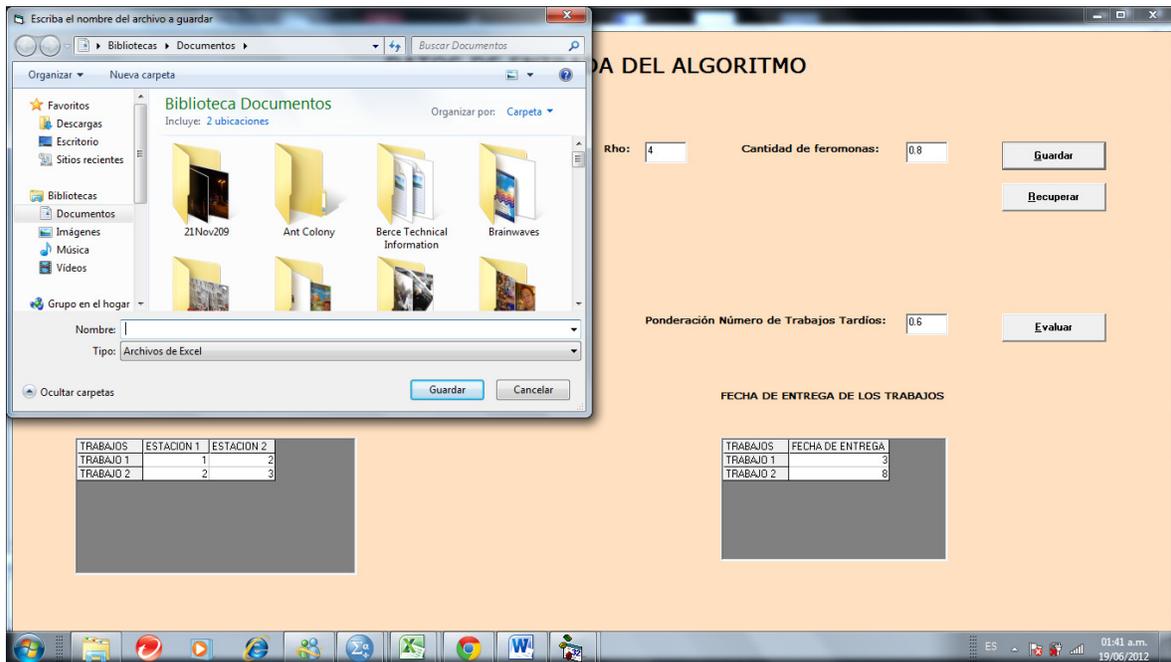


Figura 29. Paso 7 de utilización del programa FF-MUCH. Fuente: El autor.

Visual Basic® lo que hace es crear un archivo con la información que se digitó del problema que se esté resolviendo, para luego almacenarla en una hoja de cálculo. Cuando se da clic en el botón **Recuperar** Visual Basic busca el archivo que generó en Excel y con esa información y se carga el formulario, como se ve en la figura, mientras que el botón guardar almacena la información.

DATOS DE ENTRADA DEL ALGORITMO

Número de hormigas: 10 Alfa: 2 Beta: 3 Rho: 0.01 Cantidad de feromonas: 0.1 **Guardar**

NUMERO DE MAQUINAS POR ESTACION: EST. 1 EST. 2
2 1 **Recuperar**

Ponderación Makespan: 0.3 Ponderación Tardanza: 0.4 Ponderación Número de Trabajos Tardíos: 0.3 **Evaluar**

TIEMPO DE PROCESAMIENTO DE LOS TRABAJOS EN CADA ESTACION

TRABAJOS	ESTACION 1	ESTACION 2
TRABAJO 1	4	5
TRABAJO 2	3	7
TRABAJO 3	2	4
TRABAJO 4	3	6
TRABAJO 5	2	9

FECHA DE ENTREGA DE LOS TRABAJOS

TRABAJOS	FECHA DE ENTREGA
TRABAJO 1	9
TRABAJO 2	15
TRABAJO 3	9
TRABAJO 4	16
TRABAJO 5	11

Figura 30. Paso 8 de utilización del programa FF-MUCH. Fuente: El autor.

Una vez que se haya digitado todos los datos y se le haya dado clic en evaluar, se procede hacer el cálculo del scheduling, cuyo resultado se almacena en el siguiente pantallazo donde se muestra la secuencia de trabajos y máquinas que hace parte de la solución. Hay que aclarar que el programa realiza la programación estación por estación, por lo que los resultados se muestran estación por estación:

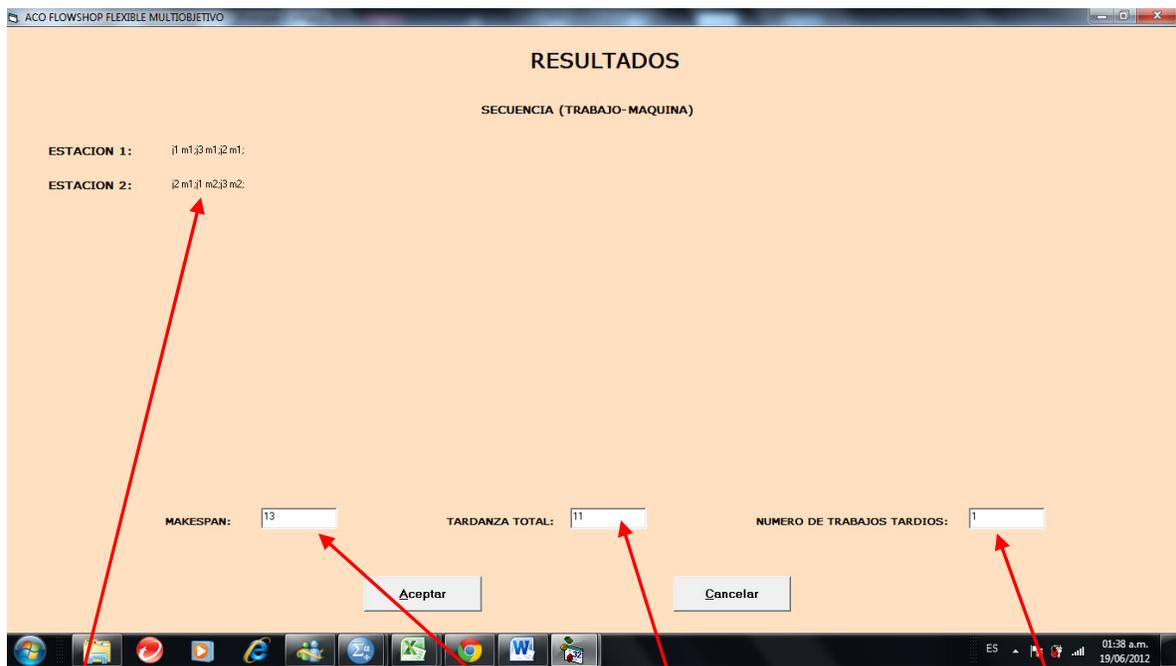


Figura 31. Paso 9 de utilización del programa FF-MUCH. Fuente: El autor.

Solución generada por estación dada por trabajo y máquina

Valores obtenidos del makespan, tardanza total y el número de trabajos tardíos

Si el usuario quiere volver a digitar valores se da clic en aceptar y se responde si a la pregunta de si desea realizar otro scheduling:

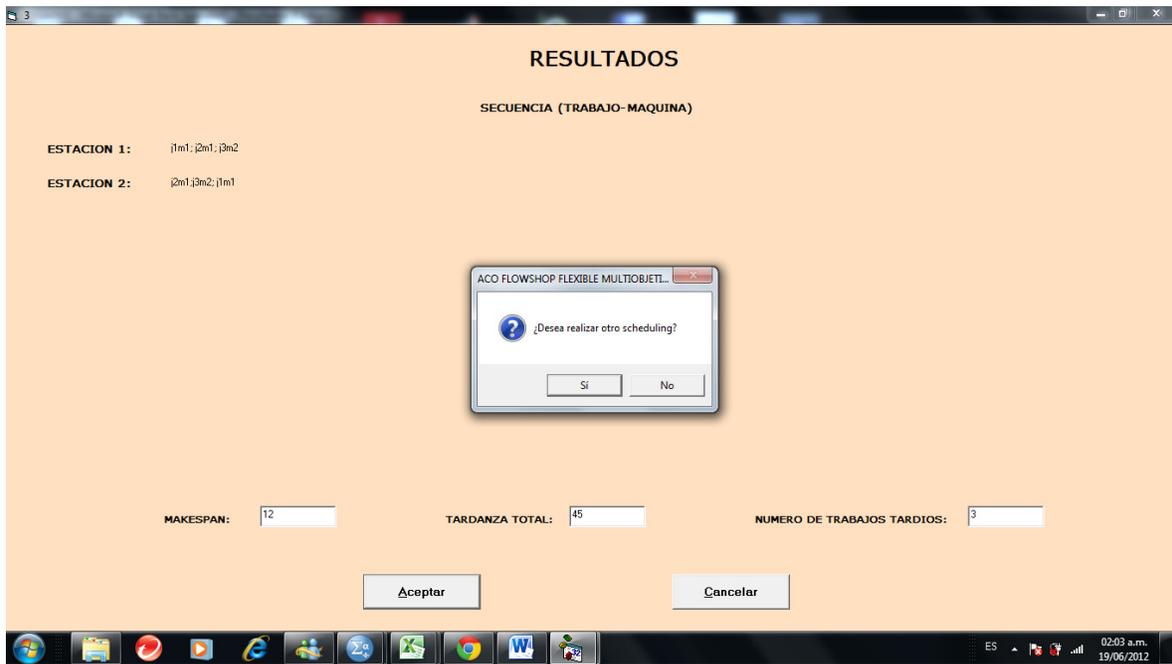


Figura 32. Paso 10 de utilización del programa FF-MUCH. Fuente: El autor.

Si la respuesta es sí se despliega otro cuadro de mensajes a saber, el cual muy similar: cuando se inicia el programa y se realiza nuevamente el mismo procedimiento:

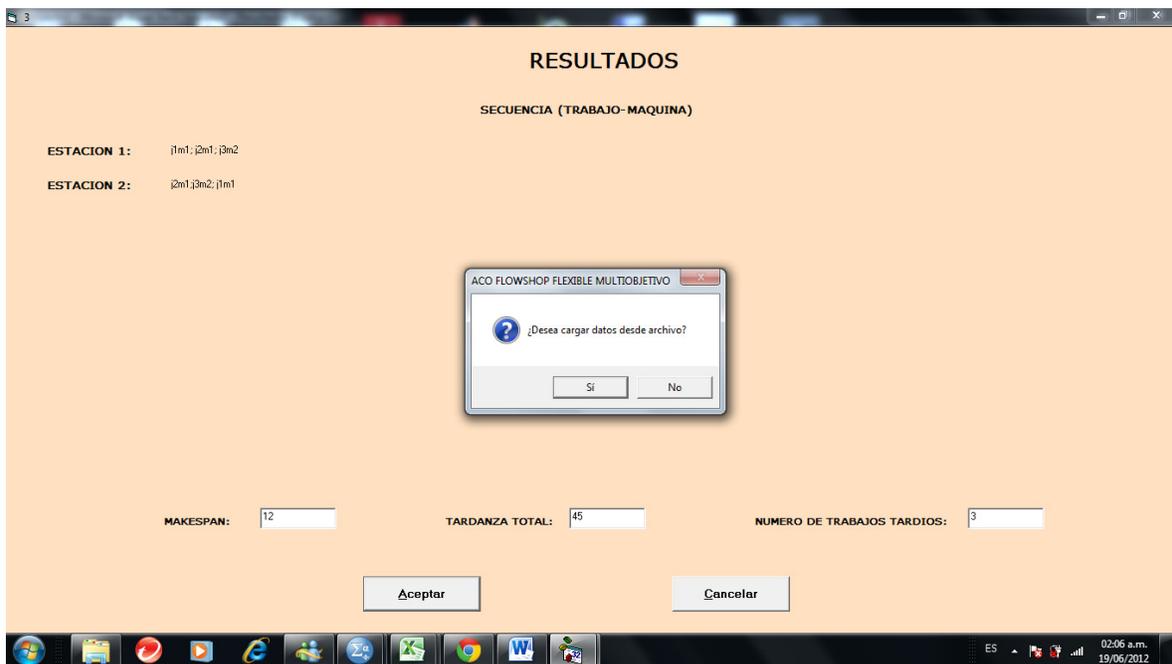


Figura 33. Paso 11 de utilización del programa FF-MUCH. Fuente: El autor.

Luego sale el primer formulario nuevamente para cargar toda la información. En dado caso de que ya no se quiera seguir trabajando, se da clic en el botón cancelar (del formulario resultados) y se despliega el siguiente cuadro de mensaje donde pregunta si desea salir del programa:

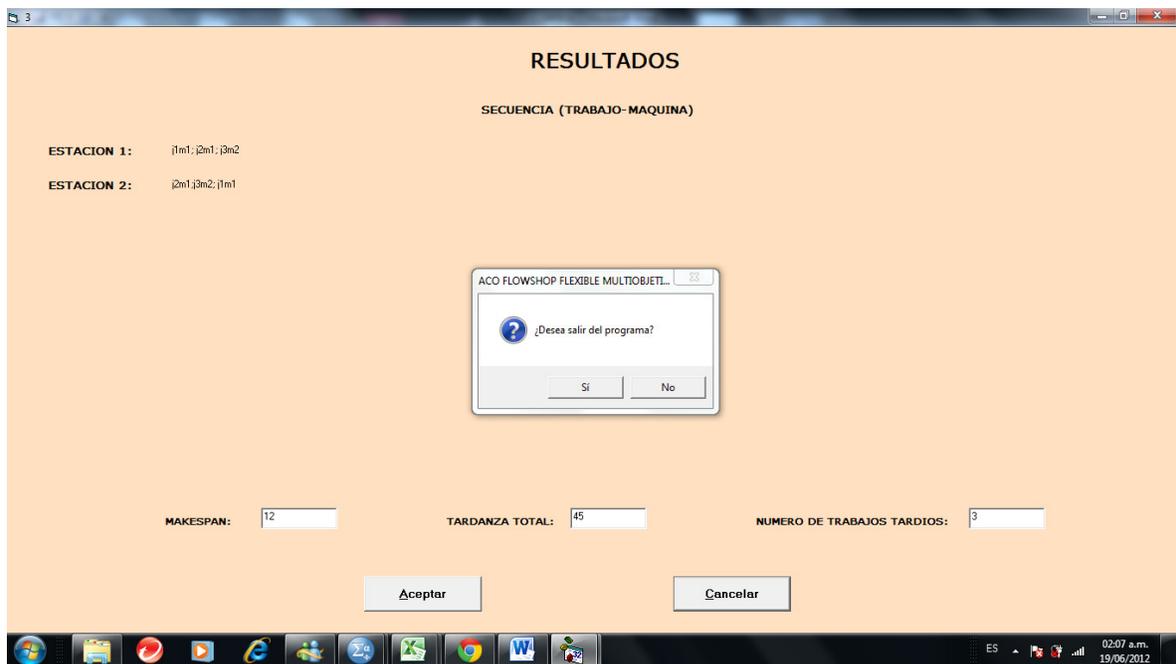


Figura 34. Paso 12 de utilización del programa FF-MUCH. Fuente: El autor.

Si se le da clic en Sí, se cierra automáticamente el programa. Si no, queda en el formulario actual.

UNIVERSIDAD DE LA SABANA
INSTITUTO DE POSTGRADOS- FORUM
RESUMEN ANALÍTICO DE INVESTIGACIÓN (R.A.I)

ORIENTACIONES PARA SU ELABORACIÓN:

El Resumen Analítico de Investigación (RAI) debe ser elaborado en Excel según el siguiente formato registrando la información exigida de acuerdo la descripción de cada variable. Debe ser revisado por el asesor(a) del proyecto. EL RAI se presenta (quemado) en el mismo CD-Room del proyecto.

No.	VARIABLES	DESCRIPCIÓN DE LA VARIABLE
1	NOMBRE DEL POSTGRADO	Maestría en Gerencia de Operaciones
2	TÍTULO DEL PROYECTO	APLICACIÓN DE LA META-HEURÍSTICA COLONIA DE HORMIGAS PARA LA RESOLUCIÓN DE PROBLEMAS MULTI-OBJETIVO DE PROGRAMACIÓN DE LA PRODUCCIÓN EN FLOWSHOPS HÍBRIDOS (FLEXIBLES)
3	AUTOR(es)	FABRICIO ANDRES NIEBLES ATENCIO
4	AÑO Y MES	SEPTIEMBRE DE 2012
5	NOMBRE DEL ASESOR(a)	ELYN LIZETH SOLANO CHARRIS
6	DESCRIPCIÓN O ABSTRACT	This thesis studies the problem of scheduling a set of jobs in a hybrid flowshop. Since this problem is known to be NPhard, a meta-heuristic procedure based on Ant Colony Optimization (ACO) is proposed. The paper considers the multi-objective problem of minimizing the makespan, the total tardiness and the number of tardy jobs. An experimental study is carried out using wellknown instances from the literature. Because this is the first time in literature that these three objectives are considered simultaneously for the hybrid flowshop problem, the proposed procedure is compared with mono-objective heuristics for each optimization criterion. Experimental results show the efficiency and effectiveness of the proposed ACO algorithm.
7	PALABRAS CLAVES	Flowshop flexible, algoritmo de colonia de hormigas, optimización multiobjetivo.
8	SECTOR ECONÓMICO AL QUE PERTENECE EL PROYECTO	MANUFACTURA
9	TIPO DE ESTUDIO	TESIS
10	OBJETIVO GENERAL	Desarrollar una herramienta computacional, basada en la metodología de optimización por colonia de hormigas con el fin de automatizar el proceso de toma de decisiones para la programación de las operaciones, en empresas con configuraciones tipo flowshop flexible y múltiples criterios a optimizar.

11	OBJETIVOS ESPECÍFICOS	<ul style="list-style-type: none"> • Proponer un algoritmo de Colonia de Hormigas, e implementarlo en una herramienta computacional, para la solución del problema de programación de las operaciones en un flowshop flexible, cuyos criterios de optimización a considerar son: el makespan, la tardanza total y el número de trabajos tardíos. • Realizar un estudio experimental a través de un software estadístico, con el fin de parametrizar el algoritmo propuesto. • Comparar los resultados generados por la herramienta desarrollada, con aquellos obtenidos con el software LEKIN®, con el fin de poder validar el funcionamiento de dicha herramienta. • Probar el algoritmo propuesto, utilizando instancias conocidas de la literatura, para evaluar su desempeño en términos de la calidad de las soluciones obtenidas, convergencia y el tiempo de ejecución empleado en su cálculo.
12	RESUMEN GENERAL	<p>Con el fin de mejorar los niveles de competitividad, las empresas de manufactura y de servicio están obligadas a la implementación constante de procedimientos formales que les permitan optimizar sus procesos. En ese sentido, en lo referente a las operaciones de manufactura, la logística de producción, y más específicamente la programación de operaciones, juega un papel importante en cuanto al uso eficiente de los recursos. La programación de operaciones (scheduling, en inglés) es una rama de la optimización combinatoria que consiste en la asignación de recursos para la realización de un conjunto de actividades con el fin de optimizar uno o varios objetivos. Debido a la complejidad intrínseca en la mayoría de los problemas de programación de la producción, los cuales son del tipo NP-duro (esto es, el tiempo que requieren para resolver un caso particular de un problema crece en el peor de los casos de manera exponencial con respecto al tamaño del problema), los métodos exactos convencionales de resolución tales como: programación lineal, entera y mixta, entre otros, no son eficientes en términos del tiempo de cálculo para llegar a la solución óptima. Por lo tanto, se hace necesario el uso de enfoques alternativos para resolver este tipo de problemas en un tiempo razonablemente corto para el tomador de decisiones, sobre todo aquellas que se toman diariamente. Dentro de estos enfoques se encuentran las metaheurísticas, que consisten en procedimientos formales desarrollados con el fin de superar esta dificultad que se presenta con los métodos tradicionales. Los procedimientos meta-heurísticos más comunes para la resolución de problemas combinatorios son: los algoritmos genéticos, la búsqueda tabú, la colonia de hormigas y el recocido simulado entre otros. En ese sentido, el objetivo de esta tesis es desarrollar un sistema de soporte para la toma de decisiones que permita automatizar el proceso de programación de operaciones en configuraciones de producción tipo flowshop flexible, en donde se pretenda optimizar múltiples criterios, a través de una herramienta computacional basada en la metaheurística colonia de hormigas y así responder la siguiente pregunta de investigación: ¿De qué manera el uso de la metaheurística, Colonia de Hormigas, considerando múltiples objetivos afecta la calidad de las soluciones para programar las operaciones en un flowshop flexible? Se estudia un problema multiobjetivo de programación de la producción, en el cual se minimizan el lapso de fabricación o makespan, la tardanza total y el número de trabajos tardíos para una configuración tipo flowshop flexible. Se desarrolla un estudio experimental empleando instancias conocidas en la literatura. Puesto que no se ha encontrado evidencia en la literatura científica de que se haya estudiado este problema multiobjetivo con anterioridad, el procedimiento propuesto es comparado con respecto a soluciones heurísticas para los respectivos problemas mono-objetivo. El estudio experimental realizado muestra que el método propuesto en esta investigación proporciona buenos resultados para instancias con dos y cinco estaciones, aún para un</p>

13 CONCLUSIONES.	<p>En este trabajo de grado se utilizó la metaheurística de colonia de hormigas para resolver un problema multi-objetivo de programación de la producción, cuyos objetivos a optimizar fueron el makespan, la tardanza total y el número de trabajos tardíos, a través de una herramienta computacional denominada FF-MUCH. La optimización por colonia de hormigas ha mostrado ser eficiente para resolver problemas de scheduling con un solo objetivo y por lo tanto se puede extender su utilización para problemas multi-objetivos.</p> <p>Como primera medida, se compararon las soluciones obtenidas por FF-MUCH con el software LEKIN ® para instancias de 20 trabajos utilizando la heurística cuello de botella General SB routine. El algoritmo FF-MUCH presenta muy buenas soluciones frente a LEKIN ® para dos y cinco estaciones en las instancias analizadas, en donde FF-MUCH domina en varios casos fuertemente a los resultados producidos por la herramienta LEKIN ® para los tres criterios simultáneamente, especialmente para dos estaciones. Para el caso de 50 y 100 trabajos, se comparó el algoritmo propuesto con la heurística para calcular la cota inferior propuesta por (Santos et. al, 1995). Para estos casos, el algoritmo FF-MUCH presenta valores muy cercanos al valor de la cota inferior para el makespan para 2 estaciones, pero aumenta la diferencia porcentual conforme aumenta el número de trabajos.</p> <p>Por otro lado, la calidad de las soluciones disminuye conforme aumenta el número de estaciones, aunque no sucede esto si aumenta el número de trabajos. Para el escenario con instancias de cinco estaciones, el algoritmo presenta un desempeño aceptable, donde en la mitad de los casos se obtuvieron muy buenas soluciones con relación a LEKIN ® para 20 trabajos, especialmente para dos de los tres criterios considerados (makespan y el número de trabajos tardíos) y para 50 y 100 trabajos, también aproximadamente en la mitad de las instancias evaluadas, se alcanzaron valores muy cercanos a la cota inferior del makespan. Para ocho estaciones, el desempeño del algoritmo no es bueno, tanto para 20, 50 y 100 trabajos en las instancias consideradas.</p> <p>Lo anterior puede estar sucediendo tal vez porque el algoritmo para instancias moderadamente grandes (en este caso para el número de estaciones), la información heurística compartida que maneja el algoritmo es mayor, lo mismo que el comportamiento aleatorio del mismo, por lo que aumenta el riesgo de que las decisiones tomadas por la regla de transición de estados ocasionen cambios bruscos en dicha información que evita que la metaheurística obtenga soluciones de buena calidad, por lo que sería conveniente evaluar una técnica complementaria que permita mejorar las soluciones encontradas en estos escenarios. Una alternativa puede ser la búsqueda local o la búsqueda tabú, con las cuales algunos estudios muestran que el algoritmo colonia de hormigas tiene su mejor desempeño, especialmente la búsqueda local (Yagmahan y Mutlu-Yenesey, 2010). Aun</p>
-------------------------	---

14	FUENTES BIBLIOGRÁFICAS	<p>Alfieri A. Workload simulation and optimisation in multi-criteria hybrid flowshop scheduling: a case study International Journal of Production Research. Vol. 47, No. 18, 15 September 2009, 5129–5145.</p> <p>Allaoui Hami, Lamouri Samir, Lebbar Maria. A robustness framework for a stochastic hybrid flow shop to minimize the makespan. 2006. IEEE.</p> <p>Allaoui Hami, Artiba Abdelhakim. Scheduling two-stage hybrid flow shop with availability constraints. Computers & Operations Research 33 (2006) 1399–1419.</p> <p>Allaoui Hami, Artiba A. Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. Computers & Industrial Engineering 47 (2004) 431–450. Science Direct.</p> <p>Amin-Naseri Mohammad Reza, Beheshti-Nia Mohammad Ali. Hybrid flow shop scheduling with parallel batching. International Journal of Production Economics 117 (2009) 185–196. Science Direct.</p> <p>Brah, S. A., & Hunsucker, J. L. (1991). Branch and bound algorithm for the flowshop with multiple processors. European Journal of Operation Research, 51, 88–99.</p> <p>Behnamian J., Ghomi Fatemi, M. Zandieh. A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. Expert Systems with Applications. Science Direct.</p> <p>Behnamian J., Ghomi Fatemi, M. Zandieh. Hybrid flowshop scheduling with machine and resource-dependent processing times. Applied Mathematical Modelling. Science Direct 17 pags.</p> <p>Botta-Genoulaz Valérie. Hybrid Flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. International Journal of Production Economics 64 (2000) 101-111. Science Direct.</p>
----	------------------------	---