

# **Diseño de modelo para la programación de producción para una empresa de empaques de caucho**

Sebastián Enrique Ortiz Gaitán

Escuela Colombiana de Ingeniería Julio Garavito  
Decanatura de Ingeniería Industrial  
Maestría en Ingeniería Industrial  
Bogotá D.C., Colombia  
2015

# **Diseño de modelo para la programación de producción para una empresa de empaques de caucho**

Sebastián Enrique Ortiz Gaitán

Trabajo de investigación presentado para optar al título de  
Magíster en Ingeniería Industrial

Director

Ing. Carlos Rodrigo Ruíz Cruz MSc, CPIM

Escuela Colombiana de Ingeniería Julio Garavito  
Decanatura de Ingeniería Industrial  
Maestría en Ingeniería Industrial  
Bogotá D.C., Colombia  
2015

© Únicamente se puede usar el contenido de las publicaciones para propósitos de información. No se debe copiar, enviar, recortar, transmitir o redistribuir este material para propósitos comerciales sin la autorización de la Escuela Colombiana de Ingeniería. Cuando se use el material de la Escuela se debe incluir la siguiente nota “Derechos reservados a Escuela Colombiana de Ingeniería” en cualquier copia en un lugar visible. Y el material no se debe notificar sin el permiso de la Escuela.

## **Agradecimientos**

En primer lugar quisiera expresar mi más profundo agradecimiento a mi asesor Carlos Ruiz, por su ayuda en el proceso de enfocar la idea de realizar este proyecto, así como por su disponibilidad, guía y apoyo durante el desarrollo del mismo.

También quiero agradecer a la empresa Empaques de Caucho Ortiz, en cabeza del gerente Enrique Ortiz, por permitirme realizar este proyecto en base a sus problemáticas, por siempre estar dispuesto a brindarme la información necesaria sobre la empresa y también por inspirar la lluvia de ideas que se realizó como paso inicial del proyecto.

Me gustaría agradecer también a los profesores Sonia Jaimes, por sus oportunos consejos para encaminar el desarrollo de la metaheurística y William Guerrero, como Director de la Maestría de Ingeniería Industrial, por siempre estar pendiente de mis progresos con este trabajo de Grado.

Un agradecimiento especial a la empresa C3 comunicaciones, donde he estado laborando mientras he realizado este trabajo, por brindarme el tiempo necesario para realizar la Maestría y por su paciencia mientras he cumplido con todas las responsabilidades tanto laborales como académicas.

Finalmente quisiera agradecer a mis padres, a mis amigos y a mis compañeros de clase por su apoyo moral y su acompañamiento, por brindarme espacio de esparcimiento y alegría durante la realización del trabajo.

## Resumen

Los sistemas de producción Job Shop (JSP) se caracterizan por que los trabajos a ser procesados tienen secuencias diferentes de producción en un conjunto de máquinas. Una generalización de este tipo de sistemas es el Job shop Flexible (FJSP), en el cual para realizar algunas de las operaciones se cuenta con más de una máquina y tratan dos sub-problemas simultáneamente: asignación de máquinas y secuenciación de operaciones.

La empresa Empaques de Caucho Ortiz es una pequeña empresa en Bogotá dedicada a la producción de empaques de caucho, cuya producción se desarrolla en un entorno de producción bajo pedido, donde las órdenes sólo se generan una vez el cliente las realiza y comúnmente tienen asociado un tiempo de entrega (due date), y que cuenta también con un sistema de producción enfocado hacia el proceso que se puede clasificar como una configuración de Job shop flexible sin recirculaciones.

En este trabajo se propone un modelo matemático cuya función objetivo es minimizar el número de trabajos tardíos y como método de solución se propone una metaheurística de búsqueda tabú cuyo desempeño será evaluado con la ayuda de datos reales de la empresa, de instancias empleadas en la literatura a las cuales se les aplica un método propuesto para la generación de tiempos de entrega y finalmente de herramientas provenientes de la teoría del diseño de experimentos. Finalmente se analiza la competitividad del método de solución propuesto, la robustez de los resultados, los tiempos computacionales y la consistencia de las soluciones en casos reales.

Los resultados muestran que, dada la complejidad de los problemas tratados, el modelo matemático aplicado en el caso de la empresa solamente es funcional en casos donde hay hasta 4 trabajos, logrando obtener un óptimo de cero trabajos tardíos, para problemas más grandes se recurrió al uso de la metaheurística, la cual permitió obtener resultados factibles cercanos al óptimo (entre uno y dos trabajos tardíos) en tiempos computacionales cortos y representan cotas superiores de solución.

## Tabla de contenido

1	Introducción.....	9
1.1	Problemática (Justificación).....	9
1.2	Objetivos y Pregunta de Investigación.....	10
1.3	Alcance y Limitaciones.....	10
1.4	Metodología.....	11
1.5	Descripción.....	12
2	Descripción del proceso de producción.....	13
2.1	La empresa y sus productos.....	13
2.2	Variables y factores de mayor influencia en el proceso.....	14
2.3	Proceso de producción.....	14
3	Descripción del sistema de producción.....	21
3.1	Producción bajo pedido o Make to Order (MTO).....	21
3.2	Sistema Job Shop.....	21
4	Desarrollo del modelo a emplear.....	35
4.1	Modelo matemático.....	35
4.2	Supuestos del modelo.....	36
4.3	Definición de índices y conjuntos.....	37
4.4	Formulación del modelo.....	40
4.5	Aplicación del modelo matemático.....	43
4.6	Métodos de aproximación.....	44
5	Pruebas computacionales aplicados a instancias.....	54
6	Propuesta para la empresa.....	59
6.1	Modelo matemático.....	60
6.2	Método de aproximación.....	62
6.3	Pruebas computacionales y resultados.....	63
7	Conclusiones y Recomendaciones.....	67
	Referencias Bibliográficas.....	69
	Abreviaturas.....	74

## Lista de Figuras

<i>Figura 2.1</i> Visión general de la cadena de abastecimiento de los productos fabricados por la empresa Empaques de caucho Ortiz.....	13
<i>Figura 2.2</i> Diagrama de flujo de la empresa Empaques de caucho Ortiz .....	15
<i>Figura 2.3</i> Diagrama multi-producto de la empresa Empaques de caucho Ortiz.....	17
<i>Figura 2.4</i> Disposición de la planta de la empresa Empaques de caucho Ortiz.....	19
<i>Figura 2.5</i> Diagrama de recorrido en la empresa Empaques de caucho Ortiz .....	20
<i>Figura 4.5.1</i> Resultados del experimento evaluando instancias pequeñas.....	48
<i>Figura 4.5.2</i> Resultados del experimento evaluando instancias medianas.....	48
<i>Figura 4.5.3</i> Resultados del experimento evaluando instancias grandes .....	49
<i>Figura 4.5.4</i> Resultados del experimento evaluando el impacto en la tardanza máxima.....	49
<i>Figura 4.5.5</i> Resultados del experimento evaluando el impacto en la tardanza total .....	50
<i>Figura 4.5.6</i> Resultados del experimento evaluando el impacto en el número de trabajos tardíos .....	50
<i>Figura 5.1</i> Gráfica Instancia vs. Tiempo de procesamiento.....	57
<i>Figura 5.2</i> Gráfica Número de trabajos vs. Tiempo de procesamiento.....	57
<i>Figura 5.3</i> Diagrama de Gantt para el ejemplo con cero trabajos tardíos (Instancia Fattahi 7)...	58
6 Propuesta para la empresa .....	59
<i>Figura 6.1</i> Diagrama de Gantt ejemplo usado en el modelo matemático .....	64
<i>Figura 6.2</i> Diagrama de Gantt para el ejemplo 1 con datos de la empresa .....	66
<i>Figura 6.3</i> Diagrama de Gantt para el ejemplo 2 con datos de la empresa (B.T) .....	66

## Lista de tablas

<i>Tabla 3.1</i> Referencias bibliográficas para FJSP minimizando makespan. ....	23
<i>Tabla 3.2</i> Referencias bibliográficas para FJSP minimizando variables de nivel de servicio. ....	32
<i>Tabla 4.1</i> Instancias de Fattahi usadas para probar el modelo matemático.....	44
<i>Tabla 5.1</i> Instancias de Fattahi usadas para probar la metaheurística .....	54
<i>Tabla 5.2</i> Instancias de Hurink (s_data) usadas para probar la metaheurística.....	54
<i>Tabla 5.3</i> Resultados obtenidos usando las instancias de Fattahi.....	55
<i>Tabla 5.4</i> Resultados obtenidos usando las instancias de Hurink (s_data).....	55
<i>Tabla 5.5</i> Desempeño de las instancias de Fattahi .....	56
<i>Tabla 5.6</i> Desempeño de las instancias de Hurink (s_data) .....	56
<i>Tabla 5.7</i> Ejemplo de secuenciación con cero trabajos tardíos (Instancia Fattahi 7).....	58
<i>Tabla 6.1</i> Ejemplo de datos de la empresa estudiada .....	59
<i>Tabla 6.2</i> Ejemplo 1 de aplicación a datos de la empresa estudiada .....	63
<i>Tabla 6.3</i> Ejemplo 2 de aplicación a datos de la empresa estudiada .....	64
<i>Tabla 6.4</i> Comparación de resultados entre el modelo exacto y la metaheurística .....	64
<i>Tabla 6.5</i> Secuenciación con datos de la empresa (ejemplo 1) .....	65
<i>Tabla 6.6</i> Secuenciación con datos de la empresa (ejemplo 2) .....	65



## 1 Introducción

### 1.1 Problemática (Justificación)

La empresa Empaques de Caucho Ortiz es una pequeña empresa dedicada a la producción de empaques de caucho, cuya producción se desarrolla en un entorno de producción bajo pedido (Make to Order, MTO) con un sistema de producción enfocado hacia el proceso que se puede clasificar como una configuración de Job shop flexible.

Las empresas que tienen definido un programa de producción están en capacidad de tomar decisiones respecto a la secuenciación de sus trabajos basados en información real y objetiva y no en aspectos meramente empíricos; lo cual puede llegar a ofrecer como resultado el logro de objetivos propios de la gestión de operaciones como son el cumplimiento con las fechas de entrega de los pedidos, reducción de costos, aumentando los niveles de servicio y la satisfacción del cliente (Osorio y otros, 2008).

Sin embargo, en casos como el de la empresa objeto de estudio, caracterizado por la falta de procedimientos para definir la programación de la producción, existe una necesidad de encontrar alternativas que permitan mejorar su operación y de esta forma su productividad y competitividad. Factores como la limitada infraestructura y espacio para mantener el producto en proceso y terminado, objetivo de inventarios cero, plazos de entrega de los pedidos cortos e imprevistos en los procesos de producción generan restricciones adicionales en la situación presentada.

Se ha comprobado un problema de cumplimiento con la entrega de los pedidos, dado que, de acuerdo con un corta descripción de la situación actual realizada por el gerente de la empresa estudiada y un muestreo realizado previamente, en el último año el porcentaje de entregas atrasadas ha subido en un poco más del 5% donde, empíricamente, en casos donde los pedidos son más grandes generalmente se les busca una salida pronta del sistema de producción, provocando retrasos en otros pedidos pequeños que están en espera (hay semanas donde, dependiendo de la cantidad de trabajos recibidos y siendo casos muy delicados, pueden llegar a tener tres trabajos tardíos), lo cual se ha reflejado en la terminación de negociaciones con tres de sus clientes (aproximadamente el 40% del mercado que maneja la empresa) y en consecuencia, se ha venido presentando una situación en la cual los ingresos apenas logran cubrir los gastos y los costos de operación de la empresa.

Por lo anteriormente dicho, se justifica diseñar un modelo para la programación de la producción que contribuya al mejor uso de los recursos de la empresa, buscando adicionalmente el cumplimiento de todos los requerimientos y pedidos de los clientes, en cantidad, calidad y oportunidad, con el propósito de aumentar el nivel de satisfacción y confianza de los clientes con la empresa.

## 1.2 Objetivos y Pregunta de Investigación

**Pregunta:** ¿A través de qué modelo se puede soportar la toma de decisiones de la empresa objeto de estudio en relación a la programación de la producción, la cual se encuentra en un entorno de producción bajo pedido y una configuración Job shop flexible?

**Objetivo General:** Diseñar un modelo de programación de la producción para la empresa Empaques de Caucho Ortiz considerando sus características particulares de encontrarse en un entorno de producción bajo pedido y tener una configuración de Job shop flexible y que permita mejorar los indicadores asociados a los niveles de servicio.

### Objetivos Específicos:

- Revisar la literatura relacionada con modelos de programación de producción para sistemas de producción similares al estudiado.
- Analizar detalladamente el sistema de producción así como las variables y factores que afectan el procedimiento de programación de la producción.
- Representar matemáticamente el modelo para la programación de la producción propuesto, considerando las particularidades del sistema y entorno.
- Definir un método de solución del modelo propuesto.
- Evaluar el desempeño del modelo propuesto en relación con las métricas asociadas a la no entrega puntual de los pedidos.

## 1.3 Alcance y Limitaciones

Este proyecto se limita al diseño del modelo de programación de la producción, será decisión de la empresa aplicar el modelo propuesto en su sistema de producción. Se asume que no existen operaciones de mantenimiento ni reprocesos (los trabajos recorren cada estación una vez y no regresan en casos donde se requeriría volver a alguna estación).

## 1.4 Metodología

Para el desarrollo de este proyecto se empleara la siguiente metodología:

1. Para la revisión de la literatura relacionada al modelo a estudiar se realizará una búsqueda en bases de datos especializadas, de forma que los artículos que sean revisados sirvan como un punto de referencia al modelo que se va a diseñar.
2. Identificar y describir tanto el sistema de producción empleado por la empresa como las variables y factores que más influyen en el desarrollo de la producción, para lo cual se realizara inicialmente un diagrama de flujo del proceso de la empresa, determinando actividades esenciales dentro del sistema de producción y sus respectivas secuencias de actividades.
3. Identificar y describir las variables y factores que más influyen en el desarrollo de la producción, de manera que se puedan determinar causas de fallas en el sistema de producción y estar en capacidad de reducir sus efectos a corto y largo plazo.
4. Formular un modelo matemático que se ajuste al sistema de producción previamente establecido y considerando las variables de mayor impacto para la producción de la empresa, teniendo como referencia aquellos modelos encontrados en la literatura que tengan una mejor aproximación al sistema de producción estudiado.
5. Plantear y poner a prueba alternativas para la solución del modelo matemático, para lo cual se plantearan algoritmos para la secuenciación y asignación de máquinas y actividades del sistema de producción a través de métodos de aproximación tales como heurísticas, empleando macros en Excel o herramientas como Legin para estimar el posible desempeño del modelo.
6. Realizar un análisis de desempeño de los métodos de solución planteados, basados en los resultados arrojados por las pruebas de las heurísticas, enfatizando en las posibles mejoras de variables operativas (relacionados principalmente con las variables asociadas con el cumplimiento de los pedidos) a través del análisis de los datos.
7. Proponer un procedimiento para la programación de la producción de la empresa con base en los resultados obtenidos por el modelo propuesto para posteriormente plantear recomendaciones a la empresa y realizar las respectivas conclusiones del proyecto.

## **1.5 Descripción**

En este trabajo se presenta el modelo de programación de producción desarrollado y los resultados computacionales que dicho modelo genera. Las secciones 2 y 3 ofrecen una descripción general del sistema y del proceso a ser estudiado, así como de las variables que más impacto generan en el desempeño del sistema. En la sección 4 se realiza la formulación y el desarrollo del modelo matemático y del método de solución asociado. La sección 5 describe los resultados computacionales de la ejecución de los métodos de solución y realiza un análisis del desempeño de cada uno. Finalmente en la sección 6, con base en los análisis realizados, se propone un procedimiento para la programación de la producción de la empresa Empaques de Caucho Ortiz y se realizarán igualmente recomendaciones y se presentarán conclusiones acerca de este trabajo.

## 2 Descripción del proceso de producción

### 2.1 La empresa y sus productos

Empaques de Caucho “Ortiz” es una empresa pequeña dedicada a la fabricación de empaques de caucho para uso industrial y automotriz la cual ha estado operando desde hace más de 30 años. Dentro de la cadena de abastecimiento se puede situar a esta empresa como un proveedor de partes secundarias para sus clientes (véase figura 2.1), las cuales serán descritas posteriormente en la descripción de los productos.

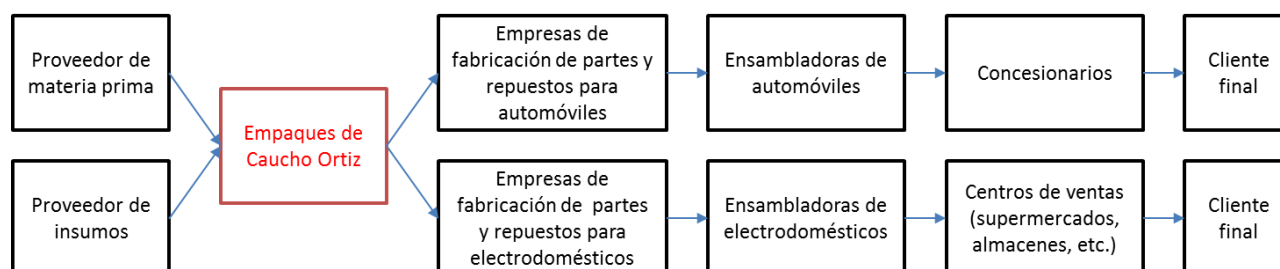


Figura 2.1 Visión general de la cadena de abastecimiento de los productos fabricados por la empresa Empaques de caucho Ortiz.

Fuente: Elaboración propia

Inicialmente contamos con que la empresa cuenta con al menos 60 referencias de productos diferentes, los cuales se pueden clasificar en:

- Tapones (representa aproximadamente el 40% de las ventas)
- Gromets (representa aproximadamente el 25% de las ventas)
- Mangueras (representa aproximadamente el 15% de las ventas)
- Fundas (representa aproximadamente el 5% de las ventas)
- Soportes (representa aproximadamente el 3% de las ventas)
- Empaques cónicos (representa aproximadamente el 3% de las ventas)
- Bridas (representa aproximadamente el 3% de las ventas)
- Ruedas (representa aproximadamente el 3% de las ventas)
- Empaques para fumigadoras (representa aproximadamente el 3% de las ventas)
- Empaques para boleras (actualmente no se producen pero se siguen ofreciendo)

## **2.2 Variables y factores de mayor influencia en el proceso**

Uno de los factores considerados por la empresa como de vital importancia es el cumplimiento con la entrega de trabajos, sin embargo, este cumplimiento trata principalmente de entregar con el pedido, por lo cual el plazo de entrega es crucial para medir el desempeño de la empresa. Este tipo de empresas maneja sus negocios de tal forma que la entrega de los pedidos esta especialmente ligada, aparte de los plazos de entrega, tanto a la disponibilidad de materia prima y de insumos como al cumplimiento de los pactos por parte del cliente y es en ambos aspectos donde se pueden presentar retrasos con la entrega de los productos; por lo tanto, se afirma que para esta empresa se deben asumir mayores consecuencias por la no entrega de un pedido que por la entrega retrasada del mismo, sin querer implicar que por entregas retrasadas no se deba pagar un precio, por lo cual se asume que para el modelo realizado y la heurística aplicada se debe tener un enfoque principalmente en minimizar la cantidad de trabajos tardíos, siendo afectado implícitamente dentro del modelo el valor de las tardanzas de cada trabajo. También hay que destacar que actualmente se aplica la regla FIFO (first in, first out) para realizar la secuencia de trabajos.

## **2.3 Proceso de producción**

En el caso de esta empresa, el proceso consiste de las siguientes actividades reflejado en el siguiente diagrama de flujo (figura 2.2):

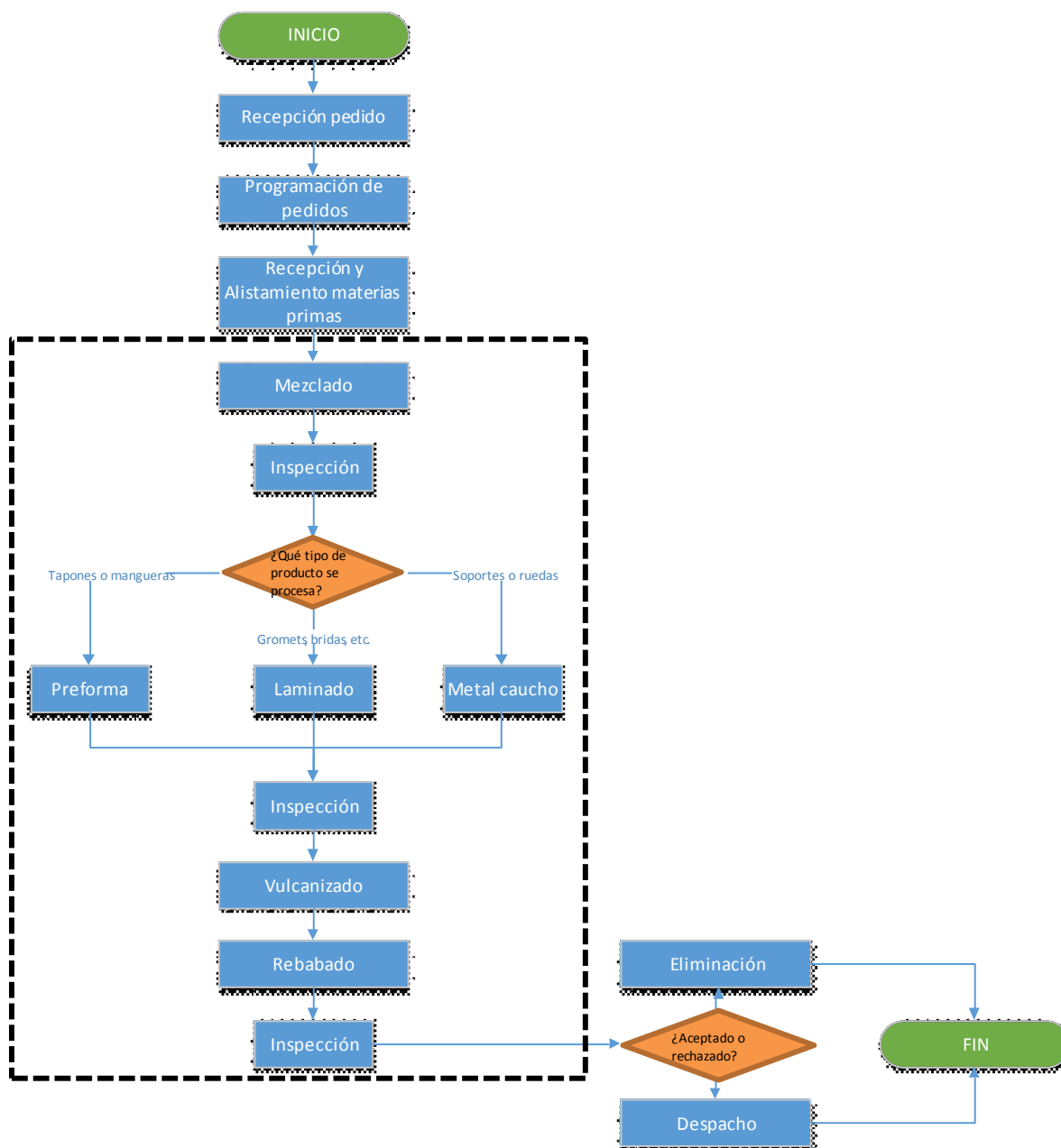


Figura 2.2 Diagrama de flujo de la empresa Empaques de caucho Ortiz  
Fuente: Elaboración propia

A continuación se presenta una descripción más detallada de cada operación del proceso

1. Recepción de pedido: La empresa recibe un pedido por parte de los clientes, aclarando que cada cliente tiene sus propios productos, es decir, no existen dos clientes que piden una misma referencia de producto y no todos los clientes piden de todos los tipos de productos.

2. Alistamiento de Materias primas e insumos: Se realiza un alistamiento considerando que para cada producto se emplea una “receta” diferente, la cual depende de las características que debe tener el producto final en cuanto a dureza, elasticidad, resistencia a la temperatura, etc.

3. Mezclado: Para este proceso se emplea un molino, el cual se encarga de mezclar el caucho (ya sea natural o sintético o de ambos) con aditivos que permiten mejorar las características del caucho resultante, el cual permite que el producto final se ajuste a los requisitos del cliente.

En este punto se debe realizar un proceso diferente, dependiendo del tipo de pieza que se está procesando:

4. Preforma: Este proceso se realiza especialmente para tapones y mangueras y consiste en tomar el caucho ya mezclado y, mediante el uso de un extruder, se procesa una cuerda de caucho, la cual es cortada antes de pasar al molde y ser vulcanizada.

5. Laminado: Este proceso, como su nombre lo indica, consiste en tomar el caucho mezclado y pasarlo por una laminadora, la cual procesa láminas de diferentes grosores y se emplea en la mayoría de los productos que fabrica la empresa (gromets, fundas, bridas, entre otros).

6. Metal-caucho: Este proceso se realiza principalmente con las ruedas y los soportes y se le denomina así debido a que se emplea una barra de metal para el procesamiento del caucho. El metal inicialmente se desengrasa y una vez limpio se le aplica pegante y pintura para luego empezar a colocar el caucho de tal forma que vaya cubriendo la barra en forma de espiral. Una vez hecho esto se deja secar durante 24 horas y después de ese tiempo se procede al vulcanizado de la pieza.

Los pasos finales son igualmente comunes a todas las piezas:



7. **Vulcanizado:** En este proceso se incluye el moldeado de la pieza, para cada referencia hay un molde diferente; posteriormente se ubica el molde en la maquina vulcanizadora previamente calentada y se deja la pieza durante cierto tiempo, el cual depende básicamente del tipo de caucho empleado y de ciertas características que se requiere que tenga la pieza (básicamente puede durar entre 2 y 10 minutos). Una vez pasado ese tiempo se deja enfriar el molde para posteriormente retirar la pieza.

8. **Rebado:** Consiste en remover los residuos resultantes después del vulcanizado.

9. **Inspección:** Durante todo el proceso de producción se realizan constantes inspecciones para verificar que el caucho, desde su estado inicial, cumpla con las características que se requieren, y al salir de cada uno de los procesos se va revisando el estado del material. Durante el rebado se va realizando una última inspección para determinar cuáles piezas están en condiciones de ser entregadas al cliente.

10. **Despacho:** Las piezas son empacadas y entregadas al cliente.

Para el desarrollo del modelo de programación de producción, solo se tendrán en cuenta los procesos de mezcla, preforma, laminación, metal-caucho, vulcanización y rebado y para cada producto, el proceso a realizar se refleja en la siguiente grafica (Véase Figura 2.3):

Proceso	Tapones	Gromets	Mangueras	Soportes	Bridas	Empaques cónicos	Fundas	Ruedas aspiradoras	Empaques para fumigadoras
Mezcla	1	1	1	1	1	1	1	1	1
Preforma	2		2						
Laminación		2			2	2	2		
Proceso metal-caucho				2				2	2
Vulcanización	3	3	3	3	3	3	3	3	3
Rebado	4	4	4	4	4	4	4	4	4

Figura 2.3 Diagrama multi-producto de la empresa Empaques de caucho Ortiz  
Fuente: Elaboración propia

Cada vez que una orden es aceptada, en la empresa se procede a realizar el alistamiento de las herramientas y máquinas para su procesamiento y de acuerdo con la disposición en planta de cada operación (Véase Figura 2.4) se realiza el siguiente proceso: se procede a llevar la materia prima y los insumos a la zona de mezclado, luego, dependiendo del producto pasa ya sea a la zona de laminado, de preforma o de metal-caucho, una vez se termina el proceso correspondiente, se pasa a la zona de vulcanización y finalmente se llega a la zona de rebabado, donde se recolectan los productos terminados y son llevados a la oficina, donde se realiza el despacho del pedido.

Dado que es una pequeña empresa y los recursos para realizar sus pedidos son limitados, la cantidad de pedidos que recibe a la semana tiende a ser baja, generalmente se reciben entre 3 y 7 pedidos a la semana y en cada pedido generalmente se producen entre 1 y 4 tipos de productos.

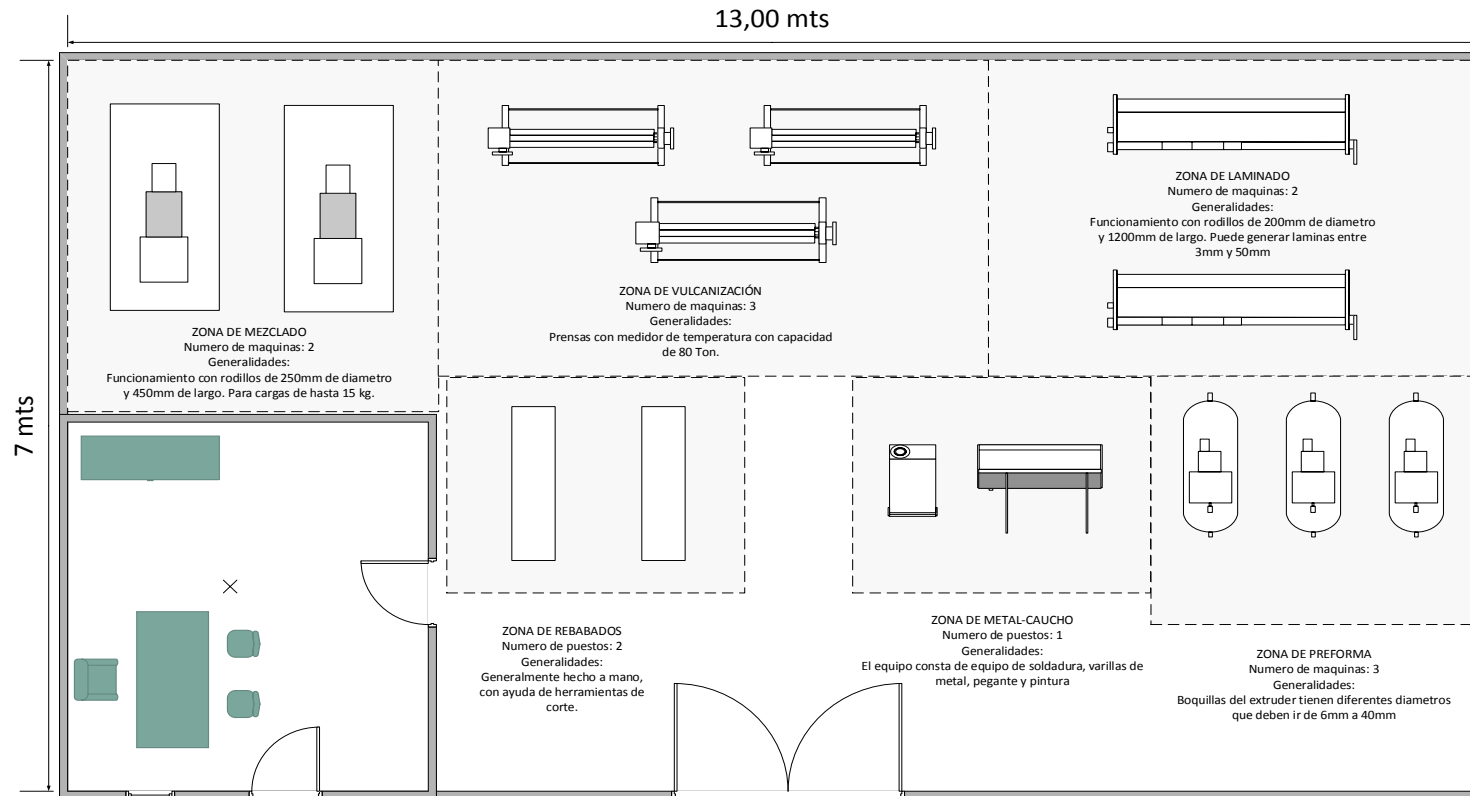


Figura 2.4 Disposición de la planta de la empresa Empaques de caucho Ortiz  
Fuente: Elaboración propia

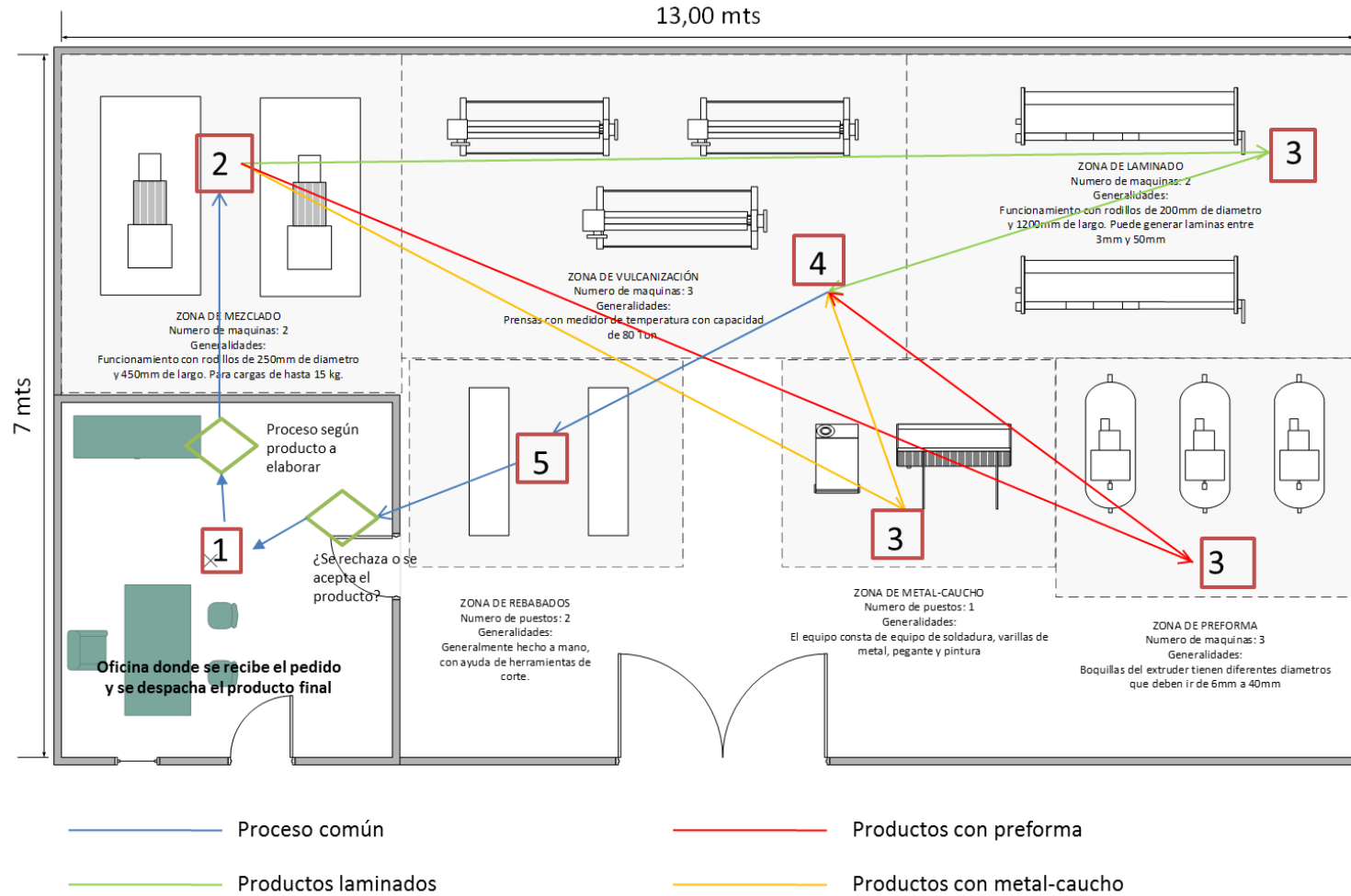


Figura 2.5 Diagrama de recorrido en la empresa Empaques de caucho Ortiz  
 Fuente: Elaboración propia

### **3 Descripción del sistema de producción**

#### **3.1 Producción bajo pedido o Make to Order (MTO)**

Esta es una estrategia de producción en el cual la fabricación de los productos inicia en el momento en que llega la demanda externa, haciendo que las empresas que lo implementan manejen cantidades mínimas o nulas de inventario de producto terminado, lo cual busca garantizar que haya bajos costos de inventarios de producto terminado. Uno de los principales retos de este sistema es el de reducir u optimizar el tiempo de proceso del pedido ya que si bien hay mayores facilidades para el flujo de la información, el flujo de los productos hacia los clientes se convierte en la principal preocupación de las empresas y se hace importante que se cumplan con sus necesidades (Zoin, 2010).

Si bien implementar este sistema es un poco complejo precisamente por las grandes transformaciones que se deben aplicar en las empresas y sus implicaciones a nivel estructural y financiero, puede llegar a ser una gran ventaja competitiva si el sistema de producción es capaz de abastecer rápidamente a los clientes con los pedidos que realizan y adicionalmente ofrecer productos personalizados para cada cliente. Este tipo de estrategias de producción han sido implementadas principalmente por empresas que ofrecen productos de gama alta, productos especializados o personalizados, productos con bajo volumen de pedidos y su uso se hace adecuado también cuando los productos no tienen un diseño modular y la tasa de uso de sus componentes es irregular dada la naturaleza impredecible de la demanda (Zoin, 2010).

#### **3.2 Sistema Job Shop**

Es un problema de optimización en el cual se tratan de asignar recursos a diferentes trabajos en momentos particulares del proceso, cuyo fin es el de minimizar el tiempo total del proceso (makespan). Este problema tiene una serie de elementos que se deben tener en cuenta al momento de trabajar (Parveen, 2010):

- Un número finito de trabajos, los cuales consisten de una cadena de operaciones.

- Un número finito de máquinas, las cuales manejan por mucho una operación al tiempo, siendo cada operación procesada durante un periodo de tiempo sin ningún tipo de interrupción.

Si bien estos son los elementos básicos, se pueden presentar variaciones en la forma de abarcar el problema, por ejemplo, las máquinas pueden estar relacionadas entre sí, o ser independientes, pueden existir tiempos muertos entre trabajos, pueden existir restricciones en cuanto a los trabajos que se pueden hacer en ciertas máquinas, etc.

Una generalización del Job shop es el Job shop flexible, que consiste en un número determinado de máquinas y de trabajo, cada trabajo tiene un número determinado de operaciones ligados a su respectiva secuencia, incluyendo máquinas paralelas en algunas de las estaciones de trabajo. En estos modelos de Job shop flexibles se tratan dos subproblemas simultáneamente: asignación de máquinas y secuenciación de operaciones (Pinedo, 2008).

Este tipo de problemas, tal como otros que se ajustan a sistemas de producción reales generalmente son abarcados en algoritmos NP (tiempo polinomial no determinístico), especialmente de tipo NP-Hard (iguales o más complejos que el NP) (Mastrolilli y Gambardella, 1999), los cuales son muy complejos y robustos de trabajar, lo que hace que no sean rápidos durante su ejecución en programas de computación y que virtualmente sean imposibles de resolver óptimamente. La ventaja que tiene la programación de secuencias es que permite establecer los plazos y también es flexible en cuanto a la forma de establecer los trabajos a ser programados (Hopp y Spearman, 1996).

Este tipo de problemas, si bien pueden llegar a ser resueltos a través de métodos exactos, para efectos prácticos tienden a ser demasiado complejos y dispendiosos para programar, por lo cual se pueden emplear diferentes métodos de aproximación. Este tipo de problemas por lo general son abarcados teniendo como objetivo la minimización del makespan (tiempo total de proceso) y para el caso particular del Job shop flexible tenemos las siguientes referencias:

Tabla 3.1 Referencias bibliográficas para FJSP minimizando makespan.

Fuente: Elaboración propia

AÑO	NOMBRE	AUTOR	PALABRAS CLAVE	METODO DE SOLUCION	OBJETIVO DEL MODELO	RESTRICCIONES O SUPUESTOS ADICIONALES	RESULTADOS
1993	Applying tabu search to the job-shop scheduling problem	Mauro Dell'Amico, Marco Trubian	No aplica	Búsqueda Tabú	Minimizar makespan	No aplica	De los 46 problemas utilizados que tienen solución óptima conocida, el algoritmo logra replicar 33 de esos óptimos, para los restantes el gap respecto al óptimo era menor del 1%. Para los 7 problemas abiertos que emplea el autor, el algoritmo obtuvo mejores cotas superiores para 5 de ellos.
1996	Tabu Search for the Flexible-Routing Job Shop Problem	John B. Chambers, J. Wesley Barnes	No aplica	Búsqueda Tabú	Minimizar makespan	Se asume que cada máquina puede realizar más de un tipo de operación, aun cuando mantiene la restricción de realizar una operación a la vez	Para algunas de las políticas de despacho flexibles y en algunos de los modelos utilizados (especialmente p1p2p3 y c1) utilizadas por los autores, se encontró que para algunas de ellas se mejoran las soluciones en un promedio de 10.2%, si bien se sacrifica la velocidad de procesamiento, llegando incluso a tiempos de 1100 segundos para los problemas más grandes
1996	A fast taboo search algorithm for the job shop problem	Eugeniusz Nowicki, Czeslaw Smutnicki	Scheduling; Heuristics; Job-Shop; Taboo Search	Búsqueda Tabú	Minimizar makespan	No aplica	1. Para las instancias de Fisher y Thompson, Adams y otros y Laurence: Se encontró que el algoritmo propuesto es más rápido (124 segundos frente a los 271 que puede hacer el segundo más rápido) y que de los 27 problemas trabajados en 25 de ellos puede igualar o mejorar los tiempos de los otros algoritmos. En cuanto al makespan, el algoritmo propuesto encuentra óptimos para la mayoría de problemas con resultados conocidos previamente.

1999	Effective Neighborhood Functions for the Flexible Job Shop Problem	Monaldo Mastrolilli, Luca Maria Gambardella	Flexible job shop, tabu search	Búsqueda Tabú	Minimizar makespan	No aplica	Respecto al Hudata, se demostró que este método fue al menos 15 veces más rápido en términos de tiempo computacional que otros métodos probados y mejoro 74 resultados en las 129 instancias probadas. Respecto al Dpdata y BC data se mostró 5 y 20 veces más rápido. Se obtuvieron 120 cotas superiores mejores y 77 soluciones óptimas. Para casos de Job shop flexible encontró 38 soluciones óptimas de 43 problemas utilizados.
2007	A genetic algorithm for the Flexible Job-shop Scheduling Problem	F. Pezzella, G. Morganti, G. Ciaschetti	Job-shop Scheduling; Genetic algorithms; Flexible manufacturing systems	Algoritmo genético	Minimizar makespan	No aplica	Este algoritmo ofrece mejores resultados respecto a otras versiones de algoritmos genéticos, encontrando rangos de desviaciones entre 1.5 y 10.5. En cuanto a los algoritmos de búsqueda tabú, los resultados son en su mayoría iguales al algoritmo genético propuesto, solamente siendo mejores en un único caso.
2007	A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems	Jie Gao, Linyan Sun, Mitsuo Gen	Flexible job shop scheduling; Genetic algorithms; Variable neighborhood descent; Critical path	Algoritmo genético híbrido (con variables de ascendencia de vecindario)	Minimizar makespan, minimizar la carga total de las máquinas y minimizar la carga de la maquina critica	No aplica	Con XWdata, el algoritmo propuesto obtiene mejores resultados que otros algoritmos utilizados, siendo al menos 1 unidad menor en todos los objetivos. Con BRdata, el makespan es mejor para algunos casos así como el makespan promedio, teniendo principalmente hacia el límite inferior para cada caso estudiado. Para el BCdata y DPdata ocurre algo similar pero lo valores son más próximos hacia los limites superiores. Para el HUdata se demostró que para la mayoría de los casos los errores relativos son menores en el algoritmo propuesto, encontrando diferencias aproximadas de hasta 0.3



<b>2008</b>	Flexible job shop scheduling with overlapping in operations	Parviz Fattahi, Fariborz Jolai, Jamal Arkat	Flexible job shop; Scheduling; Overlapping; Hierarchical approach; Simulated annealing	Algoritmo de recocido simulado	Minimizar makespan	Las operaciones se pueden traslapar	Se ha podido constatar que el algoritmo propuesto ofrece mejores resultados del makespan cuando trata con operaciones traslapadas, ofreciendo los mismos resultados a comparación de los métodos tradicionales para optimizar esta modalidad del problema
<b>2009</b>	An artificial immune algorithm for the flexible job-shop scheduling problem	A. Bagheri, M. Zandieh, Iraj Mahdavi, M. Yazdani	Scheduling; Flexible job-shop; Makespan; Artificial immune algorithm	Algoritmo Inmune artificial	Minimizar makespan	Existe flexibilidad parcial o total en los problemas a tratar (parcial cuando un subconjunto de máquinas aptas para realizar una operación y total cuando todas las maquinas pueden hacer la operación)	<ol style="list-style-type: none"> <li>1. Para el Kacem data, el algoritmo artificial inmune obtuvo mejores resultados en los valores de Cmax, la carga de trabajo por máquina y la carga de trabajo total respecto a la mayoría de los algoritmos usados, excepto con los algoritmos genéticos híbridos.</li> <li>2. En el caso del Fadata, los resultados indican que para problemas pequeños no existen diferencias entre los resultados de los algoritmos comparados, pero para los problemas más grandes se comienzan a ver diferencias significativas en los valores del Cmax a favor del algoritmo artificial inmune, siendo en algunos casos incluso más de un 30% menor que el de las aproximaciones planteadas.</li> <li>3. Para el BRdata y HUdata el algoritmo de Pezzella fue el que mejores resultados proporciono, dado que el promedio de los errores en este algoritmo fue menor.</li> </ol>
<b>2009</b>	Mathematical models for job-shop scheduling problems with routing and process plan flexibility	Cemal Özgüven, Lale Özbakır, Yasemin Yavuz	Job-shop scheduling; Routing flexibility; Process plan flexibility; Mixed-integer programming	Programación Lineal Entera Mixta	Minimizar makespan	No aplica	El primer modelo, en términos generales ofrece resultados óptimos en tiempos mucho más cortos y utilizando menos variables y restricciones. El segundo modelo encuentra soluciones óptimas en problemas más pequeños, pero en los más grandes hay un aumento considerable en el valor óptimo.

<b>2009</b>	A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems	Li-Ning Xing, Ying-Wu Chen, Peng Wang, Qing-Song Zhao, Jian Xiong	Combinatorial optimization; Ant Colony Optimization; Flexible Job Shop Scheduling	Algoritmo por optimización basada en colonia de hormigas	Minimizar makespan	Existe flexibilidad parcial o total en los problemas a tratar (parcial cuando un subconjunto de máquinas aptas para realizar una operación y total cuando todas las máquinas pueden hacer la operación)	Se mostró que este algoritmo en promedio muestra resultados iguales o por lo menos muy similares al óptimo (las desviaciones en su mayoría son menores de 2).
<b>2009</b>	Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability	Parviz Fattahi, Alireza Fallahi	Dynamic scheduling; Flexible job shop scheduling; Genetic algorithm; Stability; Efficiency	Algoritmo genético	Optimizar la eficiencia (minimizando una función del makespan) y la estabilidad (minimizando una función de desviaciones de tiempo de inicio y penalidades)	Se asume que hay un comportamiento más dinámico en el arribo de los trabajos, incluso después de realizada la programación, lo que implica tener en cuenta restricciones que impliquen penalidad en caso de que se tengan que reprogramar trabajos	En problemas pequeños, el algoritmo ofrece los mismo resultados óptimos del valor objetivo respecto del método de ramas que se usa para optimizar este tipo de problemas, con la salvedad de que el tiempo que demora en encontrar dicha solución es mucho menor (mientras en el algoritmo propuesto tarda máximo 2 segundos, con el otro método puede demorar más de 10 min en el peor de los casos). Para los problemas medianos, encuentra una solución dentro del intervalo determinado por la ramificación, sin embargo encuentra el mejor valor en cuestión de segundos. Para los problemas grandes si encuentra una solución, a diferencia de la ramificación.
<b>2010</b>	Discrepancy search for the flexible job shop scheduling problem	Abir Ben Hmida, Mohamed Haouari, Marie-José Huguet, Pierre López	Scheduling; Allocation; Flexible job shop; Discrepancy search; Neighborhood structures	Búsqueda por discrepancia	Minimizar makespan	El algoritmo planteado tiene restricciones de precedencia entre dos operaciones del mismo trabajo y de cada operación con otras de otro trabajo, y también tiene en cuenta variables de flexibilidad en las operaciones	Se obtuvo que la búsqueda por discrepancia obtiene mejor resultados cuando las variables de flexibilidad tienen valores grandes. En términos de desempeño, es mejor que los algoritmos usados para la comparación de resultados, si bien los valores en un principio son iguales.

<b>2010</b>	Parallel hybrid metaheuristics for the flexible job shop problem	Wojciech Bozejko, Mariusz Uchroński, Mieczysław Wodecki	Scheduling; Flexible job shop problem; Hybrid metaheuristics; Tabu search; Population-based algorithm	Metaheurísticas híbridas paralelas	Minimizar makespan	Las metaheurísticas están programadas en base a variables que representan permutaciones de las secuencias de todas las operaciones de cada trabajo en alguna de las máquinas; adicionalmente se deben cumplir teoremas y propiedades relacionadas con dichas variables.	Para ciertos paquetes de datos el método propuesto ofrece mejores resultados, sin embargo, se han evidenciado varios casos en el cual este método solo ofrece tiempos de ejecución más rápidos, sin mejorar realmente los resultados de otros algoritmos utilizados para la comparación.
<b>2010</b>	An effective heuristic for flexible job-shop scheduling problem with maintenance activities	Shijin Wang, Jianbo Yu	Flexible job-shop scheduling problema; Availability constraints; Filtered beam search; Preventive maintenance	Algoritmo de búsqueda de rayo filtrado	Minimizar makespan, minimizar la carga total de las máquinas y minimizar la carga de la máquina crítica	Existen actividades de mantenimiento preventivo, en consecuencia se consideran restricciones de disponibilidad de máquinas	Cuando se tiene suficientes recursos para los mantenimientos, los resultados muestran que el algoritmo propuesto ofrece mejores en cuanto a la carga total de las máquinas, pero no ocurre lo mismo para los otros dos objetivos, si bien la diferencia es de apenas 1 unidad. También se demuestra que el algoritmo solo funciona si hay suficientes recursos de mantenimiento.
<b>2011</b>	A Hybrid Algorithm for Flexible Job-shop Scheduling Problem	Jianchao Tang, Guoji Zhang, Binbin Lin, Bixi Zhang	Flexible job-shop scheduling problem; Chaos particle swarm optimization; Genetic algorithm	Algoritmo híbrido (algoritmo genético y optimización de enjambre de partículas)	Minimizar makespan	Basado en parámetros auto adaptable con el fin de evitar que las búsquedas caigan en una misma zona indefinidamente.	1. Con los datos de Kacem, el algoritmo híbrido obtenía los resultados más bajos, al igual que el algoritmo de Zhang, pero en un menor tiempo. 2. Con los datos de Brandimarte, el algoritmo híbrido mostró mejores resultados para los diferentes paquetes de datos, teniendo una desviación máxima de aprox. 5%

2011	Mixed integer goal programming models for the flexible job-shop scheduling problems with separable and non-separable sequence dependent setup times	Cemal Özgüven, Yasemin Yavuz, Lale Özbakır	Mixed integer programming; Flexible job shop; Process plan flexibility; Sequence dependent setup	Programación entera mixta	Minimizar el makespan y balancear la carga de las maquinas	Hay flexibilidad de planeación y también hay tiempos de alistamiento dependientes de la secuencia separables y no separables	Se demostró que cuando el alistamiento es separable, el programa arroja mejores resultados del makespan y de la carga de trabajos para los diferentes problemas tratados. Sin embargo, realizando pruebas estadísticas, se encontró que la diferencia entre los valores del makespan promedio en ambos casos es significativa, mientras que para la carga de trabajos no es significativa.
2011	A genetic algorithm with tabu search procedure for flexible job shop Scheduling with transportation constraints and bounded processing times	Q. Zhang, H.Manier, M.-A.Manier	Flexible Job Shop Scheduling Problem with Transportation; Bounded processing times; Genetic algorithm; Tabu search; Flexible manufacturing system; Robotic cell; Surface treatment facility; Disjunctive graph	Algoritmo genético con búsqueda tabú	Minimizar makespan y el almacenamiento (tiempo de espera antes y después de cada máquina mientras esta en producción)	Se asume que hay transporte de recursos entre operaciones, lo cual implica que hay restricciones de capacidad de transporte y del movimiento de trabajos entre puestos de trabajo, no hay restricciones de espera entre operaciones y también los tiempos de procesamiento son limitados.	Se observa que el algoritmo propuesto obtiene mejores resultados de makespan cuando se establece el almacenamiento mayor a 0, sino que se permite un valor en el cual el makespan es el óptimo para el problema tratado.
2011	A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization	Ghasem Moslehi, Mehdi Mahnam	Flexible job-shop scheduling; Multi-objective optimization; Particle swarm optimization; Local search	Optimización de enjambre de partículas con búsqueda local	Minimizar makespan, carga de trabajo total y máxima	No aplica	Para problemas pequeños, el algoritmo propuesto no tiene resultados diferentes al de otros utilizados en la comparación, sin embargo, tampoco hay algún algoritmo dominante en este grupo de problemas. Entre más grande el problema, se aprecia como el algoritmo muestra resultados más equitativos respecto a los demás, si

	and local search						bien los valores no reflejan diferencias entre los valores.
<b>2011</b>	Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm	Nasr Al-Hinai, T.Y.ElMekkawy	Robust; Stable; Flexible job shop scheduling problem; Machine breakdowns	Algoritmo genético híbrido	Minimizar makespan y optimizar la estabilidad minimizando la cantidad de paradas por averías.	Se asumen paradas por averías en las máquinas de manera aleatoria	Entre los valores de estabilidad obtenidos, se puede apreciar que los resultados son mejores para las pruebas más robustas (siendo casi en su totalidad negativas), al igual que se pudo comprobar que entre menor sean tanto la incidencia de las fallas como la frecuencia de las mismas, el sistema es más estable.
<b>2012</b>	An efficient knowledge-based algorithm for the flexible job shop scheduling problem	Hamid Karimi, Seyed Habib A. Rahmati, M. Zandieh	Flexible job shop scheduling problem; Variable neighborhood search; Knowledge module; Neighborhood structures; Knowledge-based algorithm	Búsqueda por vecindario variable con un módulo de conocimiento	Minimizar makespan	No aplica	Se comprobó que efectivamente, este algoritmo obtuvo resultados mejores que la búsqueda por vecindario convencional. De la misma forma demostró brindar mejores resultados en el makespan respecto a los demás algoritmos utilizados en la comparación.
<b>2012</b>	Modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems	Wannaporn Teekeng, Arit Thammano	Flexible Job-Shop Scheduling Problems; Genetic Algorithm; Fuzzy Roulette Wheel Selection; Hierarchical Clustering	Algoritmo genético (selección por ruleta difusa y cruces por agrupación jerárquica)	Minimizar makespan	No aplica	Se demostró que el algoritmo propuesto ofrecía resultados con un error relativo promedio menor (21) frente a otros algoritmos probados (22-27).
<b>2012</b>	Chemical-reaction optimization for solving fuzzy job-shop	Jun-qing Li, Quan-kePan	Fuzzy job-shop scheduling problem; Chemical-	Algoritmo híbrido de optimización por reacción química	Minimizar el máximo tiempo de terminación difusa	En este caso se consideran actividades de mantenimiento flexibles y los	Se demostró que los resultados del algoritmo propuesto, eran mejores, especialmente al evaluar valores promedios de la función objetivo, aunque se demostró que hasta los

	scheduling problem with flexible maintenance activities		reaction optimization; Tabu search; Flexible maintenance activity			tiempos de procesamiento se asumen como difusos	peores resultados eran mejores que los peores resultados de otros algoritmos probados.
<b>2013</b>	Evaluation of mathematical models for flexible job-shop scheduling problems	Yunus Demir, S. Kürsat Isleyen	Flexible job-shops; Scheduling; Mixed integer programming	Programación Lineal Entera Mixta	Minimizar makespan	Cada modelo empleado se basa en una variable binaria que tiene una función diferente: posición en la secuencia, precedencias o por periodos de tiempo de procesamiento	Se demuestra que el modelo M2, que usa variables de precedencia, obtuvo un mejor resultado, tanto en muestras pequeñas, así como un desempeño más eficiente en cuanto al uso de la CPU y al número de variables y restricciones del modelo
<b>2013</b>	An integrated search heuristic for large-scale flexible job shop scheduling problems	Yuan Yuan, Hua Xu	Scheduling; Flexible job shop; Harmony search; Large Neighborhood search; Makespan	Algoritmo integrado de búsqueda armónica híbrida y de búsqueda de vecindarios grandes	Minimizar makespan	No aplica	En BRData, el algoritmo propuesto es en la mayoría de los casos mejor que los algoritmos evolucionarios usados para la comparación, llegando a tener desviaciones cercanas incluido del 12%. En DPdata se demuestra que el algoritmo propuesto ofrece mejores resultados que los algoritmos usados para la comparación, excepto que las desviaciones por mucho llegan al 1%.
<b>2013</b>	A hybrid harmony search algorithm for the flexible job shop scheduling problem	Yuan Yuan, Hua Xu, Jiadong Yang	Scheduling; Flexible job shop; Harmony search; Local search; Neighborhood structure; Makespan	Algoritmo de búsqueda armónica híbrida	Minimizar makespan	Existe flexibilidad parcial o total en los problemas a tratar (parcial cuando un subconjunto de máquinas aptas para realizar una operación y total cuando todas las máquinas pueden hacer la operación)	Para el Kacem data arroja los mismos resultados óptimos de otros algoritmos utilizados, sin embargo, encuentra la respuesta óptima en un menor tiempo. Para el Fdata ocurre algo similar, solo que para algunas muestras, especialmente medianas, encuentra mejores soluciones en un tiempo más corto, teniendo desviaciones entre 0.1 y 5.6%. Para el BRdata ocurre lo mismo, solo que las desviaciones llegan hasta 10.7%
<b>2013</b>	Flexible job shop	Yuan Yuan, Hua Xu	Scheduling; Flexible job	Algoritmo híbrido de	Minimizar makespan	No aplica	Para el Kacem data, se prueba que el algoritmo propuesto es uno de los

	scheduling using hybrid differential evolution algorithms		shop; Differential evolution; Local search; Neighborhood structure; Makespan	evolución diferencial			más eficientes, si bien hay otros algoritmos que ofrecen resultados y rendimientos similares y en algunos casos mejores. Para el BRdata muestra que en promedio, los resultados del algoritmo propuesto son mejores que los otros contra los que se está realizando la comparación, si bien el desempeño computacional no es el mejor. Para el BCdata, se demuestra que el algoritmo propuesto es ofrece resultados óptimos en tiempos relativamente más cortos que otros algoritmos.
<b>2013</b>	A Hybrid Artificial Bee Colony Algorithm with Local Search for Flexible Job-Shop Scheduling Problem	Arit Thammano, Ajchara Phuang	Flexible job-shop scheduling problem; Hybrid artificial bee colony algorithm; Local search technique; Swarm intelligence	Algoritmo de colonia de abejas artificial con búsqueda local	Minimizar makespan	No aplica	Al observar la desviación promedio de los resultados de cada algoritmo implementado respecto al límite inferior de cada prueba, se demuestra que el algoritmo propuesto tiene un mejor resultado (17.8 vs valores por encima de 20 en los otros algoritmos)

Por otra parte, se pudo constatar que si bien existen múltiples aproximaciones al problema del Job shop flexible minimizando el makespan, no ocurre lo mismo cuando se buscan optimizar otro tipo de variables, como es el caso de aquellas relacionadas con el nivel de servicio, donde las referencias son mucho más escasas. Dentro de las pocas referencias se pueden destacar:

Tabla 3.2 Referencias bibliográficas para FJSP minimizando variables de nivel de servicio.

Fuente: Elaboración propia

AÑO	NOMBRE	AUTOR	METODO DE SOLUCION	OBJETIVO DEL MODELO	RESTRICCIONES O SUPUESTOS ADICIONALES	RESULTADOS
2002	Tardiness minimization in a flexible job shop: A tabu search approach	Cintia Rigão Scrich, Vinícius Amaral Armentano, Manuel Laguna	Búsqueda Tabú	Minimizar la tardanza total	Existe flexibilidad en los problemas a tratar y en este caso particular se manejan tres niveles de flexibilidad (baja, media y alta)	A través de la comparación con la regla de despacho MDD, se pudo constatar que en el caso de una búsqueda tabú jerárquica hay una influencia principalmente del factor de tardanza, dado que entre más alto sea el factor de tardanza, el porcentaje de mejoramiento aumenta. En el caso de la búsqueda tabú de múltiples comienzos, hay una influencia tanto del factor de tardanza como del nivel de flexibilidad en los resultados, dado que entre más alto sea el factor de tardanza, el porcentaje de mejoramiento aumenta y entre mayor sea el nivel de flexibilidad, el valor de la tardanza disminuye.
2002	Minimizing the total tardiness in a flexible job-shop	N. Zribi, I. Kacem, A. El-Kamel, P. Borne	Aprox. jerárquica con algoritmo genético y búsqueda tabú. Aprox. integrada con algoritmo genético.	Minimizar la tardanza total	No aplica	A través del uso de instancias, se comprobó que la aproximación integrada ofreció mejores resultados, las tardanza totales eran en promedio un 25% menor respecto a la aproximación jerárquica.
2002	Minimizing the total tardiness in a flexible job-shop	N. Zribi, I. Kacem, A. El-Kamel, P. Borne	Aprox. jerárquica utilizando algoritmo genético y búsqueda tabú. Aprox. integrada utilizando algoritmo genético.	Minimizar la tardanza total	No aplica	A través del uso de instancias, se comprobó que la aproximación integrada ofreció mejores resultados, las tardanza totales eran en promedio un 25% menor respecto a la aproximación jerárquica.
2012	A Hybrid EA for Reactive Flexible Job-shop	Lin Lin, Mitsuo Gen, Yan Liang, Katsuhisa	Algoritmo evolucionario hibrido	Minimizar la tardanza total de los trabajos.	Se asumen que hay relaciones predeterminadas de precedencia y si una operación no se pueda	Tras una comparación con otros algoritmos bajo el mismo entorno de Job shop reactivo, se encontró que el algoritmo propuesto obtenía mejores resultados al obtener tardanzas mucho más



	Scheduling	Ohno			realizar en una máquina, el tiempo de procesamiento se toma como un valor muy grande.	pequeñas, llegando a diferencias de incluso de 3 unidades de tiempo.
<b>2012</b>	Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm	James C. Chen, Cheng-Chun Wu, Chia-Wen Chen, Kou-Huang Chen	Algoritmo genético con algoritmo genético agrupado	Minimizar makespan, tardanza total y tiempo total de inactividad	No aplica	Se demuestra que para la producción de armas, los resultados de utilizar los algoritmos propuestos tienen un mejor desempeño en términos de la función objetivo respecto a la programación de producción utilizada (EDD y SPT).
<b>2013</b>	An investigation into minimizing total energy consumption and total weighted tardiness in job shops	Ying Liu, Haibo Dong, Niels Lohse, Sanja Petrovic, Nabil Gindy	Algoritmo genético sin clasificación dominante	Minimizar consumo total de energía y la tardanza total ponderada	Al estar implicado el ahorro de energía, el problema incluye variables y restricciones relacionadas con el consumo de energía de las máquinas.	Se encontró que el programa de producción encontrado con la meta heurística NSGA-II provee mejores soluciones para optimizar el consumo de energía, presentando un consumo menor de energía entre el 4.9 y el 66.1 %, pero que esta sacrifica el segundo objetivo, ya que el 50% de los trabajos se retrasan.
<b>2013</b>	An efficient algorithm for solving the flexible job shop scheduling problem	Karin Thörnblad, Ann-Brith Strömberg, Michael Patriksson, Torgny Almgren	Modelos de indexación de tiempo (programación lineal entera mixta)	Minimizar makespan y tardanza	No aplica	Se pudo comprobar que, para el makespan solo fue posible encontrar el valor óptimo para instancias pequeñas, siendo encontradas para las demás instancias valores factibles con un desfase de mínimo 10%. Para el caso de la tardanza, el modelo propuesto no encontró valores óptimos pero si valores factibles con un desfase entre 1% y 4%.
<b>2013</b>	On the Optimization of Schedules of a Multitask Production Cell	Karin Thörnblad	Modelos de indexación de tiempo (programación lineal entera mixta)	Minimizar la tardanza total y maximizar la utilización de las células.	Se considera la disponibilidad de personal al momento de realizar la programación. Se clasifican los productos de 3 categorías: ordenes planeadas, ordenes colocadas y trabajos en espera	Se comprobó que los modelos de indexación de tiempos utilizando variables de “clavo” tienen un desempeño superior respecto a los demás modelos en cuanto a tiempos de computación y al tamaño de instancias que puede procesar, dado que se observó que dicho modelo era capaz de procesar problemas de instancias grandes con tiempos computacionales racionales, a diferencia de los demás modelos.

<b>2013</b>	An efficient algorithm for solving the flexible job shop scheduling problem	Karin Thörnblad, Ann-Brith Strömberg, Michael Patriksson, Torgny Almgren	Modelos de indexación de tiempo (programación lineal entera mixta)	Minimizar makespan y tardanza	No aplica	Se pudo comprobar que, para el makespan solo fue posible encontrar el valor óptimo para instancias pequeñas, siendo encontradas para las demás instancias valores factibles con un desfase de mínimo 10%. Para el caso de la tardanza, el modelo propuesto no encontró valores óptimos pero si valores factibles con un desfase entre 1% y 4%.
<b>2013</b>	On the Optimization of Schedules of a Multitask Production Cell	Karin Thörnblad	Modelos de indexación de tiempo (programación lineal entera mixta)	Minimizar la tardanza total y maximizar la utilización de las células.	Se considera la disponibilidad de personal al momento de realizar la programación. Se clasifican los productos de 3 categorías: ordenes planeadas, ordenes colocadas y trabajos en espera	Se comprobó que los modelos de indexación de tiempos utilizando variables de “clavo” tienen un desempeño superior respecto a los demás modelos en cuanto a tiempos de computación y al tamaño de instancias que puede procesar, dado que se observó que dicho modelo era capaz de procesar problemas de instancias grandes con tiempos computacionales racionales, a diferencia de los demás modelos.

## 4 Desarrollo del modelo a emplear

### 4.1 Modelo matemático

Los primeros modelos matemáticos utilizados para la programación de la producción datan de la década de 1960 (Özgüven, 2010). Los primeros trabajos sobre estos modelos formulan el problema de la secuenciación de trabajos mediante el uso de programación entera y con el paso de los años se ha abarcado este problema desde diferentes perspectivas. Si bien estos métodos exactos aun cuando dan con la solución óptima de los problemas no son necesariamente los más eficientes en cuanto a procesamiento y capacidad de cálculo se refiere.

Cuando estos problemas son demasiado complejos para ser solucionados a través de métodos exactos, se debe utilizar otros métodos que permitan la obtención de soluciones que sean lo más próximo posible al óptimo y así tener un buen entendimiento del sistema estudiado.

Se realizó un trabajo en el cual se realizaba el análisis de tres clases de modelos de programación entera (Demir e Isleyen, 2013), los cuales se diferencian inicialmente en el tipo de variables utilizadas para tomar las decisiones y a partir de este punto se realizan formulaciones de acuerdo a los datos que se requieren para realizar la asignación y secuenciación de trabajos.

Las variables que en dicho trabajo se emplean son:

- Variables indexadas en el tiempo: La variable de decisión se hace cargo de determinar en qué periodo se debe procesar una operación en su máquina respectiva, vendría a ser el modelo más complejo dado que el tiempo se toma como un parámetro discreto e implica distribuirlo en las unidades más cortas posibles para poder realizar la planeación.
- Variables de precedencia: Los modelos que emplean este tipo de variables son los más trabajados y los que generalmente ofrecen mejores resultados en cuanto a tiempos de procesamiento teniendo en cuenta su relativamente baja complejidad (emplea

una cantidad inferior de variables y restricciones). La variable de decisión define cuál es la operación que se realiza primero en una determinada máquina, de lo contrario, la solución no sería factible.

- Variables de secuencia-posición: Este tipo de variables tienen como base el supuesto de que cada máquina tiene una secuencia, la cual contiene un número específico de posiciones; por lo tanto esta variable define cuál es la posición de cada operación dentro de la secuencia de cada máquina.

En el trabajo de Demir e Isleyen, se propone la utilización de un modelo de precedencia dado que es el que mejores resultados en cuanto al makespan que se obtiene para cada instancia y también demostró ser el que menos tiempo de procesamiento requería para obtener ese resultado. Para este caso se empleará un modelo de este tipo dado que, en primer lugar, no existe mucha literatura relacionado con este tipo de modelos trabajados para objetivos ligados al nivel de servicio (contrario a los otros modelos, donde los modelos de precedencia han sido tratados y utilizados con mayor frecuencia en la literatura) y también debido a que el nivel de complejidad de este tipo de modelos es el apropiado para realizar un análisis del sistema estudiado.

#### **4.2 Supuestos del modelo**

- Se trabajan con indicadores relacionados con la entrega de los trabajos dado son los que la empresa considera que tiene un mayor impacto en su desempeño.

- Se asume que al principio de la operación todas las máquinas están disponibles y pueden trabajar en un trabajo a la vez y que ningún trabajo puede iniciar una operación sin haber terminado la operación anterior o hasta que hayan máquinas disponibles para realizar dicha operación, es decir, que no se pueden ejecutar más de una operación de un trabajo al tiempo.

- Siempre hay personal disponible para realizar cada operación.

- Los tiempos de transporte entre operaciones y tiempos de alistamiento de máquinas son despreciables.
- No existen paradas de emergencia a lo largo del proceso y que todos los trabajos tienen la misma importancia, es decir, que al no haber prioridad en los trabajos, una vez comenzada una operación en una máquina, esta permanece ocupada hasta que termine su trabajo.
- Contrario a muchos sistemas de este tipo, donde existe la posibilidad de que algún trabajo tenga que pasar varias veces por alguna operación en particular, también conocido como recirculación del trabajo, en este sistema en particular no se presenta esta característica, los trabajos pasan de una estación a la siguiente apenas vayan concluyendo sus respectivas operaciones debido básicamente a que por la naturaleza del producto y de la materia prima utilizada no existe la posibilidad de reproceso, sin embargo, este supuesto se puede despreciar según el caso en que sea aplicado dado que el modelo se plantea para que sea útil incluso en caso de que este supuesto no se cumpla.

### **4.3 Definición de índices y conjuntos**

Este es un problema que contiene  $m$  máquinas y  $n$  trabajos. Cada trabajo tiene una ruta de operaciones preestablecida anteriormente y cada operación se puede procesar en un conjunto de máquinas disponibles. La siguiente es la notación para los índices, parámetros y variables utilizadas en el modelo:

#### **Índices:**

*i*: índice de los trabajos

*j*: índice de las operaciones

*k*: índice de las máquinas

*l*: índice de las posiciones

#### **Parámetros:**

*n*: número total de trabajos

$m$ : número total de máquinas

$a_{kij}$ : indica las máquinas habilitadas para realizar la operación  $O_{ij}$

$$a_{kij} ::= \begin{cases} 1 & \text{si } O_{ij} \text{ se puede hacer en la máquina } k \\ 0 & \text{de lo contrario} \end{cases}$$

$p_{kij}$ : tiempo de proceso de la operación  $O_{ij}$  en la máquina  $k$

$M$ : Número muy grande

$d_i$ : due date del trabajo  $i$

**Variables de decisión:**

$x_{ijkl}$ : variable que determina la decisión de secuenciar  $O_{ij}$

$$x_{ijkl} ::= \begin{cases} 1 & \text{si } O_{ij} \text{ se hace en la máquina } k \text{ en la posición } l \\ 0 & \text{de lo contrario} \end{cases}$$

$v_{ijk}$ : variable que determina la decisión de asignar  $O_{ij}$

$$v_{ijk} ::= \begin{cases} 1 & \text{si } O_{ij} \text{ se hace en la máquina } k \\ 0 & \text{de lo contrario} \end{cases}$$

$TM_{kl}$ : tiempo de inicio de la máquina  $k$  en la posición  $l$

$PS_{ij}$ : tiempo de proceso de  $O_{ij}$  una vez se haya elegido una máquina

$ti_{ij}$ : tiempo de inicio de  $O_{ij}$

$c_i$ : tiempo requerido para completar el trabajo  $i$

$T_i$ : tardanza del trabajo

$U_i$ : especifica si el trabajo es tardío

$$U_i ::= \begin{cases} 1 & \text{si } c_i - d_i > 0 \\ 0 & \text{de lo contrario} \end{cases}$$

Para efectos de que el modelo sea funcional al momento de ejecutarlo es necesario que  $x_{ijkl}$  y  $v_{ijk}$  estén presentes en la programación, aun cuando aparentemente sean redundantes, esto se debe a que mientras  $v_{ijk}$  se encarga de la asignación de las operaciones,

$x_{ijkl}$  se encarga de ubicar cada operación en la posición que le corresponde dentro de la secuencia de la máquina a la cual fue asignada.

Para el modelo propuesto como el mejor por Demir e Isleyen la notación es la siguiente:

**Índices:**

$i, h$ : índice de los trabajos

$j, g$ : índice de las operaciones

$k$ : índice de las máquinas

**Parámetros:**

$n$ : número total de trabajos

$m$ : número total de máquinas

$p_{kij}$ : tiempo de proceso de la operación  $O_{ij}$  en la máquina  $k$

$M$ : Número muy grande

$d_i$ : due date del trabajo  $i$

**Variables de decisión:**

$z_{ijhghk}$ : variable que determina la decisión de secuenciar  $O_{ij}$

$z_{ijhghk} : \begin{cases} 1 & \text{si } O_{ij} \text{ precede a } O_{hg} \text{ en la máquina } k \text{ en la posición } l \\ 0 & \text{de lo contrario} \end{cases}$

$v_{ijk}$ : variable que determina la decisión de asignar  $O_{ij}$

$v_{ijk} : \begin{cases} 1 & \text{si } O_{ij} \text{ se hace en la máquina } k \\ 0 & \text{de lo contrario} \end{cases}$

$t_{ij}$ : tiempo de inicio de  $O_{ij}$

$c_i$ : tiempo requerido para completar el trabajo  $i$

$c_{ijk}$ : tiempo requerido para completar  $O_{ij}$  en la máquina  $k$

$s_{ijk}$ : tiempo de inicio de  $O_{ij}$  en la máquina  $k$

$T_i$ : tardanza del trabajo

$U_i$ : especifica si el trabajo es tardío

$$U_i ::= \begin{cases} 1 & \text{si } c_i - d_i > 0 \\ 0 & \text{de lo contrario} \end{cases}$$

De una forma similar al anterior modelo, en este caso  $z_{ijhgk}$  y  $v_{ijk}$  aparentemente son redundantes, sin embargo, ambos son necesarios dado que  $v_{ijk}$  se encarga de realizar la asignación de las operaciones en su respectiva máquina, mientras que  $z_{ijhgk}$  busca secuenciar las operaciones basado en precedencias.

#### 4.4 Formulación del modelo

Como se dijo anteriormente, el modelo a emplear contiene variables de secuenciación. Los modelos que utilizan este tipo de variables fueron utilizados inicialmente para formular problemas de Job Shop tradicional (Wagner, 1959) y luego fueron adaptados para trabajos con problemas de Job Shop flexible (Lee et al., 2002 y Fattahi, 2007). A continuación se presenta la estructura del modelo, teniendo como función objetivo minimizar el número total de trabajos tardíos (basado en Demir e Isleyen, 2013):

$$\text{Min } z = \sum_i U_i$$

$$c_i \geq t_{ij} + PS_{ij} \quad \forall i \quad (1.1)$$

$$T_i \geq c_i - d_i \quad \forall i \quad (1.2)$$

$$T_i \leq U_i * M \quad \forall i \quad (1.3)$$

$$\sum_k p_{kij} * v_{ijk} = PS_{ij} \quad \forall i, j \quad (1.4)$$

$$t_{ij} + PS_{ij} \leq t_{ij+1} \quad \forall i, \forall j = 1, \dots, J_i - 1 \quad (1.5)$$

$$TM_{kl} + PS_{ij} * x_{ijkl} \leq TM_{kl+1} \quad \forall i, j, k, \forall l = 1, \dots, d_k - 1 \quad (1.6)$$

$$TM_{kl} \leq t_{ij} + (1 - x_{ijkl}) * M \quad \forall i, j, k, l \quad (1.7)$$

$$TM_{kl} + (1 - x_{ijkl}) * M \geq t_{ij} \quad \forall i, j, k, l \quad (1.8)$$



$$v_{ijk} \leq a_{kij} \quad \forall i, j, k \quad (1.9)$$

$$\sum_i \sum_j x_{ijkl} = 1 \quad \forall k, l \quad (1.10)$$

$$\sum_k v_{ijk} = 1 \quad \forall i, j \quad (1.11)$$

$$\sum_l x_{ijkl} = v_{ijk} \quad \forall i, j, k \quad (1.12)$$

$$ti_{ij}, PS_{ij} \geq 0 \quad \forall i, j \quad (1.13)$$

$$TM_{kl} \geq 0 \quad \forall k, l \quad (1.14)$$

$$x_{ijkl} \in \{0,1\} \quad \forall i, j, k, l \quad (1.15)$$

$$v_{ijk} \in \{0,1\} \quad \forall i, j, k \quad (1.16)$$

La restricción (1.1) define el tiempo requerido para completar el trabajo  $i$ . Las restricciones (1.2) y (1.3) definen las tardanzas de cada trabajo. La restricción (1.4) determina el tiempo de proceso de cada operación. La restricción (1.5) asegura que se respete la precedencia entre las operaciones (una operación no puede comenzar hasta que su precedente haya sido terminado). La restricción (1.6) asegura que cada máquina solo pueda procesar una operación a la vez, esta restricción es de carácter no lineal, lo cual agrega complejidad al modelo. Las restricciones (1.7) y (1.8) aseguran que una operación no puede iniciar hasta que la maquina a la cual fue asignada esté disponible y hasta que su precedente no haya sido terminada. La restricción (1.9) establece la relación entre las máquinas asignadas y las disponibles para cada operación a asignar. La restricción (1.10) se encarga de la asignación de cada operación a una máquina y en una posición determinada. Las restricciones (1.11) y (1.12) aseguran que cada operación efectivamente se procese en una máquina y en la posición que fue asignada. Las demás restricciones garantizan la no negatividad de las variables de decisión y en los casos de las variables binarias garantizan que los valores tomados sean realmente binarios.

Adicionalmente a esto se presenta a continuación a modo ilustrativo una adaptación del modelo de precedencia propuesto por Demir e Isleyen (2013), el cual tiene la siguiente estructura:

$$\text{Min } z = \sum_i U_i$$

$$c_{ijk} \geq t_{ij} + p_{kij} \quad \forall i, j = J_i \quad (2.1)$$

$$c_i \geq \sum_{k \in M_{ij}} c_{ijk} \quad \forall i, j = J_i \quad (2.2)$$

$$T_i \geq c_i - d_i \quad \forall i \quad (2.3)$$

$$T_i \leq U_i * M \quad \forall i \quad (2.4)$$

$$s_{ijk} + c_{ijk} \leq v_{ijk} * M \quad \forall i, j, \forall k \in M_{ij} \quad (2.5)$$

$$c_{ijk} \geq s_{ijk} + p_{kij} - (1 - v_{ijk}) * M \quad \forall i, j, \forall k \in M_{ij} \quad (2.6)$$

$$s_{ijk} \geq c_{hjk} - (z_{ijhjk}) * M \quad \forall i \leq h, \forall j, g, \forall k \in M_{ij} \cap M_{hg} \quad (2.7)$$

$$s_{hjk} \geq c_{ijk} - (1 - z_{ijhjk}) * M \quad \forall i \leq h, \forall j, g, \forall k \in M_{ij} \cap M_{hg} \quad (2.8)$$

$$\sum_{k \in M_{ij}} s_{ijk} \geq \sum_{k \in M_{ij}} c_{ij-1k} \quad \forall i, \forall j = 2, \dots, J_i \quad (2.9)$$

$$\sum_{k \in M_{ij}} v_{ijk} = 1 \quad \forall i, j \quad (2.10)$$

$$s_{ijk}, c_{ijk} \geq 0 \quad \forall i, j, k \quad (2.11)$$

$$c_i \geq 0 \quad \forall i \quad (2.12)$$

$$v_{ijk} \in \{0,1\} \quad \forall i, j, k \quad (2.13)$$

$$z_{ijhjk} \in \{0,1\} \quad \forall i \leq h, \forall j, g, \forall k \in M_{ij} \cap M_{hg} \quad (2.14)$$

La restricción (2.1) y (2.2) definen el tiempo requerido para completar el trabajo  $i$ . Las restricciones (2.3) y (2.4) definen las tardanzas de cada trabajo. La restricción (2.5) asegura que los tiempos de inicio y terminación de una operación en caso de que no se asigne a una máquina. La restricción (2.6) determina el tiempo de proceso de cada operación. Las restricciones (2.7) y (2.8) se encargan de que dos operaciones no se procesen al mismo en caso de que existan máquinas que puedan procesar ambas. La

restricción (2.9) asegura que se respete la precedencia entre las operaciones (una operación no puede comenzar hasta que su precedente haya sido terminado). La restricción (2.10) asegura que cada operación efectivamente se procese en una máquina. Las demás restricciones garantizan la no negatividad de las variables de decisión y en los casos de las variables binarias garantizan que los valores tomados sean realmente binarios.

#### **4.5 Aplicación del modelo matemático**

El modelo planteado con variables de secuencia posición fue probado en instancias hechas por Fattahi (2007), usando la regla lexicográfica para encontrar un makespan y se generaron tiempos de entrega aleatorios entre  $0.5 C_{max}$  y  $1.5 C_{max}$  (Thörnblad et al, 2013). Para la ejecución de este modelo se utilizó el software GAMS versión 23.5.2 y se empleó un computador con procesador de 3.2 GHz y 4 GB de RAM. Como medida de respaldo se empleo también una plataforma on-line que permite el procesamiento de estos modelos en tiempos mucho más cortos y usando modelos más complejos de los que podría permitir GAMS.

Este modelo solo obtuvo soluciones óptimas en tiempos computacionales razonables para instancias muy pequeñas, siendo estas soluciones infactibles para problemas más grandes. La razón fundamental para que estos resultados se hayan presentado radica en la complejidad del modelo, dado que este problema por su naturaleza NP-Hard, la cantidad de restricciones, variables y parámetros a procesar aumenta exponencialmente; y a esto debemos añadir el hecho de que la restricción (1.6) consta de un término no lineal (producto de dos variables), lo cual hace que la complejidad sea aun mayor y dificulte la obtención de una resultado factible si asumimos que todas las restricciones deben cumplirse obligatoriamente. Como consecuencia de esto, el método de aproximación desarrollado tiene como finalidad encontrar cotas superiores a las soluciones de estas instancias.

Tabla 4.1 Instancias de Fattahi usadas para probar el modelo matemático

Fuente: Elaboración propia

Instancia	Número de trabajos	Número de operaciones	Número de máquinas	Umin
Fat01	2	2	2	1
Fat03	3	2	2	NA
Fat06	3	3	3	NA
Fat07	3	3	5	NA

#### 4.6 Métodos de aproximación

Anteriormente se expresaba que dada la alta complejidad para solucionar este tipo de problemas por métodos exactos, representados por un alto consumo de recursos computacionales (tiempo y memoria RAM), se plantea la opción de desarrollarlos a través de métodos de aproximación (heurísticas), los cuales encuentran en tiempos considerablemente más cortos una solución factible que este lo más cercano posible al óptimo.

Para el caso del sistema estudiado se empleará la búsqueda tabú, la cual es considerada como una metaheurística de búsqueda agresiva, en otras palabras, evita que la búsqueda quede atrapada en un óptimo local (Riojas, 2005). Se caracteriza por su uso de la memoria, la cual puede ser de corto plazo (almacena soluciones de iteraciones recientes para evitar ciclos) o de largo plazo (el cual diversifica la búsqueda tratando de analizar la mayor cantidad de sectores posibles o intensificar la búsqueda sobre algún atributo que tenga mayor influencia en la consecución de mejores soluciones); dicha memoria está basada en la construcción de lo que se conoce como una lista tabú y un mecanismo que permite realizar una selección apropiada del siguiente movimiento.

En la lista tabú se realiza un registro de todas las soluciones o movimientos que no se deben elegir, bien pueden ser soluciones o movimientos visitados anteriormente, el inverso de dichas soluciones o movimientos o también se podrían considerar algún aspecto en particular de dichas soluciones. El tamaño de la lista tabú equivale al número de iteraciones o movimientos que permanece un elemento dentro de la lista. Existen casos donde los elementos de la lista tabú pueden ser “olvidados”, con lo cual se puede permitir que estos salgan de la lista antes de cumplirse el número de iteraciones que debe

permanecer ahí. Para estos casos se puede implementar un criterio de aspiración, los cuales se pueden clasificar de esta manera:

- Por default: Ocurre cuando todos los movimientos están dentro de la lista tabú y en la siguiente iteración se debe excluir de la lista aquel movimiento que tenga menos iteraciones pendientes dentro de la lista.
- Por objetivo: Ocurre cuando un movimiento que permite una mejora en el valor de la función objetivo es excluido de la lista.
- Por dirección de búsqueda: ocurre si se cumple que un movimiento que está encaminado hacia una solución determinada permite obtener una mejora y el movimiento realizado en esa iteración también logra una mejora en el valor de la función objetivo.

Para la solución inicial se procede, con el apoyo del diseño de un experimento, a encontrar una solución utilizando reglas de despacho; se procederá a realizar una explicación de lo que se realizó en el experimento y de sus resultados, sin embargo, como se aclarará más adelante, las diferencias entre el experimento y este caso que se está tratando hace que no se sigan al pie de la letra las conclusiones del experimento.

#### **4.6.1 Diseño del experimento**

Para el diseño del experimento que se desarrolló, que es de tipo exploratorio, se emplearon un paquete de instancias creadas por Chambers y Barnes (1996), las cuales incluyen el número de trabajos, el número de operaciones para cada trabajo y las máquinas disponibles para hacer cada operación. Dado que las instancias en sus condiciones originales no tienen el dato del tiempo de entrega, se optó por utilizar el procedimiento sugerido por Thörnblad et al. (2013), en el cual se utiliza una regla de despacho con la instancia original para establecer el makespan del mismo para luego generar aleatoriamente los tiempos de entrega para cada trabajo.

Para este caso se empleó una regla lexicográfica en cada instancia para determinar su makespan, es decir, se secuencian los trabajos en el orden en que aparecen en la instancia; y con este valor se generaron tiempos de entrega aleatorios distribuidos uniformemente entre  $0.5C_{max}$  y  $1.5C_{max}$ . Se seleccionó la distribución uniforme con el fin

de asegurar que cada valor del rango tenga la misma probabilidad de ser incluida y evitar sesgos o valores muy centrales. El rango fue seleccionado con el fin de garantizar que exista una probabilidad de encontrar una secuencia en la cual no se presenten trabajos tardíos.

Una vez se generaron los tiempos de entrega, se emplearon las siguientes reglas:

- EDD (Earliest Due Date o Tiempo de entrega más cercano): Los trabajos son organizados ascendentemente en base al momento de entrega.

- CR (Critical Ratio o Razón Crítica): Se calcula de acuerdo a:

$$CR = \frac{\text{Tiempo restante antes de la entrega}}{\text{Tiempo de procesamiento restante}}$$

La regla del CR establece que los trabajos son organizados del menor CR al mayor.

- MS (Minimum Slack u Holgura Mínima): el tiempo de holgura se define como:

$$TH = \text{Tiempo de entrega} - \text{Tiempo de procesamiento}$$

Esta regla establece que los trabajos son ordenados ascendentemente en base a su tiempo de holgura.

Se realizó una clasificación de las instancias según su tamaño, considerando que las instancias utilizadas contienen datos de 10 o 15 trabajos:

- Pequeño (10 trabajos, 10 operaciones y entre 11 y 13 máquinas disponibles).
- Mediano (15 trabajos, 10 operaciones y entre 11 y 13 máquinas disponibles).
- Grande (15 trabajos, 15 operaciones y entre 15 y 17 máquinas disponibles).

Para la obtención del makespan como para el desarrollo del experimento con las tres instancias se utilizó el software LEKIN y se empleó un computador con procesador de 3.2 GHz y 4 GB de RAM.

Para el experimento se contemplaron tres factores, asumiendo desde el principio ningún factor como poco significativo, los cuales fueron:

- Tamaño de instancia: Combina el número de trabajos, número de máquinas a emplear cada trabajo y tiempos de procesamiento.

- Número de estaciones con más de una máquina (1, 2 y 3), cuando habían estaciones con más de una máquina disponible los tiempos de procesamiento eran los mismos, luego la asignación se podía hacer en cualquiera de las máquinas y no cambiaba ni la secuenciación ni los resultados.

- Regla de despacho (EDD, MS y CR)

Para el desarrollo del experimento hay dos supuestos a tener en cuenta: el primero es que al igual que el caso real, no existen recirculaciones, por lo cual se consideró esta herramienta como conveniente para emplear en este caso. Lo segundo es que por limitaciones del software utilizado (el cual solo consideraba la no presencia de recirculaciones) y por el uso exclusivo de instancias que se adaptaban al software, se asumía que las estaciones eran de una sola máquina y en caso de que en alguna estación existiera más de una máquina, éstas eran homogéneas, por lo cual al momento de aplicar alguna regla de despacho tiene un comportamiento que se asemeja mucho más a un Job shop tradicional, por lo cual la asignación de máquinas no se consideraba y para el desarrollo de la programación básicamente se presentaba con un enfoque en la secuenciación de trabajos.

Para la obtención de los resultados del experimento se utilizó el software Minitab y el objetivo de dicho experimento era establecer el impacto de los tres factores en el comportamiento de tres variables de respuesta estudiadas: Tardanza máxima, Tardanza total y Número de trabajos tardíos.

A continuación se muestran los resultados del experimento, aclarando que en las graficas de araña los resultados representan la distribución de los resultados entre las tres reglas analizadas:

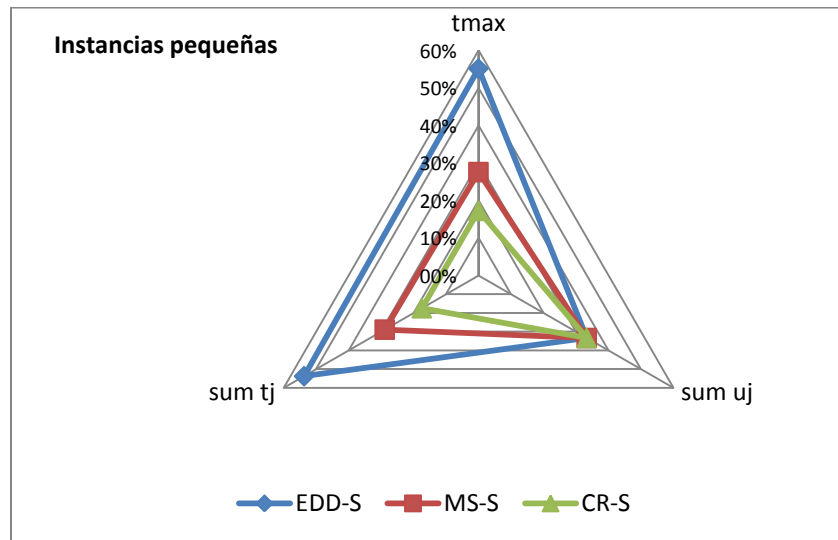


Figura 4.5.1 Resultados del experimento evaluando instancias pequeñas  
Fuente: Elaboración propia

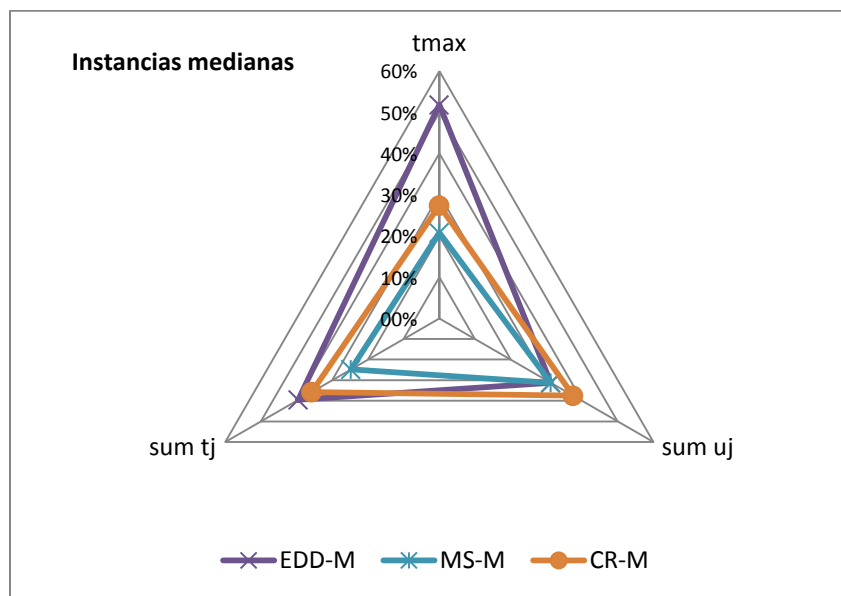


Figura 4.5.2 Resultados del experimento evaluando instancias medianas  
Fuente: Elaboración propia



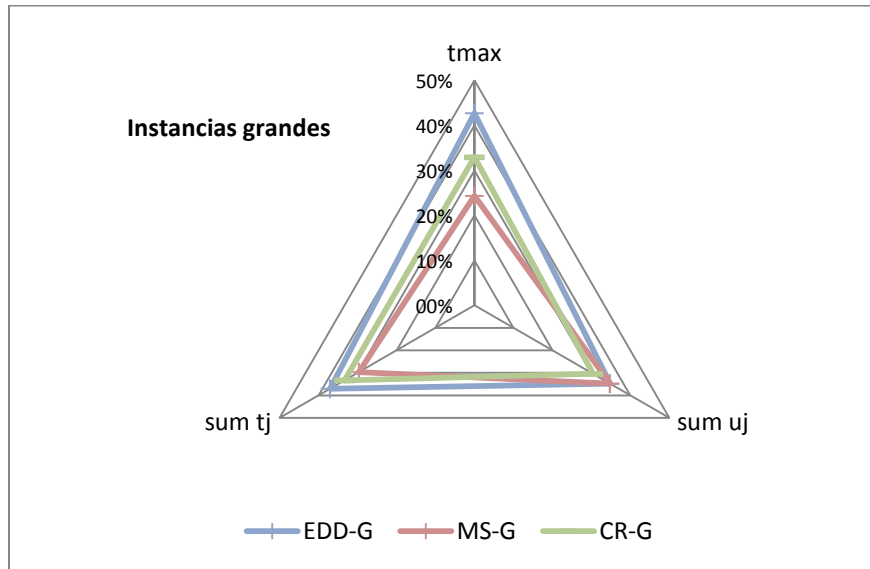


Figura 4.5.3 Resultados del experimento evaluando instancias grandes  
Fuente: Elaboración propia

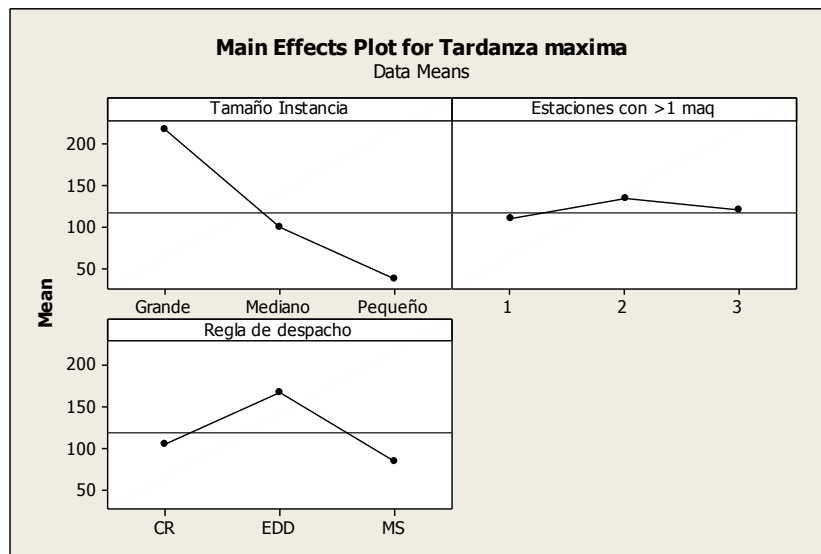


Figura 4.5.4 Resultados del experimento evaluando el impacto en la tardanza máxima  
Fuente: Elaboración propia

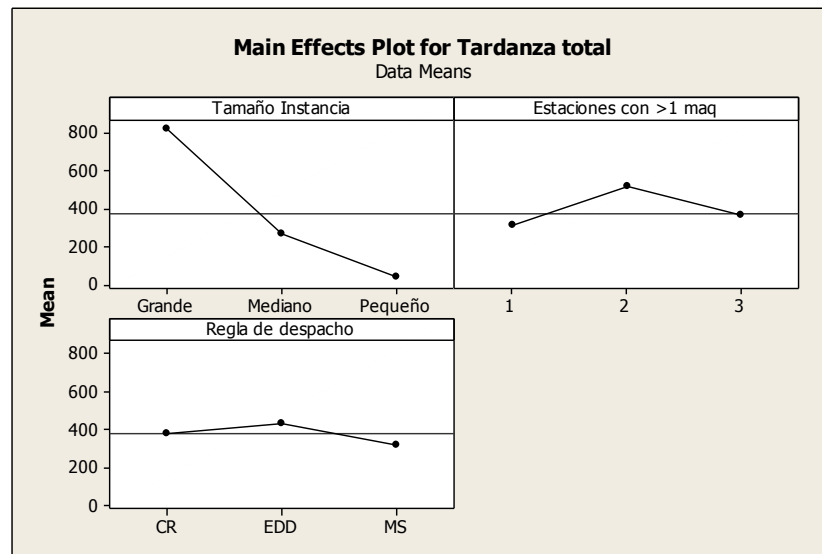


Figura 4.5.5 Resultados del experimento evaluando el impacto en la tardanza total  
Fuente: Elaboración propia

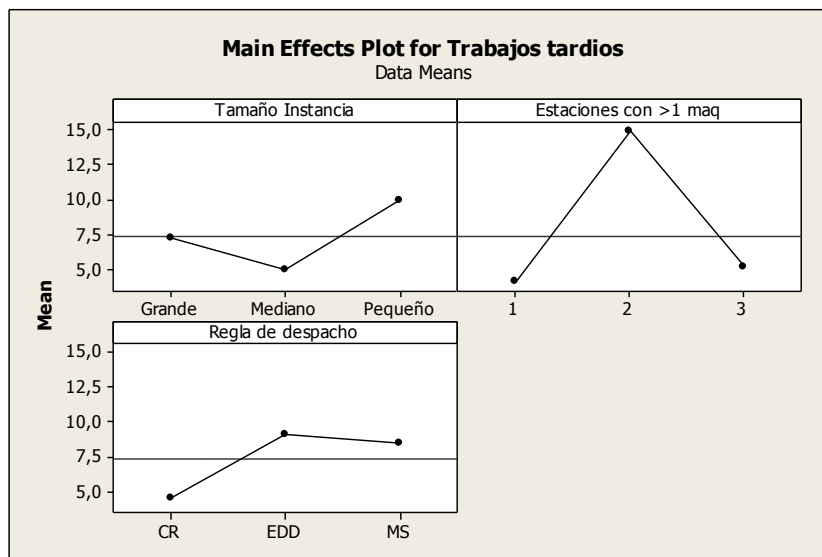


Figura 4.5.6 Resultados del experimento evaluando el impacto en el número de trabajos tardíos  
Fuente: Elaboración propia

Si se analiza particularmente el efecto de las reglas de despacho, se puede apreciar que, si bien todas se mantienen cercanas al impacto promedio, el impacto varía dependiendo de la variable a evaluar y del tamaño de la instancia evaluada. En términos generales la regla de despacho de la holgura mínima sería la regla de despacho que

convendría utilizar dado que sus resultados son los más favorables, sin embargo, tanto esta regla como la de radio crítico ofrecían mejores desempeños bajo las condiciones del experimento y dado que las condiciones del sistema varían según las características de cada trabajo, su desempeño no necesariamente es el más indicado.

Para la heurística implementada, al igual que en el experimento, se emplean instancias que contienen la misma información, con la diferencia de que no solo se utilizaron las mismas de experimento sino que se utilizaron otras creadas por diferentes autores, tales como Hurink (1994) y Fattahi (2007), las cuales se escogieron teniendo en cuenta un aspecto que los hiciera semejantes al sistema real: en cada uno de los tipos de productos que maneja la empresa la cantidad de operaciones es la misma y en consecuencia, los paquetes de instancias seleccionados tienen todos sus trabajos con la misma cantidad de operaciones.

#### **4.6.2 Heurística desarrollada**

De manera similar que en el experimento realizado, se utilizó la regla de despacho FIFO (primero que llega, primero que sale) para determinar el due date de los trabajos a través de la obtención del makespan; se usa dicha regla para la secuencia de los trabajos para luego ser asignados a la maquina que ofrece el menor tiempo de procesamiento. Para los due dates es asignado un valor aleatorio entre  $0.5 C_{max}$  y  $C_{max}$  para cada trabajo, esto debido a que al ser valores asignados aleatoriamente hay una probabilidad que no puede ser descartable de que el algoritmo asigne para todos los trabajos due dates que sean mayores al  $C_{max}$ .

Una vez se tiene el due date de cada trabajo se procede a crear la solución inicial, para esto se tomo la decisión de utilizar la regla de despacho EDD por su facilidad de implementación en cualquier tipo de sistema, por su capacidad de adaptarse a cualquier cambio en el mismo y también por el hecho de que esta regla de despacho tiene un mejor desempeño en sistemas donde los due dates tienen una gran influencia en su comportamiento, como en este caso.

Este EDD solo es aplicado para la secuenciación de trabajos, se incluye el supuesto de que la solución inicial mantendrá la misma asignación de trabajos que en el FIFO y el EDD obtiene una solución inicial tratando el problema como un Job shop clásico y a partir de esta configuración se procede a crear la lista tabú.

Para la lista tabú se tuvo en cuenta que para la gran mayoría de las instancias utilizadas ocurría que muchas de las operaciones solo se podían hacer en una única máquina y que, al igual que en el experimento, habían estaciones con más de una máquina disponible en las cuales los tiempos de procesamiento eran los mismos, en consecuencia la asignación se podía hacer en cualquiera de las máquinas sin afectar los resultados finales, si bien se podría contemplar en estudios posteriores un análisis de carga en cada máquina para analizar que opciones de optimización real tendría un cambio a otra máquina de la misma estación. Debido a esto se toma la decisión de no tener en cuenta cambios de asignación de trabajos y crear una lista tabú basado en movimientos de intercambio y movimiento de reordenamiento según ciertos parámetros.

En primera instancia se realiza una búsqueda tabú relativamente corta, cuyo parámetro principal es que se ejecuta un número máximo de iteraciones de 6, esto dado básicamente a las limitaciones tanto del software utilizado (Excel) como del computador utilizado. Los movimientos a realizar se basan en reestructurar la secuenciación ordenando ascendentemente las tardanzas que cada trabajo obtiene en la iteración anterior, luego se incluye la secuencia dentro de la lista tabú y permanece allí durante 2 iteraciones, si hay empate se aplica FIFO.

En caso de que la nueva secuenciación sea igual a alguna de las de la lista tabú se ordena descendientemente de acuerdo al due date. Si se llega al caso en que el algoritmo llegara a un valor de cero trabajos tardíos o se cumplen las 6 iteraciones, el algoritmo termina, se extrae el valor mínimo del número de trabajos tardíos y la iteración en la cual fue encontrada. A continuación se presenta una descripción paso a paso del funcionamiento del algoritmo.

### **Paso 1: Inicialización**

- Se extraen los datos existentes de la instancia y Se aplica la regla lexicográfica a los trabajos.

- Se calcula el  $C_{max}$  de la instancia y con base en el  $C_{max}$  se calculan los límites superior ( $C_{max}$ ) e inferior ( $0.5 C_{max}$ ) de los due dates y asignarlos aleatoriamente.

**Paso 2:**

- Se genera la solución inicial usando la regla de despacho EDD.
- Se almacena la secuencia y los valores de tardanza en cada trabajo, la tardanza total y el número de trabajos tardíos.

**Paso 3:**

- Se crea el primer vecino a partir de una nueva secuenciación ordenando los trabajos ascendentemente según su tardanza individual

- Se almacena la solución inicial en la lista tabú y se mantiene allí durante dos iteraciones.

- Si la nueva solución coincide con la inicial se crea otra ordenando descendentemente los trabajos según su due date.

- Si la nueva solución permite encontrar un número de trabajos tardíos de cero continua al paso 5, en caso contrario continua al paso 4.

**Paso 4:**

- Se repite el mismo procedimiento del paso 3 pero utilizando la solución anterior hasta un máximo de 6 iteraciones, ingresando cada solución anterior en la lista tabú y conservándola durante 2 iteraciones.

- Se almacena la mejor solución a medida que se realizan las iteraciones.

- Si la nueva solución permite encontrar un número de trabajos tardíos de cero o si se realizan las 6 iteraciones continua al paso 5.

**Paso 5: Finalización**

- Se extrae el valor de la mejor solución y la secuencia de dicha solución y se finaliza el algoritmo.

## 5 Pruebas computacionales aplicados a instancias

Para el procesamiento del algoritmo se utilizó el lenguaje de programación VBA de Excel y se empleó un computador con procesador de 3.2 GHz y 4 GB de RAM. Se utilizaron instancias creadas por Fattahi (Fat) y por Hurink, específicamente el s\_data (Hsd) (Behnke et al., 2012):

*Tabla 5.1* Instancias de Fattahi usadas para probar la metaheurística

Fuente: Elaboración propia

Instancia	Número de trabajos	Número de operaciones	Número de máquinas
<b>Fat01</b>	2	2	2
<b>Fat03</b>	3	2	2
<b>Fat06</b>	3	3	3
<b>Fat07</b>	3	3	5
<b>Fat10</b>	4	3	5
<b>Fat11</b>	5	3	6
<b>Fat13</b>	6	3	7
<b>Fat18</b>	9	4	8
<b>Fat20</b>	12	4	8

*Tabla 5.2* Instancias de Hurink (s\_data) usadas para probar la metaheurística

Fuente: Elaboración propia

Instancia	Número de trabajos	Número de operaciones	Número de máquinas
<b>Hsd01</b>	6	6	6
<b>Hsd02</b>	10	10	10
<b>Hsd16</b>	20	5	5
<b>Hsd25</b>	15	10	10
<b>Hsd29</b>	20	10	10
<b>Hsd55</b>	7	7	7

Para dichas instancias se corrió el algoritmo en 10 ocasiones, para las cuales se obtuvieron los siguientes resultados para el número de trabajos tardíos y para el tiempo computacional empleado para el procesamiento del algoritmo:

Tabla 5.3 Resultados obtenidos usando las instancias de Fattahi

Fuente: Elaboración propia

Instancia	CPU time (seg)			Número de trabajos tardíos				Número de iteraciones	
	Mejor	Promedio	Desviación	Mejor	Promedio	Mín.	Máx.	Necesario para llegar a $U_{\min}$	Promedio
<b>Fat01</b>	1,84	2,58	0,90	1	1	1	2	1	2
<b>Fat03</b>	1,79	1,98	0,16	1	2	1	3	1	3
<b>Fat06</b>	1,93	2,26	0,30	2	2	2	3	1	3
<b>Fat07</b>	1,79	2,40	0,42	0	2	0	3	1	2
<b>Fat10</b>	2,18	2,40	0,12	1	2	1	4	2	3
<b>Fat11</b>	2,54	2,81	0,28	1	4	1	5	4	3
<b>Fat13</b>	2,84	3,02	0,18	1	4	1	5	2	3
<b>Fat18</b>	3,61	5,05	0,70	0	3	0	6	2	2
<b>Fat20</b>	5,90	6,38	0,50	1	4	1	7	2	2

Tabla 5.4 Resultados obtenidos usando las instancias de Hurink (s\_data)

Fuente: Elaboración propia

Instancia	CPU time (seg)			Número de trabajos tardíos				Número de iteraciones	
	Mejor	Promedio	Desviación	Mejor	Promedio	Mín.	Máx.	Necesario para llegar a $U_{\min}$	Promedio
<b>Hsd01</b>	3,50	3,81	0,25	0	1	0	1	2	2
<b>Hsd02</b>	8,13	10,56	1,49	0	2	0	5	1	4
<b>Hsd16</b>	6,71	12,51	4,92	0	4	0	7	2	2
<b>Hsd25</b>	13,29	15,90	2,16	1	2	1	3	2	2
<b>Hsd29</b>	18,14	21,39	3,54	0	3	0	5	6	4
<b>Hsd55</b>	10,58	12,69	2,18	0	3	0	7	3	2

A raíz de los resultados anteriormente expuestos, se puede deducir que en términos generales esta metaheurística ofrece buenos resultados, ya que según la configuración de la solución inicial de los problemas, basada en los due dates que se calculen, está en capacidad de ofrecer secuenciaciones que permiten un cumplimiento de la totalidad de los trabajos; también cabe resaltar que en problemas grandes, la proporción de trabajos tardíos en promedio es baja a comparación de los problemas más pequeños y es especialmente notorio en las instancias de Hurink, las cuales son las que en un principio mostraron un mejor desempeño al ejecutar el algoritmo (Tablas 5.5 y 5.6):

Tabla 5.5 Desempeño de las instancias de Fattahi

Fuente: Elaboración propia

Instancia	Número de trabajos	Número de trabajos tardíos		% promedio trabajos tardíos
		Mejor	Promedio	
<b>Fat01</b>	2	1	1	50%
<b>Fat03</b>	3	1	2	67%
<b>Fat06</b>	3	2	2	67%
<b>Fat07</b>	3	0	2	67%
<b>Fat10</b>	4	1	2	50%
<b>Fat11</b>	5	1	4	80%
<b>Fat13</b>	6	1	4	67%
<b>Fat18</b>	9	0	3	33%
<b>Fat20</b>	12	1	4	30%

Tabla 5.6 Desempeño de las instancias de Hurink (s\_data)

Fuente: Elaboración propia

Instancia	Número de trabajos	Número de trabajos tardíos		% promedio trabajos tardíos
		Mejor	Promedio	
<b>Hsd01</b>	6	0	1	17%
<b>Hsd02</b>	10	0	2	20%
<b>Hsd16</b>	20	0	4	20%
<b>Hsd25</b>	15	1	2	13%
<b>Hsd29</b>	20	0	3	15%
<b>Hsd55</b>	7	0	3	43%

En términos del tiempo de computación se puede visualizar por una parte, que los resultados de aplicar el algoritmo muestran un mejor desempeño computacional de las instancias de Fattahi respecto a las de Hurink (Figura 5.1), ya que en las primeras, se observa que hay un menor tiempo de procesamiento y que los tiempos de procesamiento más cortos y los tiempos promedio de procesamiento son gráficamente muy cercanos y se mantiene esa tendencia para todos los tamaños de instancias. En el segundo grupo de instancias se observa algo diferente, entre más grande es la instancia, el tiempo de computación también crece y la diferencia entre los tiempos de procesamiento más cortos y los tiempos promedio de procesamiento son visiblemente más grandes (Figura 5.2).



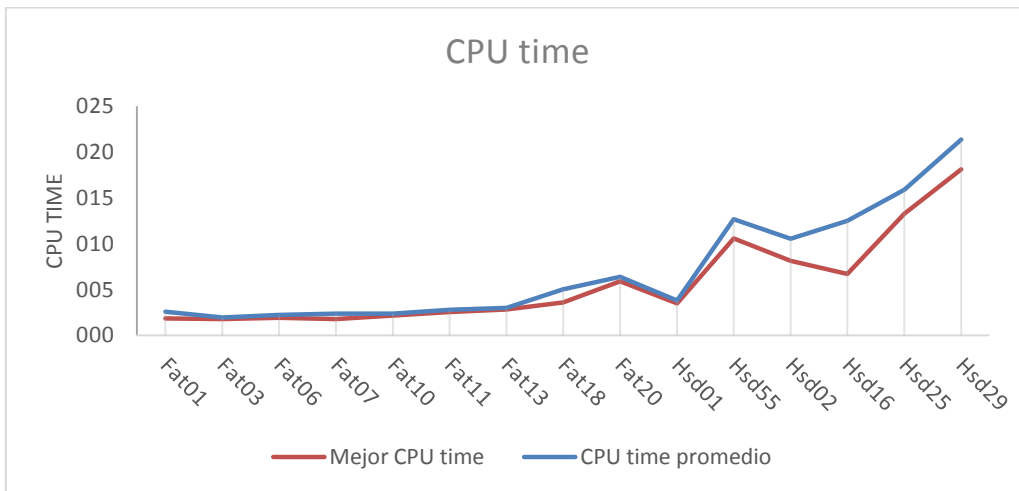


Figura 5.1 Gráfica Instancia vs. Tiempo de procesamiento

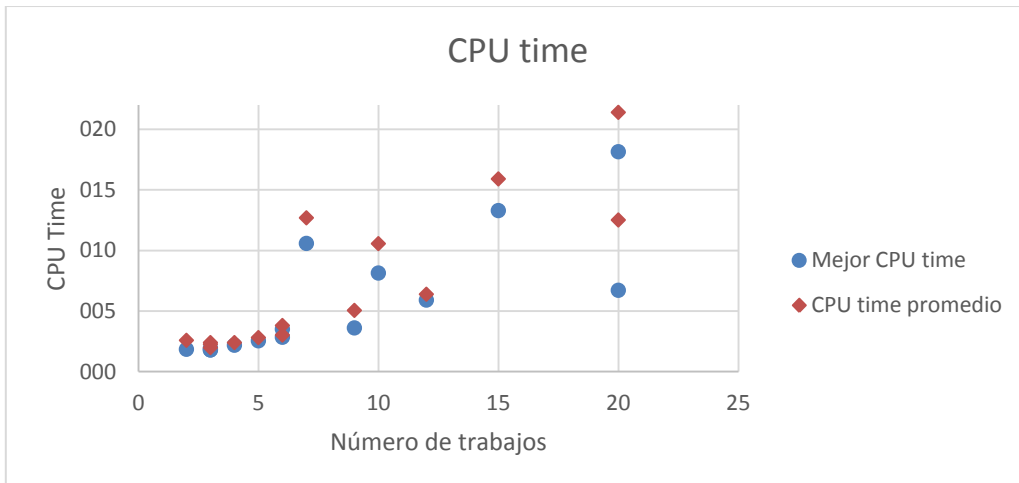


Figura 5.2 Gráfica Número de trabajos vs. Tiempo de procesamiento

En ocasiones se puede dar que por los valores asignados aleatoriamente por el due date el algoritmo permita encontrar valores de cero trabajos tardíos para ciertas instancias, tal como se pudo constatar anteriormente, sin embargo, precisamente por estos elementos aleatorios hace que la probabilidad de que se encuentren estos casos sea bastante baja. A continuación se presenta uno de dichos casos en los que el número de trabajos tardíos es cero:

Tabla 5.7 Ejemplo de secuenciación con cero trabajos tardíos (Instancia Fattahi 7)

Fuente: Elaboración propia

Trabajo (i)	Operación (j)	Máquina (M)	Tiempo inicio	Tiempo proceso	Tiempo fin	Due date
2	1	3	0	150	150	
2	2	3	150	55	205	
2	3	3	205	65	270	329
3	1	2	0	62	62	
3	2	4	62	70	132	
3	3	5	132	100	232	299
1	1	1	0	117	117	
1	2	2	117	130	247	
1	3	4	247	150	397	400

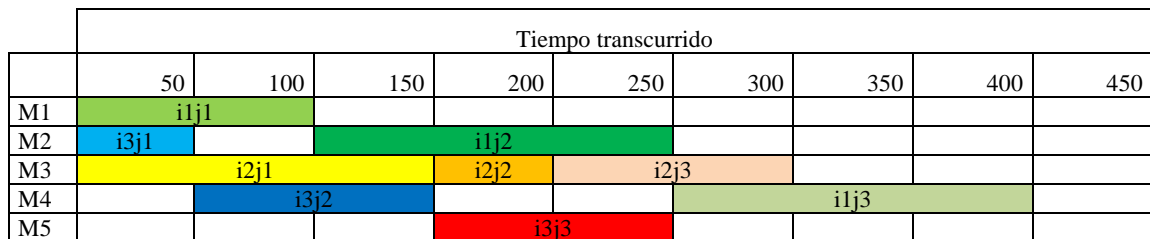


Figura 5.3 Diagrama de Gantt para el ejemplo con cero trabajos tardíos (Instancia Fattahi 7)

Fuente: Elaboración propia

## 6 Propuesta para la empresa

Para el caso de la empresa el procedimiento es similar al aplicado a las instancias, sin embargo, los due dates ya están establecidos, generalmente son determinados en común acuerdo con los clientes, por lo tanto no es necesario realizar cálculos para determinarlos y también le quita elementos aleatorios tanto al modelo exacto como a la metaheurística.

En base a los diagramas de flujo y multi-producto, las operaciones que serán incluidas serán las siguientes, aclarando nuevamente que dependiendo del tipo de producto el proceso puede variar:

*Tabla 6.1* Ejemplo de datos de la empresa estudiada

Fuente: Elaboración propia

Operación	No. Máquinas disponibles	Tiempo de proceso en min (brindada por la empresa)
<b>Mezclado</b>	2	24
<b>Preforma</b>	3	27
<b>Laminación</b>	2	23
<b>Metal caucho</b>	1	37
<b>Vulcanización</b>	3	8
<b>Rebado</b>	2	20

Estos tiempos se presentan de manera ilustrativa y corresponden al tiempo que se termina cada operación para procesar un “lote” de productos, dado que en cada pedido la cantidad de veces que pasa el molde por cada operación depende de la cantidad de producto que se pide, para este caso se asume y se trabaja con los tiempos para procesar un lote de productos. También se asume que el mismo tiempo aplica para todas las maquinas del conjunto que puede realizar cada operación.

Se tiene en cuenta que para este caso en particular, el número de máquinas en total y de operaciones por trabajo es el mismo. Hay que aclarar que para esta empresa en la mayoría de casos se incurre en trabajos tardíos debido principalmente a la diferencia de tamaño en los pedidos y en ocasiones más esporádicas hay tiempos muertos, de alistamiento, entre otros., que por motivos de fuerza mayor se deben asumir en la empresa y provocan mayores demoras en la realidad. En este modelo estos tiempos se obvian.

## 6.1 Modelo matemático

Para el caso del modelo matemático se realizó un ajuste: al tratar con máquinas homogéneas la secuenciación no depende de la máquina utilizada, por lo tanto la variable del tiempo de proceso utilizada anteriormente pasa a ser un parámetro del modelo, con lo cual se simplifica el modelo y adicionalmente permite retirar la no linealidad del modelo probado para las instancias dado que la restricción no lineal involucraba a esta variable.

A continuación se presenta la notación del nuevo modelo

### Índices:

*i*: índice de los trabajos

*j*: índice de las operaciones

*k*: índice de las máquinas

*l*: índice de las posiciones

### Parámetros:

*n*: número total de trabajos

*m*: número total de máquinas

$a_{kij}$ : indica las máquinas habilitadas para realizar la operación  $O_{ij}$

$$a_{kij} ::= \begin{cases} 1 & \text{si } O_{ij} \text{ se puede hacer en la máquina } k \\ 0 & \text{de lo contrario} \end{cases}$$

$p_{ij}$ : tiempo de proceso de la operación  $O_{ij}$

*M*: Número muy grande

$d_i$ : due date del trabajo *i*

### Variables de decisión:

$x_{ijkl}$ : variable que determina la decisión de secuenciar  $O_{ij}$

$$x_{ijkl} ::= \begin{cases} 1 & \text{si } O_{ij} \text{ se hace en la máquina } k \text{ en la posición } l \\ 0 & \text{de lo contrario} \end{cases}$$

$v_{ijk}$ : variable que determina la decisión de asignar  $O_{ij}$

$$v_{ijk} ::= \begin{cases} 1 & \text{si } O_{ij} \text{ se hace en la máquina } k \\ 0 & \text{de lo contrario} \end{cases}$$

$TM_{kl}$ : tiempo de inicio de la máquina  $k$  en la posición  $l$

$ti_{ij}$ : tiempo de inicio de  $O_{ij}$

$c_i$ : tiempo requerido para completar el trabajo  $i$

$T_i$ : tardanza del trabajo

$U_i$ : especifica si el trabajo es tardío

$$U_i ::= \begin{cases} 1 & \text{si } c_i - d_i > 0 \\ 0 & \text{de lo contrario} \end{cases}$$

Esta es la estructura del nuevo modelo

$$\text{Min } z = \sum_i U_i$$

$$c_i \geq ti_{ij} + PS_{ij} \quad \forall i \quad (3.1)$$

$$T_i \geq c_i - d_i \quad \forall i \quad (3.2)$$

$$T_i \leq U_i * M \quad \forall i \quad (3.3)$$

$$ti_{ij} + PS_{ij} \leq ti_{ij+1} \quad \forall i, \forall j = 1, \dots, J_i - 1 \quad (3.4)$$

$$TM_{kl} + p_{ij} * x_{ijkl} \leq TM_{kl+1} \quad \forall i, j, k, \forall l = 1, \dots, d_k - 1 \quad (3.5)$$

$$TM_{kl} \leq ti_{ij} + (1 - x_{ijkl}) * M \quad \forall i, j, k, l \quad (3.6)$$

$$TM_{kl} + (1 - x_{ijkl}) * M \geq ti_{ij} \quad \forall i, j, k, l \quad (3.7)$$

$$v_{ijk} \leq a_{kij} \quad \forall i, j, k \quad (3.8)$$

$$\sum_i \sum_j x_{ijkl} = 1 \quad \forall k, l \quad (3.9)$$

$$\sum_k v_{ijk} = 1 \quad \forall i, j \quad (3.10)$$

$$\sum_l x_{ijkl} = v_{ijk} \quad \forall i, j, k \quad (3.11)$$

$$ti_{ij}, PS_{ij} \geq 0 \quad \forall i, j \quad (3.12)$$

$$TM_{kl} \geq 0 \quad \forall k, l \quad (3.13)$$

$$x_{ijkl} \in \{0,1\} \quad \forall i, j, k, l \quad (3.14)$$

$$v_{ijk} \in \{0,1\} \quad \forall i, j, k \quad (3.15)$$

La restricción (3.1) define el tiempo requerido para completar el trabajo  $i$ . Las restricciones (3.2) y (3.3) definen las tardanzas de cada trabajo. La restricción (3.4) asegura que se respete la precedencia entre las operaciones (una operación no puede comenzar hasta que su precedente haya sido terminado). La restricción (3.5) asegura que cada máquina solo pueda procesar una operación a la vez, y con el cambio realizado esta restricción ya es lineal. Las restricciones (3.6) y (3.7) aseguran que una operación no puede iniciar hasta que la maquina a la cual fue asignada esté disponible y hasta que su precedente no haya sido terminada. La restricción (3.8) establece la relación entre las máquinas asignadas y las disponibles para cada operación a asignar. La restricción (3.9) se encarga de la asignación de cada operación a una máquina y en una posición determinada. Las restricciones (3.10) y (3.11) aseguran que cada operación efectivamente se procese en una máquina y en la posición que fue asignada. Las demás restricciones garantizan la no negatividad de las variables de decisión y en los casos de las variables binarias garantizan que los valores tomados sean realmente binarios.

## 6.2 Método de aproximación

Como se dijo inicialmente, ya no hay necesidad de realizar el cálculo los due dates, en consecuencia se excluye el uso de la regla de despacho FIFO en el experimento y se procede a emplear directamente la regla EDD para obtener una secuenciación inicial de los trabajos.

A partir de allí se puede aplicar la búsqueda tabú de la misma forma que con las instancias, realizando una búsqueda tabú con un máximo de 6 iteraciones, con movimientos que reestructuran la secuenciación ordenando ascendentemente las tardanzas que cada trabajo obtiene en la iteración anterior, incluyendo la secuencia dentro de la lista tabú

durante 2 iteraciones, aplicando FIFO en caso de que haya empate en las tardanzas. Si la nueva secuenciación sea igual a alguna de las de la lista tabú se orden descendientemente de acuerdo al due date. Si se llega al caso en que el algoritmo llegara a un valor de cero trabajos tardíos o se cumplen las 6 iteraciones, el algoritmo termina, se extrae el valor mínimo del número de trabajos tardíos y la iteración en la cual fue encontrada.

### 6.3 Pruebas computacionales y resultados

Se trabajaron dos casos diferentes: en el primero se tomaron los trabajos recibidos por la empresa en una semana escogida aleatoriamente y en el segundo se creó un caso en el cual solo hay tres trabajos, uno que tenga que pasar por laminado, uno que tenga que pasar por preforma y uno que tenga que pasar por metal-caucho. Para estos casos se procesaron los datos en el modelo matemático y en la metaheurística utilizada.

En el caso del modelo exacto el desempeño fue similar la versión del modelo utilizado en las instancias utilizadas, en los ejemplos diseñados el modelo obtuvo un valor óptimo en casos donde se presentan una cantidad baja de trabajos (máximo 4 trabajos), lo que llevaría, nuevamente, a que la metaheurística desarrollada sea útil para establecer cotas superiores para los problemas más grandes.

Para hacer el ejercicio en la metaheurística un poco más robusto en cuanto a resultados se realizaron 10 ejecuciones del algoritmo para los trabajos recibidos en dicha semana:

*Tabla 6.2* Ejemplo 1 de aplicación a datos de la empresa estudiada  
Fuente: Elaboración propia

n	o	m	CPU Time (seg)	U min	best CPU	CPU Time avg	desv CPU
7	4	13	9,33	2	5,543	7,079	1,253
			7,45	2			
			5,54	2			
			6,95	2			
			8,98	2			
			6,18	2			
			7,32	2			
			6,09	2			
			6,91	2			
			6,05	2			

Tabla 6.3 Ejemplo 2 de aplicación a datos de la empresa estudiada  
Fuente: Elaboración propia

n	o	m	CPU Time (seg)	U min	best CPU	CPU Time avg	desv CPU
3	4	13	10,46	1	6,891	8,215	1,255
			9,55	1			
			7,13	1			
			8,36	1			
			8,25	1			
			7,21	1			
			9,45	1			
			6,89	1			
			6,89	1			
			7,95	1			

A continuación se presenta un comparativo entre los resultados del modelo y de la metaheurística utilizada:

Tabla 6.4 Comparación de resultados entre el modelo exacto y la metaheurística  
Fuente: Elaboración propia

n	o	m	Óptimo U min	Busqueda Tabu			
				U min	Best CPU	CPU Time avg	Desv. CPU
2	4	13	0	0	8,22	6,65	1,25
3	4	13	0	1	6,89	8,22	1,25
4	4	13	0	1	6,59	8,33	1,25
5	4	13	-	2	6,29	8,45	1,25
6	4	13	-	2	7,05	8,69	1,25
7	4	13	-	2	5,54	7,08	1,25

Este fue uno de los ejemplos utilizados en el modelo matemático el cual tiene 4 trabajos y sus due dates son 113, 87, 104 y 129 minutos respectivamente y obtuvo un óptimo de cero trabajos tardíos.

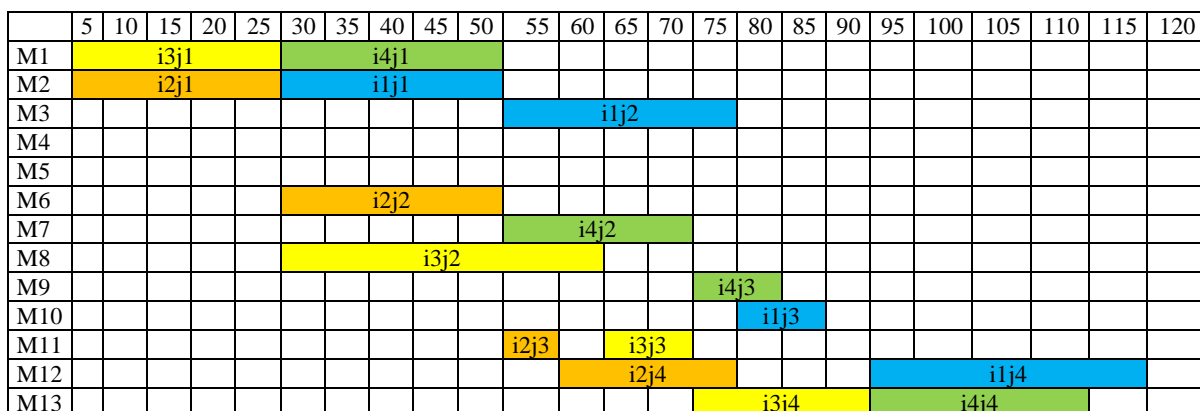


Figura 6.1 Diagrama de Gantt ejemplo usado en el modelo matemático  
Fuente: Elaboración propia



Se puede apreciar que para el caso de la empresa, este modelo aun contempla la opción de incurrir en trabajos tardíos teniendo en cuenta los due dates que maneja la empresa con sus clientes. Sin embargo, para futuras investigaciones estos resultados suponen un límite superior para estos ejemplos.

*Tabla 6.5* Secuenciación con datos de la empresa (ejemplo 1)

Fuente: Elaboración propia

Trabajo (i)	Operación (j)	Máquina (M)	Tiempo inicio	Tiempo proceso	Tiempo fin	Due date
6	1	2	0	20	20	165
6	2	3	20	22	42	165
6	3	9	42	5	47	165
6	4	12	47	18	65	165
5	1	1	0	20	20	121
5	2	6	20	24	44	121
5	3	9	47	5	52	121
5	4	12	65	18	83	121
1	1	2	20	20	40	159
1	2	5	40	26	66	159
1	3	10	66	5	71	159
1	4	12	83	18	101	159
2	1	1	20	22	42	143
2	2	6	44	21	65	143
2	3	9	52	5	57	143
2	4	12	101	18	119	143
7	1	1	42	26	68	94
7	2	8	68	42	110	94
7	3	11	110	7	117	94
7	4	12	119	21	140	94
4	1	2	40	23	63	120
4	2	8	110	31	141	120
4	3	9	141	5	146	120
4	4	12	146	18	164	120
3	1	2	63	21	84	147
3	2	8	141	42	183	147
3	3	9	183	5	188	147
3	4	13	188	21	209	147

Tiempos en minutos

*Tabla 6.6* Secuenciación con datos de la empresa (ejemplo 2)

Fuente: Elaboración propia

Trabajo (i)	Operación (j)	Máquina (M)	Tiempo inicio	Tiempo proceso	Tiempo fin	Due date
2	1	1	0	24	24	87
2	2	6	24	23	47	87
2	3	9	47	8	55	87
2	4	12	55	20	75	87
3	1	1	24	24	48	104
3	2	8	48	37	85	104
3	3	9	85	8	93	104

3	4	12	93	20	103	104
1	1	1	48	24	72	113
1	2	3	72	27	99	113
1	3	9	99	8	107	113
1	4	12	107	20	127	113

Tiempos en minutos

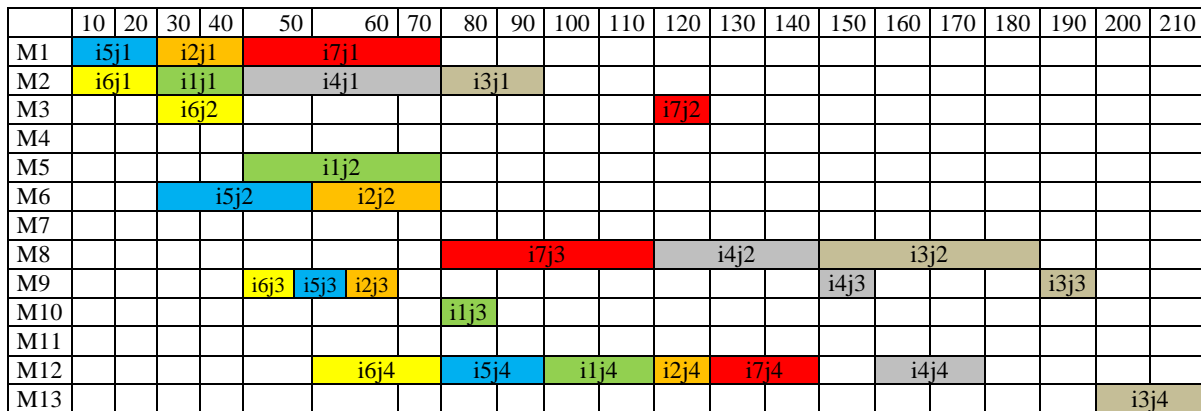


Figura 6.2 Diagrama de Gantt para el ejemplo 1 con datos de la empresa

Fuente: Elaboración propia

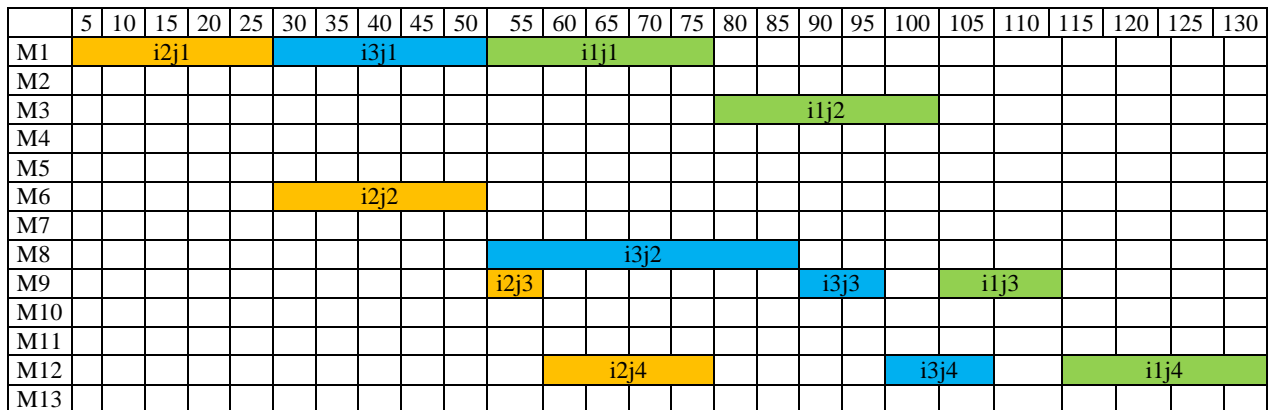


Figura 6.3 Diagrama de Gantt para el ejemplo 2 con datos de la empresa (B.T)

Fuente: Elaboración propia

## 7 Conclusiones y Recomendaciones

A través de la aplicación de la metaheurística propuesta se pudo observar la importancia de tener un buen planteamiento de la solución inicial, dado que se pudo constatar que con un buen diseño de experimentos sería factible encontrar un valor cercano al óptimo de la variable que se está estudiando (en este caso el número de trabajos tardíos), lo cual permite tener amplias posibilidades de encontrar incluso mejores soluciones al optimizar esta heurística, removiendo varios de los elementos aleatorios utilizados en este trabajo y realizando nuevas y más profundas investigaciones que permitan establecer con mayor certeza los valores óptimos de dicha variable a ser aplicadas en cualquier instancia utilizada y también con datos reales.

Se pudo también observar que esta metodología puede ofrecer resultados viables a la empresa en términos de poder brindar secuenciaciones que permitan minimizar la cantidad de trabajos tardíos bajo cualquier circunstancia en tiempos razonables que permitan a la empresa una toma de decisiones rápida según sea el caso.

A lo largo del desarrollo de este trabajo se han podido encontrar diversas formas de variar el enfoque de este trabajo, que como se dijo anteriormente, sirve como un punto de referencia para trabajar en base a este tipo de sistemas de producción asociándolos con la mejora en el desempeño de las variables asociadas con entregas tardías. Comenzando por la configuración del sistema de la empresa, ya que puede abordarse considerando que se trabaja bajo un entorno MTS o ATO, podría ser de interés observar y analizar el comportamiento del modelo realizado según los diferentes niveles de inventario de la empresa.

Por otra parte hay que observar que el modelo matemático fue basado en variables de secuencia-posición que demostró tener un alto consumo de recursos computacionales y debido a esto solo fue posible realizar pruebas con instancias muy pequeñas, las cuales tenían máximo 2 trabajos y 2 operaciones por trabajo.

Se puede optar por diseñar un modelo matemático basado en variables de precedencia o de tiempo indexado y comparar el desempeño entre los tres tipos de modelo

para validar cual tiene un mejor desempeño y brindar otro punto de referencia. De igual forma se podría considerar la opción de relajar restricciones para estos modelos en caso de que sea viable acudir a dicha medida.

Una de las ventajas al momento de realizar la metaheurística fue la realización de un experimento para determinar tanto los due dates como la solución inicial de la misma, sin embargo, sería interesante buscar otras formas más metódicas para determinar los due dates, ya que se vio que este es un factor que tiene una influencia muy importante en el desarrollo y los resultados del método de solución implementado; también se debería trabajar en otros métodos para determinar una solución inicial factible que bien puede salir con otro tipo de experimentos, bien sea aplicando otro tipo de reglas de despacho o creando una solución de forma arbitraria.

Como los métodos de solución utilizados sirven para establecer un límite superior de las variables analizadas, otro aspecto a investigar y analizar sería el comportamiento de dichas variables empleando otras metaheurísticas tales como algoritmos genéticos, algoritmos de recocido simulado, algoritmos evolucionarios, entre otros., y observar cómo cambian las soluciones, en qué instancias hay puntos de mejora, bien sea en cuanto a las variables observadas y los recursos computacionales utilizadas.

Y finalmente se espera que en el futuro se hagan pruebas usando datos reales de otros tipos de empresas de diferentes industrias y diferentes tamaños, para una mejor comprensión de su desempeño en sistemas de producción más complejos, ya que como se pudo observar, el uso de instancias o en datos de sistemas de producción más sencillos tiene sus limitaciones y sería más interesante analizar qué tipo de modelo se puede ajustar mejor a otras empresas que tengan necesidades similares a la estudiada en este caso.

## Referencias Bibliográficas

- Al-Hinai, N., & Elmekawy, T. Y. (2011). Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *International Journal of Production Economics*, *132*(2), 279–281.  
<http://doi.org/10.1016/j.ijpe.2011.04.020>
- Bagheri, a., Zandieh, M., Mahdavi, I., & Yazdani, M. (2010). An artificial immune algorithm for the flexible job-shop scheduling problem. *Future Generation Computer Systems*, *26*(4), 533–541. <http://doi.org/10.1016/j.future.2009.10.004>
- Behnke, D., & Geiger, M. (2012). Test Instances for the Flexible Job Shop Scheduling Problem with Work Centers. *Opus.Unibw-Hamburg.De*. Retrieved from <http://opus.unibw-hamburg.de/volltexte/2012/2982/>
- Ben Hmida, A., Haouari, M., Huguet, M.-J., & Lopez, P. (2010). Discrepancy search for the flexible job shop scheduling problem. *Computers & Operations Research*, *37*(12), 2192–2201. <http://doi.org/10.1016/j.cor.2010.03.009>
- Bożejko, W., Uchroński, M., & Wodecki, M. (2010). Parallel hybrid metaheuristics for the flexible job shop problem. *Computers & Industrial Engineering*, *59*(2), 323–333.  
<http://doi.org/10.1016/j.cie.2010.05.004>
- Britto, R., Mejía, G., & Caballero, J. P. (2007). Programación de la producción en sistemas de manufactura tipo taller con el algoritmo combinado cuello de botella móvil y búsqueda tabú. *Production Programming in Manufacturing Systems and Taboo Search Algorithm*, *11*(2), 203–224.
- Chen, J. C., Wu, C.-C., Chen, C.-W., & Chen, K.-H. (2012). Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm. *Expert Systems with Applications*, *39*(11), 10016–10021.  
<http://doi.org/10.1016/j.eswa.2012.01.211>
- Dell'Amico, M., & Trubian, M. (1993). Applying tabu search to the job-shop scheduling problem. *Annals of Operations Research*, *41*, 231–252.  
<http://doi.org/10.1007/BF02023076>
- Demir, Y., & Kürşat İşleyen, S. (2013). Evaluation of mathematical models for flexible job-shop scheduling problems. *Applied Mathematical Modelling*, *37*(3), 977–988.  
<http://doi.org/10.1016/j.apm.2012.03.020>

- Fattahi, P., & Fallahi, A. (2010). Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability. *CIRP Journal of Manufacturing Science and Technology*, 2(2), 114–123. <http://doi.org/10.1016/j.cirpj.2009.10.001>
- Fattahi, P., Jolai, F., & Arkat, J. (2009). Flexible job shop scheduling with overlapping in operations. *Applied Mathematical Modelling*, 33(7), 3076–3087. <http://doi.org/10.1016/j.apm.2008.10.029>
- Fattahi, P., Saidi Mehrabad, M., & Jolai, F. (2007). Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. *Journal of Intelligent Manufacturing*, 18(3), 331–342. <http://doi.org/10.1007/s10845-007-0026-8>
- Gao, J., Sun, L., & Gen, M. (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Computers & Operations Research*, 35(9), 2892–2907. <http://doi.org/10.1016/j.cor.2007.01.001>
- Ho, N. B., Tay, J. C., & Lai, E. M. K. (2007). An effective architecture for learning and evolving flexible job-shop schedules. *European Journal of Operational Research*, 179(2), 316–333. <http://doi.org/10.1016/j.ejor.2006.04.007>
- Hurink, J., Jurisch, B., & Thole, M. (1994). Tabu Search for the Job-Shop Scheduling Problem with Multi-Purpose Machines. *OR Spektrum*, 15(4):205–215.
- Kacem, I., Hammadi, S., & Borne, P. (2002). Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic. *Mathematics and Computers in Simulation*, 60(3-5), 245–276. [http://doi.org/10.1016/S0378-4754\(02\)00019-8](http://doi.org/10.1016/S0378-4754(02)00019-8)
- Karimi, H., Rahmati, S. H. a, & Zandieh, M. (2012). An efficient knowledge-based algorithm for the flexible job shop scheduling problem. *Knowledge-Based Systems*, 36, 236–244. <http://doi.org/10.1016/j.knosys.2012.04.001>
- Lee, J. Y., & Kim, Y. D. (2012). Minimizing the number of tardy jobs in a single-machine scheduling problem with periodic maintenance. *Computers and Operations Research*, 39(9), 2196–2205. <http://doi.org/10.1016/j.cor.2011.11.002>
- Lee, Y. H., Jeong, C. S., & Moon, C. (2002). Advanced planning and scheduling with outsourcing in manufacturing supply chain. *Computers & Industrial Engineering*, 43(1-2), 351–374. [http://doi.org/10.1016/S0360-8352\(02\)00079-7](http://doi.org/10.1016/S0360-8352(02)00079-7)

- Li, J. Q., & Pan, Q. K. (2013). Chemical-reaction optimization for solving fuzzy job-shop scheduling problem with flexible maintenance activities. *International Journal of Production Economics*, 145(1), 4–17. <http://doi.org/10.1016/j.ijpe.2012.11.005>
- Li, Z., Chen, Q., Mao, N., Wang, X., & Liu, J. (2013). Int . J . Production Economics Scheduling rules for two-stage fl exible fl ow shop scheduling problem subject to tail group constraint. *Intern. Journal of Production Economics*, 146(2), 667–678. <http://doi.org/10.1016/j.ijpe.2013.08.020>
- Lin, L., Gen, M., Liang, Y., & Ohno, K. (2012). A hybrid EA for reactive flexible job-shop scheduling. *Procedia Computer Science*, 12, 110–115. <http://doi.org/10.1016/j.procs.2012.09.039>
- Mastrolilli, M., & Gambardella, L. M. (1999). *Effective Neighborhood Functions for the Flexible Job Shop Problem*. Lugano, Switzerland. Retrieved from <http://www.idsia.ch>
- Moslehi, G., & Mahnam, M. (2011). A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *International Journal of Production Economics*, 129(1), 14–22. <http://doi.org/10.1016/j.ijpe.2010.08.004>
- Nakandhrakumar, R. S., Seralathan, S., Azarudeen, A., & Narendran, V. (2014). Optimization of Job Shop Scheduling Problem using Tabu Search Optimization Technique, 3(3), 1241–1244.
- Nowicki, E., & Smutnicki, C. (1996). A fast Taboo Search Algorithm for the Job Shop Problem. *Management Science*, 42(6), 797–813.
- Osorio, C. J., Castrillon, O., Toro, J., & Orejuela, J. P. (2008). Modelo de programación jerárquica de la producción en un Job shop flexible con interrupciones y tiempos de alistamiento dependientes de la secuencia. *Revista Ingeniería E Investigación*, 28(2), 72–79.
- Özgülven, C., Yavuz, Y., & Özbakir, L. (2012). Mixed integer goal programming models for the flexible job-shop scheduling problems with separable and non-separable sequence dependent setup times. *Applied Mathematical Modelling*, 36(2), 846–858. <http://doi.org/10.1016/j.apm.2011.07.037>
- Pezzella, F., Morganti, G., & Ciaschetti, G. (2008). A genetic algorithm for the Flexible Job-shop Scheduling Problem. *Computers and Operations Research*, 35(10), 3202–3212. <http://doi.org/10.1016/j.cor.2007.02.014>

- Riojas Cañari, A. C. (2005). Conceptos, algoritmo y aplicación al problema de las N-Reinas. Universidad Nacional Mayor de San Carlos.
- Scrich, C. R., Armentano, V. a., & Laguna, M. (2004). Tardiness minimization in a flexible job shop: a tabu search approach. *Journal of Intelligent Manufacturing*, *15*(1), 103.
- Tanev, I. T., Uozumi, T., & Morotome, Y. (2004). Hybrid evolutionary algorithm-based real-world flexible job shop scheduling problem: application service provider approach. *Applied Soft Computing*, *5*(1), 87–100.  
<http://doi.org/10.1016/j.asoc.2004.03.013>
- Tang, J., Zhang, G., Lin, B., & Zhang, B. (2011). A hybrid algorithm for flexible job-shop scheduling problem. *Procedia Engineering*, *15*, 3678–3683.  
<http://doi.org/10.1016/j.proeng.2011.08.689>
- Teekeng, W., & Thammano, A. (2012). Modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems. *Procedia Computer Science*, *12*, 122–128.  
<http://doi.org/10.1016/j.procs.2012.09.041>
- Thammano, A., & Phu-Ang, A. (2013). A hybrid artificial bee colony algorithm with local search for flexible job-shop scheduling problem. *Procedia Computer Science*, *20*, 96–101. <http://doi.org/10.1016/j.procs.2013.09.245>
- Thörnblad, K., Strömberg, A.-B., Patriksson, M., & Almgren, T. (2013). Scheduling optimization of a real flexible job shop including side constraints regarding maintenance fixtures and night shifts, 1–32.
- Thörnblad, K., Strömberg, A.-B., Patriksson, M., & Almgren, T. (2011). *An efficient algorithm for solving the flexible job shop scheduling problem*. Göteborg, Sweden.
- Thörnblad, K. (2011). *On the Optimization of Schedules of a Multitask Production Cell*. Department of Mathematical Sciences, Chalmers University of Technology  
Department of Mathematical Sciences, University of Gothenburg.
- Wang, S., & Yu, J. (2010). An effective heuristic for flexible job-shop scheduling problem with maintenance activities. *Computers and Industrial Engineering*, *59*(3), 436–447. <http://doi.org/10.1016/j.cie.2010.05.016>
- Xing, L.-N., Chen, Y.-W., Wang, P., Zhao, Q.-S., & Xiong, J. (2010). A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems. *Applied Soft Computing*, *10*(3), 888–896. <http://doi.org/10.1016/j.asoc.2009.10.006>



- Yuan, Y., & Xu, H. (2013a). An integrated search heuristic for large-scale flexible job shop scheduling problems. *Computers & Operations Research*, *40*(12), 2864–2877. <http://doi.org/10.1016/j.cor.2013.06.010>
- Yuan, Y., & Xu, H. (2013b). Flexible job shop scheduling using hybrid differential evolution algorithms. *Computers & Industrial Engineering*, *65*(2), 246–260. <http://doi.org/10.1016/j.cie.2013.02.022>
- Yuan, Y., Xu, H., & Yang, J. (2013). A hybrid harmony search algorithm for the flexible job shop scheduling problem. *Applied Soft Computing*, *13*(7), 3259–3272. <http://doi.org/10.1016/j.asoc.2013.02.013>
- Zhang, Q., Manier, H., & Manier, M. a. (2012). A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times. *Computers and Operations Research*, *39*(7), 1713–1723. <http://doi.org/10.1016/j.cor.2011.10.007>
- Zribi, N., Kacem, I., & Borne, P. (n.d.). Minimizing the total tardiness in a flexible job-shop, 1–18.

## Abreviaturas

MTO	Make to Order (Producción bajo pedido)
MTS	Make to Stock (Producción bajo inventario)
ATO	Assemble to Order (Ensamblaje bajo pedido)
FJSP	Flexible Job Shop Problem
JSP	Job Shop Problem
FIFO	First In, First Out (Primero que entra, primero que sale)
CR	Critical Ratio (Radio Crítico)
MS	Minimum Slack (Holgura Mínima)
EDD	Earliest Due Date (Fecha de entrega más cercana)
Cmax	Makespan (tiempo para completar todos los trabajos)