

**DISEÑO DE UNA HEURÍSTICA EFICIENTE BASADA EN TEORÍA DE  
RESTRICCIONES PARA LA PROGRAMACIÓN DE SISTEMAS JOB SHOP**

**MALKA GARCIA PEREIRA**

**UNIVERSIDAD LIBRE  
INGENIERIA INDUSTRIAL  
BARRANQUILLA  
2017**

**DISEÑO DE UNA HEURÍSTICA EFICIENTE BASADA EN TEORÍA DE  
RESTRICCIONES PARA LA PROGRAMACIÓN DE SISTEMAS JOB SHOP**

**MALKA GARCIA PEREIRA**

**Anteproyecto de trabajo de grado presentado como requisito para optar al  
título de ingeniero industrial**

**Asesor: Ronald David Suárez Díaz, ingeniero industrial  
Magister en ingeniería industrial**

**UNIVERSIDAD LIBRE  
INGENIERIA INDUSTRIAL  
BARRANQUILLA**

**2017**

## TABLA DE CONTENIDO

<b>INTRODUCCIÓN</b> .....	5
<b>1. PLANTEAMIENTO DEL PROBLEMA</b> .....	8
1.1 Descripción del Problema .....	8
1.2 Formulación del Problema .....	9
<b>2. OBJETIVOS</b> .....	11
2.1. Objetivo General .....	11
2.2. Objetivos Específicos .....	11
<b>3. JUSTIFICACIÓN</b> .....	12
<b>4. ALCANCE Y LIMITACIONES</b> .....	13
4.1. Alcance.....	13
4.2. Limitaciones.....	13
<b>5. MARCO DE REFERENCIA</b> .....	14
5.1. Estado del Arte .....	14
5.2. Marco teórico .....	15
<b>6. MARCO METODOLÓGICO</b> .....	18
6.1. Tipo de Investigación.....	18
6.2. Diseño de Investigación.....	18
6.3 Método para el desarrollo de actividades .....	19
6.4. Población y muestra .....	20
6.4.1. La Población.....	20
6.4.2. La Muestra .....	20
6.5. Obtención de la Información .....	21

6.6. Técnicas de Recolección de Datos.....	21
6.7. Análisis de la Información.....	21
<b>7. PERSONAS QUE INTERVIENEN EN EL PROYECTO .....</b>	<b>21</b>
<b>8. ASPECTOS ADMINISTRATIVOS .....</b>	<b>22</b>
8.1. Recursos Disponibles.....	22
<b>9. CRONOGRAMA .....</b>	<b>22</b>
<b>10. EJECUCION DEL PROYECTO .....</b>	<b>23</b>
10.1. Definición de Parámetros de entra y variables de interés.....	23
10.2. Desarrollo de la Heurística .....	24
10.3. Generación de instancias aleatorias .....	25
<b>11. RESULTADOS Y CONCLUSIONES.....</b>	<b>28</b>
<b>12. RECOMENDACIONES Y FUTURAS INVESTIGACIONES.....</b>	<b>32</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>33</b>

## INTRODUCCIÓN

Los procesos y procedimientos en las industrias requieren de tres elementos básicos para una gestión adecuada: la planeación, la programación y el control de la producción. En particular, el tema de la programación de la producción es uno de los elementos neurálgicos en diversas compañías, por lo que se constituye en una respuesta operativa para optimizar la producción de un bien o servicio.

Siguiendo un ordenamiento lógico, la programación de la producción debe ser un paso posterior a la planeación, ya que consiste en la fijación de planes y horarios de la producción, de acuerdo a la prioridad de la operación por realizar, determinado así su inicio y fin para lograr el nivel más eficiente. La función principal de la programación de la producción es lograr un movimiento uniforme y rítmico de los productos a través de las etapas de producción. En otras palabras, con la programación se determina cuándo se debe iniciar y terminar cada lote de producción, qué operaciones se van a utilizar, con qué máquina y con qué operarios.

La programación es una de las actividades más relevantes y complejas dentro de la gestión de la producción. Un buen programa de producción trae ventajas para la empresa, como las mencionadas a continuación [1]:

- Los pedidos se pueden entregar en las fechas estipuladas
- Se calculan las necesidades de mano de obra, maquinaria y equipo, así habrá una mejor utilización de estos recursos
- Disminuir los costos de fabricación.

Esto significa que mediante la programación se puede predecir donde podrían estar las pérdidas de tiempo o las sobrecargas entre los centros de producción o

procesos. Así mismo permite el control diario y periódico de las tareas y la correcta toma de decisiones. Todo ello para cumplir con los plazos de entrega establecidos y tener un óptimo uso de los recursos.

Sobre la planificación de la producción, los autores Urwick y Brech introducen referencias que indican que toda persona familiarizada con la industria debe apreciar la enorme ventaja que se obtiene de un análisis de las necesidades, si es posible, para todo el año. Señalan que cuando uno conoce, a intervalos regulares de tiempo, la cantidad de producción que es demandada, por lo tanto, a procesar, es posible desarrollar programas para la manufactura, que permitan a la compañía alcanzar un pleno rendimiento; que la producción se abarate enormemente; que no sea necesario mantener grandes stocks en tiempos excesivos; que el capital y el interés economicen, en general que aumente la eficacia [1]

Es necesario tener en cuenta que la programación es uno de los problemas complejos de resolver debido al alto número de posibilidades que pueden ser consideradas en la secuenciación de pedidos; justamente por ello es que hay que tener en cuenta muchos factores y considerar ciertos objetivos. Precisamente por este motivo no es posible tenerlos en cuenta a la vez y, por lo tanto, se debe establecer una metodología racional si se desean obtener resultados útiles. Por eso, se consideran diferentes niveles de planificación relacionados entre sí, con distintos niveles de detalle, y plazos de tiempo para llevarlos a cabo.

De ahí que la secuenciación de operaciones hace parte del nivel de planificación a corto plazo y principalmente, es una área de investigación desde hace más de cinco décadas, es una rama de la optimización combinatoria que ha desarrollado su propia metodología utilizando herramientas matemáticas y computacionales variadas; donde el objetivo es encontrar una secuencia óptima para la ejecución de unos trabajos y sus respectivas operaciones en las máquinas o recursos

disponibles limitados, la optimalidad de la secuencia depende de la función objetivo. Es un proceso de toma de decisiones que constituye uno de los problemas más importantes en gestión de la producción, tanto desde el punto de vista teórico como práctico [2]

El objetivo de este documento es ofrecer una visión de este problema y las variables que influyen para la realización de la producción en empresas de ambientes Job Shop, teniendo en cuenta sus inicios hasta la actualidad, destacando las referencias y las propuestas de los principales autores y comparando las herramientas más influyentes.

# 1. PLANTEAMIENTO DEL PROBLEMA

## 1.1. DESCRIPCION DEL PROBLEMA

Los sistemas productivos Job Shop se basan en la demanda, debido a que las ordenes y características de los clientes varían, haciendo que los productos tengan diferentes recorridos y secuencias a través de los procesos y máquinas, organizados por funciones. Por lo general dentro de estos sistemas la fabricación se realiza en pequeños lotes pero con variedad de productos que corresponden a solicitudes particulares de cada pedido.

Otra de las características más importante del sistema Job Shop es que este funciona bajo la priorización de órdenes de producción que se generen, en el cual se tiene en cuenta la fecha de entrega de los pedidos, los tiempos y el costo de alistar cada orden; por lo tanto, es importante realizar una adecuada programación de la secuencia que tendrán cada tarea en el proceso y dar cumplimiento a estas.

Las empresas que trabajan bajo este tipo de sistemas asumen que la programación es de carácter complejo, debido a que la secuencia en el proceso no es estable.

Al considerar cada pedido en una situación bajo el ambiente de fabricación Job Shop, surge la posibilidad de encontrar una serie de condiciones en común entre los proyectos, ya que al imaginar una empresa que maneja pedidos únicos teniendo en cuenta las determinaciones sugeridas por los clientes, puede encontrarse en una situación de desarrollo, al mismo tiempo de varios proyectos, haciendo uso de recursos como: máquinas, mano de obra, materias primas, tiempo, etc. recursos que son utilizados en los diferentes proyectos que lleva a cabo la empresa, teniendo en cuenta lo que menciona Márkus [3] al hablar de la posibilidad de realizar aplicaciones de modelos propios de la programación



multi-proyectos como alternativa de solución al problema de la programación de la producción en ambientes Job Shop.

La programación del Job shop ha sido un tema de investigación por mucho tiempo donde se ha desarrollado varias técnicas para la programación de la producción por procesos pero la aplicación de estas han sido muy complejas e incluso costosas al momento de adquirirlas, es decir, solo grandes empresas pueden conseguirlas gracias a sus altos ingresos, pero medianas o pequeñas empresas que tengan un sistema de producción como este solo pueden programarlo empíricamente debido a que sus ingresos no dan para adquirir una herramienta como esta. Siendo más específicos, por un lado, dada la alta complejidad de los algoritmos usados para resolver los problemas de tipo Job Shop Scheduling, estos algoritmos solo suelen estar en programas muy costosos, por lo que a las pequeñas y medianas empresas se les dificulta su adquisición. Por otro lado, a las empresas les resulta impráctico emplear métodos tan sofisticados desde el punto de vista matemático, por lo que optan por usar métodos muy simples de programación de la producción, que suelen resultar muy ineficientes. Este tipo de situaciones se presentan con mayor regularidad en Pymes, donde suelen usarse métodos muy empíricos para la resolución de problemas. Es así como resulta atractivo y conveniente desarrollar métodos que resulten más fáciles de estudiar, analizar e implementar, que a la vez generen una programación eficiente de la producción, con aras de mejorar la productividad dentro de la planta, incrementando el nivel de servicio ofertado a los clientes.

## 1.2. FORMULACION DEL PROBLEMA

Se pretende diseñar una heurística que solucione los problemas que se presentan en la programación de tareas de ambientes Job Shop con cuatro máquinas y compararla con otras metodologías, partiendo del modelo que se quiere realizar, se plantea el siguiente interrogante:

¿Resulta eficiente implementar una heurística basada en teoría de restricciones para la programación de la producción en sistemas Job Shop?

¿Cuál es el nivel de eficiencia de la heurística propuesta?

¿Cómo pueden beneficiarse las Pymes de ambientes de producción Jop Shop con heurística propuesta en este proyecto?

## **2. OBJETIVOS**

### **2.1. OBJETIVO GENERAL**

Diseñar una heurística eficiente basada en la teoría de restricciones para la programación de sistemas productivos Job Shop.

### **2.2. OBJETIVOS ESPECÍFICOS**

- Definir variables y parámetros más importantes aplicables en sistemas productivos Job shop.
- Construir la heurística basada en teoría de restricciones que dé solución eficiente a los problemas Job shop.
- Desarrollar una plantilla en Microsoft Excel donde se pueda resolver cualquier instancia de problemas Job Shop usando la heurística propuesta.

### 3. JUSTIFICACIÓN

Para la correcta disposición de los recursos es necesario una buena administración y por ello para dar solución a la problemática relacionada con la asignación y programación de operaciones en ambientes multi-proyecto, se han creado y utilizado modelos RCPSP, que se encargan de solucionar de manera efectiva el problema de la producción en ambientes de manufactura Job Shop, ya que esta presenta limitaciones que se relacionan con la repartición de recursos.

Es necesario entender que el departamento encargado de los multi-proyectos debe estar atento a la situación en la que diversos proyectos apuestan por los mismos recursos con el fin de sustituir una necesidad satisfactoriamente.

Para Vall [4], esto plantea la posibilidad de seguir estudiando la programación de las operaciones en los sistemas Job Shop bajo los parámetros de este tipo de modelos. Teniendo en cuenta lo anterior las empresas han intentado superar esta problemática, utilizando una combinación de herramientas heurísticas que aunque no han brindado una solución completa, si han conseguido brindar una solución alternativa eficiente y lo más importante, en un tiempo computacional aceptable.

## **4. ALCANCE Y LIMITACIONES**

### **4.1. ALCANCE**

- Este proyecto solo abarca un desarrollo teórico, no implica una aplicación en un contexto real. Para medir la eficiencia esperada de la heurística, sus resultados se compararán en instancias donde se conozca la solución óptima mediante el uso de modelación matemática.
- Solo se considerara una función objetivo, en la cual se tiene en cuenta el tiempo de terminación de todos los trabajos pendientes por programar. Dado que la capacidad de las Pymes suele ser bastante limitada, lo ideal para este tipo de entornos es tratar de maximizar la productividad. Desde el punto de vista de la programación de la producción, esto se logra terminando todos los trabajos en el menor tiempo posible.

### **4.2. LIMITACIONES**

- El proyecto se limita espacialmente a la universidad libre sede centro donde se desarrollará la aplicación del modelo computacional.
- La metodología propuesta no garantizara soluciones optimas, sin embargo, se espera brindar soluciones eficientes.

## 5. MARCO DE REFERENCIA

### 5.1. ESTADO DEL ARTE

El problema del Job Shop Scheduling (JSS de ahora en adelante) ha sido ampliamente estudiado en la literatura debido a sus múltiples aplicaciones en sistemas de la vida real. En [1] se estudian los métodos de enumeración implícita para la minimización del tiempo total de terminación. Se propone una nueva cota inferior más ajustada a las encontradas en la literatura, siendo comparadas en instancias computacionales. La teoría de restricciones ya ha sido utilizada de cierto modo para tratar este tipo de problemas. En [2] se usa el concepto de cuellos de botella para tratar de minimizar el tiempo de terminación de todos los trabajos. Se examina cada máquina sin programar, buscando quien es el cuello de botella en cada instante, seleccionando así la más lenta para reprogramar los trabajos. Los resultados computacionales mostraron que la metodología es considerablemente superior a varios métodos encontrados en la literatura. De manera similar los autores de [3] presentan una modificación del trabajo anterior, mejorando sustancialmente la eficiencia en las soluciones obtenidas.

Los métodos exactos también han sido ampliamente usados. Muestra de ello es el trabajo propuesto en [4], en donde se propone un método del tipo Branch and Bound. Para mejorar la eficiencia, se usan técnicas de poda que permitan reducir el tiempo de ejecución del algoritmo. Brucker [5] también propone un método del tipo Branch and Bound, usándolo en particular para resolver el problema del punto de referencia  $10 \times 10$ . Los resultados computacionales muestran que el tiempo de cómputo es significativamente menor a muchos algoritmos exactos presentes en la literatura.

Debido a la naturaleza NP-completa del problema, los procedimientos metaheurísticos han sido muy utilizados para resolver el problema del Job Shop Scheduling. Entre estos procedimientos pueden encontrarse: Algoritmos genéticos, recocido simulado, búsqueda tabú, entre otros. Autores como [6], [7], [8], [9] y [10] proponen diferentes variantes de algoritmos genéticos. Se

diferencian en la forma de codificar las soluciones, tipo de cruce, selección de individuos y forma de mutación. Para todos los casos mencionados, los resultados computacionales mostraron amplio rendimiento de los algoritmos, en términos de la calidad de las respuestas y tiempos de cómputo.

En [11] se muestra un estudio computacional de los problemas Job Shop. Los autores hacen una mezcla de diferentes procedimientos tales como planos cortantes y métodos de bifurcación y acotamiento. Trabajos como [12] usan búsqueda tabú para minimizar el makespan. Usando nuevas propiedades teóricas en el estudio de entornos, recalculan las rutas de programación para obtener soluciones más eficientes. Respecto a las rutas de programación mencionadas, [13] presenta un algoritmo basado en este aspecto para resolver el JJS. Otros trabajos de interés se presentan en [14], [15], [16], [17], [18]

## **5.2. MARCO TEÓRICO**

### **- Recocido Simulado**

El recocido simulado se basa en una analogía con el proceso físico de templado, en el cual se forma una estructura reticular en un sólido calentándolo en un baño caliente hasta ablandarlo y luego enfriándolo pausadamente hasta solidificarse en un estado de baja energía.

Desde el punto de vista de la optimización combinatoria, el simulated annealing es un algoritmo aleatorio de búsqueda local. Las soluciones vecinas de mejor costo son siempre aceptadas e incluso se aceptan las soluciones de peor costo, aunque con una probabilidad que decrece gradualmente durante el transcurso de la ejecución del algoritmo. Esta reducción de la tolerancia o probabilidad de aceptación es controlada por un conjunto de parámetros cuyos valores son determinados por un esquema de enfriamiento prescrito.

El simulated annealing ha sido vastamente aplicado con considerable éxito. Su naturaleza aleatoria permite convergencia asintótica a la solución óptima bajo condiciones moderadas. Desafortunadamente, la convergencia requiere

típicamente de tiempo exponencial, convirtiendo el simulate annealing en impráctico como instrumento para procurarse soluciones óptimas. En cambio, como muchos algoritmos de búsqueda local, resulta eficiente como método de aproximación donde presenta una tasa de convergencia más indulgente.

#### - **Taboo search**

La búsqueda tabú combina un algoritmo determinístico de avance iterativo con la posibilidad de aceptar soluciones que incrementen el costo. De esta manera, la búsqueda es dirigida fuera de óptimos locales y otras partes del espacio de soluciones pueden ser exploradas. Cuando es visitada una nueva solución, es considerada un vecino legítimo de la solución actual aun cuando empeore el costo.

El conjunto de vecinos legítimos queda representado por una lista tabú, cuyo objetivo es restringir la elección de soluciones para prevenir el regreso a puntos recientemente visitados. La lista tabú es actualizada dinámicamente durante la ejecución del algoritmo y define soluciones que no son aceptables en unas pocas siguientes iteraciones. Sin embargo, una solución de la lista tabú puede ser aceptada si su calidad, en algún sentido, elevada lo justifica. En este caso diremos que se ha alcanzado cierto nivel de aspiración (aspiration level).

El tabú search ha sido aplicado en una gran variedad de problemas con considerable éxito ya que presenta un esquema de gran adaptabilidad que se ajusta a los detalles del problema considerado. Por otro lado, existen vagos conocimientos teóricos que guíen ese proceso de ajuste y cada usuario debe recurrir a la información práctica disponible y a su propia experiencia.

#### - **Configuración productiva Job Shop.**

Las distribuciones productivas se distinguen entre sí asumiendo la continuidad en la cual producen, conforme al sistema de manejo de inventarios, la variedad de los



productos y al volumen de producción. Estos rasgos delimitarán la distribución productiva de la empresa.

Hay que mencionar además que existe una variación del Job Shop es el conocido Job Shop flexible (flexible Job Shop), el cual supone la posibilidad de tener máquinas en paralelo, lo cual añade complejidad a la programación, por lo que se debe tener en cuenta un elemento adicional a la hora de proyectar, que es la atribución de los productos a las máquinas utilizables dentro de un centro de trabajo, que es diferente de la secuenciación.

- **Producción bajo pedido (MTO).**

Conocida en inglés como Make To Order, el cual parte de los requerimientos reales de los clientes (pedidos) y, con base en ellos, se programan las labores de producción.

Problema de la programación de proyectos con restricciones de recursos (RSPCP), para la correcta disposición de los recursos es necesario una buena administración y por ello para dar solución a la problemática relacionada con la asignación y programación de operaciones en ambientes multi-proyecto, se han creado y utilizado modelos RCPSP, que se encargan de solucionar de manera efectiva el problema de la producción en ambientes de manufactura Job Shop-MTO, ya que esta presenta limitaciones que se relacionan con la repartición de recursos.

- **Múltiples rutas.**

Para la producción de un producto es necesario tener diferentes múltiples rutas, siempre y cuando se quiera ser competitivo, esto consiste en la concesión de cada trabajo a una máquina perteneciente a un grupo de máquinas disponibles, con capacidad y características idénticas.

## **6. MARCO METODOLÓGICO**

### **6.1. TIPO DE INVESTIGACIÓN**

El presente proyecto está establecido como numero - explicativo, debido a que pretende identificar las variables más relevantes presentadas en la programación productiva bajo Job Shop, así mismo describir los efectos que estos tienen en la secuencia de tareas de los procesos, con el fin de implementar una heurística basada en la programación de este tipo de sistemas aplicado en tres máquinas.

### **6.2. DISEÑO DE INVESTIGACIÓN**

Para esta investigación es necesario partir de la revisión bibliográfica efectuada para lograr la identificación de las características propias de ambientes de proyecto y ambientes Job Shop de producción. En estas indagaciones se exhiben los elementos en común entre los tipos de ambientes, permitiendo realizar un paralelo con ellos. Seguido a lo anterior, se encontraron todos aquellos elementos propios de la resolución de problemas relacionados con ambientes de proyecto y ambientes Job Shop de producción.

Por otro lado, una vez identificadas las variables necesarias para cumplir con los objetivos de este proyecto, se planteó un modelo basado en la teoría de las restricciones cuyo objetivo es la programación relacionada con ambientes de proyecto y ambiente Job Shop de producción, considerando las restricciones de recursos y máquinas en paralelo (múltiples rutas).

Por último, para lograr la validación del modelo es necesaria la aplicación de instancias generada, por medio del cual se examinaría los efectos del modelo respecto a las diferentes variables a tener en cuenta, lo que permitiría concluir acerca de la eficacia y eficiencia del modelo planteado.

### 6.3. MÉTODO PARA EL DESARROLLO DE ACTIVIDADES

Para dar resultado a los objetivos específicos de esta investigación, es necesaria la ejecución de las siguientes actividades:

<b>Objetivos</b>	<b>Desarrollo de Actividades</b>
<b>Objetivo específico 1</b>	<p>Identificación los modelos de programación de producción en ambientes Job Shop más relevantes en la literatura. Con esto se busca extraer los modelos referentes más importantes que sirven como referencia actualmente</p> <p>Definición de parámetros más importantes en ambientes Job Shop. Basándose en los modelos previamente consultados, se extraerán los parámetros más importantes en dichos modelos. Estos parámetros contienen información tal como tiempos de procesamiento, números de estaciones, rutas de procesamiento, entre otros.</p> <p>Definición de las variables más importantes en los modelos de programación de la producción en ambientes Job Shop. Al igual que se hizo con los parámetros, se identificarán las variables más importantes en los modelos de programación Job Shop. Estas variables pueden ser orden de secuenciación, Makespan, etc.</p>
<b>Objetivo específico 2</b>	<p>Identificación de los recursos restrictivos en un sistema de producción Job Shop. Mediante esto, podrán emplearse los conceptos de teoría de restricciones para explotar al máximo las propiedades del sistema.</p>

	<p>Definición de la función objetivo. Debe definirse la función objetivo de interés (Makespan, tardanza total, etc.) con el propósito de saber cómo guiar el algoritmo a desarrollar.</p> <p>Diseño de la heurística. Una vez definida la función de interés, usando teoría de restricciones, se propondrá un procedimiento heurístico que permita obtener soluciones eficientes al problema propuesto.</p>
<p><b>objetivo específico 3</b></p>	<p>Proposición de instancias aleatorias. Se propondrán instancias aleatorias para resolverlas computacionalmente con la heurística propuesta.</p> <p>Comparación con otros procedimientos. Las instancias propuestas también serán resueltas con otros procedimientos propuestos en la literatura, con el objetivo de medir su eficiencia computacional.</p>

## 6.4. POBLACION Y MUESTRA

### 6.4.1. La población

Estará dada por todos los sistemas donde la configuración de producción es de tipo Job Shop.

### 6.4.2. La muestra

Debido a que es un trabajo experimental la muestra será de instancias aleatoria que se generen computacionalmente,

## **6.5. OBTENCIÓN DE LA INFORMACION**

Para la obtener la información concerniente a tiempos de procesamientos y número de trabajos se generaran instancias aleatorias, mediante distribuciones estadísticas de probabilidad.

## **6.6. TÉCNICA DE RECOLECCION DE DATOS**

No se aplicara una técnica para la recolección de información debido a que los datos que serán utilizados en este proyecto serán generados aleatoriamente de manera computacional basado en un diseño experimental.

## **6.7. ANÁLISIS DE INFORMACIÓN**

Al obtener la información será necesaria la utilización de las siguientes herramientas:

- Hoja de cálculo de Excel para el desarrollo de instancias aleatoria y el análisis experimental de los resultados obtenidos.
- Visual Basic de Excel para la ejecución de la heurística del proyecto.

## **7. PERSONAS QUE INTERVIENEN EN EL PROYECTO**

Para la realización de esta investigación se contó con un proponente principal, estudiante del programa de Ingeniería Industrial de la Universidad Libre, Seccional Barranquilla

- Malka Irina García Pereira

### Asesor del Proyecto:

- Ing. Ronald David Suarez Díaz, Tutor del proyecto.

## 8. ASPECTOS ADMINISTRATIVOS

### 8.1. Recursos disponibles

- **Recursos Humanos:** Ing. Ronald David Suarez Díaz
- **Recursos Financieros:** Aportes propios de los investigadores
- **Recursos Institucionales:** Biblioteca de la Universidad

## 9. CRONOGRAMA

El siguiente cronograma presenta las actividades a desarrollar para el cumplimiento de los objetivos establecidos en este proyecto, el cual ilustra desde la presentación del anteproyecto hasta la sustentación final.

ACTIVIDADES	Meses				
	Mayo	Junio	Julio	Agosto	Septiembre
Presentación de anteproyecto					
Definición de parámetro y variables más importantes en Ambientes Job shop					
Identificación de recursos restrictivos en un sistemas Job Shop					
Definición de la función Objetivo					

Diseño de la Heurística					
Proposición de instancias Aleatorias					
Comparación con otros procedimientos					
Resultados y conclusión de la investigación					
Presentación del proyecto					
Sustentación del proyecto					

## 10. EJECUCION DE PROYECTO

### 10.1. Definición de Parámetros de entrada y variables de interés

Los parámetros de entrada a tener en cuenta en problemas de programación Job Shop son:

*N*: Conjunto de trabajos

*M*: Conjunto de Máquina

$p_{ij}$ : Tiempo de procesamiento del trabajo *i* en la máquina *j*

$S_i$ : Secuencia de máquinas para el trabajo *i*

Por otro lado, las variables de interés del problema son:

$C_j$ : Tiempo de finalización de tareas para la máquina *j*

$C_{max} = \max\{C_j\}$ : Máximo tiempo de terminación

$U_j$ : Orden de procesamiento de trabajos en la máquina *j*

$T_i$ : Tiempo en el que el trabajo *i* es finalizado

La variable que se mide en la función objetivo corresponde al  $C_{max}$ , el cual corresponde al instante de tiempo en el que todos y cada uno de los trabajos son terminados. Este objetivo conlleva a hacer eficiente el uso de la capacidad instalada dentro de la planta, pues se pretendería terminar todas las tareas en el menor tiempo posible.

Los problemas Job Shop son complejos ya que las tareas o proyectos pueden presentar secuencias de procesamiento diferentes. Partiendo del principio de recurso restrictivo promulgado por la teoría de restricciones, se buscará balancear el uso de las máquinas más críticas, de manera que dichos recursos se mantengan ocupados la mayor parte del tiempo dentro de la programación establecida.

## **10.2. Desarrollo de la heurística**

La heurística tiene como fuente clave el uso de un concepto fundamental de teoría de restricciones: El concepto de cuello de botella. Debe recordarse que un cuello de botella corresponde al recurso más lento dentro de una secuencia de procesamiento, y por ello define en gran medida el tiempo de ciclo de la producción. Esto es claro y evidente para sistemas de manufactura continuos; sin embargo, no se puede aplicar de una forma tan clara en sistemas de producción Job Shop, debido a que cada trabajo puede presentar secuencias de producción diferentes.

La idea clave del proyecto consiste en las siguientes premisas:

- Los recursos que más limiten la producción deben trabajar el mayor tiempo posible
- Si en cierto instante un recurso se vuelve cuello de botella, debe asignarse al trabajo disponible que lo desocupe más rápido, de manera que este pueda hacer uso de otros recursos que pueden llegar a ser críticos.

Lo anterior se sintetiza en los siguientes pasos:



**Paso 1:** Hacer  $T_i = 0 \forall i$  y  $C_j = 0 \forall j$

**Paso 2:** Definir la variable  $W_i = 1 \forall i$ . Esta variable indica, dentro de la secuencia de cada trabajo, cuál es la próxima máquina a visitar. Por defecto, todos los trabajos deben ir a la primera máquina de su secuencia.

**Paso 3:** Para cada trabajo calcular:

$$f_i = \frac{p_{S_i(W_i)}^i}{\max\{p_{S_k(W_k)}^i\}}, \text{ donde } i \leq k \leq |S_i| \quad (1)$$

**Paso 4:** Encontrar:

$$\operatorname{argmin}_j C_j \quad (2)$$

**Paso 5:** Encontrar el trabajo con mayor  $f_i$ , tal que  $T_i \geq C_j$ . Hacer  $W_i = W_i + 1$ ,  $C_j = T_i + p_{ij}$ ,  $T_i = C_j$ . Luego regresar al Paso 3. Terminar cuando todos los trabajos queden programados

### 10.3. Generación de instancias aleatorias

Dado que se trata de un proyecto de carácter teórico, la eficiencia de la heurística debe validarse de alguna manera. Para ello, en la literatura los autores emplean instancias generadas de forma aleatoria, y luego resuelven dichas instancias con el algoritmo que se propone. Las mismas instancias son resueltas con otras técnicas relevantes en el problema, comparando los resultados obtenidos para luego extraer las respectivas conclusiones de interés en el marco del proyecto.

En general, para este proyecto deben generarse las entradas  $N$ ,  $M$ ,  $p_{ij}$  y  $S_i$ . Es decir, debe generarse el número de trabajos, el número de máquinas, el tiempo de cada trabajo en cada máquina y la secuencia u orden en el que cada trabajo visita

las máquinas. Para hacer lo más sencilla posible la evaluación de la eficiencia de la heurística, se trabajará bajo las siguientes premisas:

- Los tiempos de procesamiento se ajustarán a una distribución uniforme
- Todos los trabajos pasarán por todas las máquinas

Así, los tiempos de procesamiento se generarán mediante la siguiente ecuación:

$$p_{ij} = a + r * [b - a] \quad (3)$$

En la Ecuación 1,  $a$  representa el tiempo de procesamiento más bajo deseado,  $b$  el tiempo más alto deseado y  $r$  es un número aleatorio entre 0 y 1. Gracias a estas consideraciones, se espera que al final las máquinas queden balanceadas en cuanto a sus tiempos de terminación, de manera que se obtenga una cota inferior representativa de la solución óptima, evitando así tener que obtener la solución óptima mediante programación matemática. Esto se evidenciará con más detalle en el numeral de resultados.

Para que la complejidad de soluciones posibles no sea extremadamente grande, se optará por trabajar instancias con 4 máquinas; así, el total de posibles secuencias de procesamiento será de  $4! = 24$ . Para las pruebas se usará un total de 20 trabajos, ya que usar una cantidad mayor podría implicar una mayor capacidad de cómputo. Para la implementación del algoritmo, se diseñó una herramienta en VBA para Excel, denominada HeuroJob 1.0. Un vistazo de la herramienta se observa a continuación:

Heurística - Microsoft Excel (Error)

Archivo Inicio Insertar Diseño de página Fórmulas Datos Revisar Vista Programador Comple

Calibri 11 Fuente Alineación General

Portapapeles Fuente Alineación Núm

F6C1 1

	1	2	3	4	5	6	7	8
1	Generar Datos							
2					Permutación		Trabajo	
3	1	2	3	4	1		1	1
4	1	2	4	3	2		2	4
5	1	3	2	4	3		3	4
6	1	3	4	2	4		4	4
7	1	4	2	3	5		5	3
8	1	4	1	2	6		6	4
9	2	1	2	1	7		7	2
10	2	1	4	4	8		8	2
11	2	3	3	4	9		9	1
12	2	3	4	1	10		10	3
13	2	4	1	3	11		11	1
14	2	4	3	1	12		12	1
15	3	1	2	4	13		13	3
16	3	1	4	2	14		14	3
17	3	2	1	4	15		15	4
18	3	2	4	1	16		16	1
19	3	4	1	2	17		17	2
20	3	4	2	1	18		18	2
21	4	1	2	3	19		19	2
22	4	1	3	2	20		20	1
23	4	2	1	3	21			

Trabajo	Secuencia	Ejecutar Heurística	Cmax M1	Cmax M2	Cmax M3	Cmax M4		
1	1	4	2	3	85	86	55	40
2	4	1	2	3	11	27	53	22
3	4	2	3	1	17	87	50	20
4	4	3	2	1	63	55	47	81
5	3	1	4	2	19	99	45	19
6	4	1	3	2	76	49	67	27
7	2	4	1	3	90	50	24	80
8	2	4	1	3	40	90	20	11
9	1	4	2	3	11	83	51	54
10	3	2	4	1	59	75	10	32
11	1	3	4	2	12	32	70	29
12	1	3	4	2	74	43	44	73
13	3	2	1	4	79	11	21	73
14	3	1	2	4	66	69	79	39
15	4	2	3	1	14	36	20	65
16	1	3	2	4	51	83	43	75
17	2	3	3	4	73	95	12	53
18	2	1	4	4	19	84	28	81
19	2	4	3	1	30	100	66	35
20	1	3	4	2	43	26	11	82
Peso								
Ocupación								

El botón azul sirve para generar los datos de prueba (tiempos de procesamiento y secuencias de producción). El botón rojo, sirve para ejecutar la heurística. Los resultados, respecto al tiempo de terminación de cada máquina, se muestran en las celdas de color rojo.

## 11. RESULTADOS Y CONCLUSIONES

La heurística se probó en un total de 100 instancias, cada una con 20 trabajos. Debido a la dificultad de usar modelos de programación lineal para obtener la solución óptima para las instancias, se optó por usar soluciones relajadas. Para obtener dichas soluciones luego de una programación, se aplica la ecuación:

$$C_{max}^{opt} \geq \frac{C_1 + C_2 + C_3 + C_4}{4} \quad (3)$$

En lo anterior es fácil probar que (3) corresponde a una cota inferior para  $C_{max}^{opt}$ . Supóngase una instancia donde los tiempos de terminación de  $n$  máquinas son  $C_1, C_2, \dots, C_n$ , con  $C_1 \leq C_2 \leq \dots \leq C_n$ . Por contradicción, supongamos que  $C_{max}^{opt} < \bar{C}$ . Tendríamos que:

$$C_{max}^{opt} < \frac{C_1 + C_2 + \dots + C_n}{n} = \frac{C_1 + C_2 + \dots + C_{C_{max}^{opt}}}{n}$$

$$\rightarrow n * C_{max}^{opt} < C_1 + C_2 + \dots + C_{n-1} + C_{C_{max}^{opt}}$$

$$\rightarrow (n - 1) * C_{max}^{opt} < C_1 + C_2 + \dots + C_{n-1}$$

$$C_{max}^{opt} < \frac{C_1 + C_2 + \dots + C_{n-1}}{n - 1} \rightarrow C_{max}^{opt} < C_{n-1}$$

Lo anterior es una contradicción ya que por hipótesis  $C_{max}^{opt}$  es mayor a  $C_{n-1}$ , con lo que se demuestra la cota.

Lo anterior es muy claro. El mejor escenario posible resulta cuando todas las máquinas quedan niveladas (terminan al mismo tiempo). Si alguna demora menos de este promedio, implica automáticamente que al menos una gastará un tiempo superior al promedio. Por tanto,  $C_{max}^{opt}$  es la mejor solución posible.

Usando la herramienta HeuroJob 1.0, se obtuvieron los siguientes resultados:

N°	Cmax M1	Cmax M2	Cmax M3	Cmax M4	MAX	Óptimo*	Gap %
1	1062	991	969	1008	1062	1007,5	5,4%
2	1269	1181	1239	1211	1269	1225	3,6%
3	878	887	907	849	907	880,25	3,0%
4	724	752	822	744	822	760,5	8,1%
5	964	975	906	947	975	948	2,8%
6	911	963	963	873	963	927,5	3,8%
7	1037	1030	1005	1035	1037	1026,75	1,0%
8	755	765	709	725	765	738,5	3,6%
9	811	850	806	853	853	830	2,8%
10	888	861	839	925	925	878,25	5,3%
11	1202	1290	1197	1170	1290	1214,75	6,2%
12	1035	1040	1040	1086	1086	1050,25	3,4%
13	609	594	589	596	609	597	2,0%
14	1037	1086	1074	1081	1086	1069,5	1,5%
15	679	631	624	638	679	643	5,6%
16	49	33	56	28	56	41,5	34,9%
17	991	972	968	988	991	979,75	1,1%
18	1005	1009	1033	1039	1039	1021,5	1,7%
19	1089	1090	1112	1098	1112	1097,25	1,3%
20	1197	1115	1115	1113	1197	1135	5,5%
21	998	1046	994	996	1046	1008,5	3,7%
22	828	848	827	866	866	842,25	2,8%
23	612	677	765	646	765	675	13,3%
24	502	480	536	510	536	507	5,7%
25	977	985	899	925	985	946,5	4,1%
26	1163	1168	1202	1293	1293	1206,5	7,2%
27	700	763	704	726	763	723,25	5,5%
28	938	910	907	912	938	916,75	2,3%
29	372	338	315	315	372	335	11,0%
30	570	623	585	581	623	589,75	5,6%

31	1181	1186	1163	1115	1186	1161,25	2,1%
32	1033	1088	1064	1055	1088	1060	2,6%
33	769	782	761	758	782	767,5	1,9%
34	336	350	383	408	408	369,25	10,5%
35	1032	1093	1013	1028	1093	1041,5	4,9%
36	1077	1150	1145	1118	1150	1122,5	2,4%
37	942	960	952	898	960	938	2,3%
38	931	967	923	933	967	938,5	3,0%
39	1042	1056	1102	1032	1102	1058	4,2%
40	523	519	476	486	523	501	4,4%
41	1185	1116	1098	1118	1185	1129,25	4,9%
42	1143	1138	1159	1159	1159	1149,75	0,8%
43	1064	1087	1040	1062	1087	1063,25	2,2%
44	1031	1069	1043	1066	1069	1052,25	1,6%
45	486	521	481	491	521	494,75	5,3%
46	1223	1149	1227	1156	1227	1188,75	3,2%
47	1150	1177	1157	1132	1177	1154	2,0%
48	1214	1225	1231	1196	1231	1216,5	1,2%
49	956	992	911	965	992	956	3,8%
50	744	888	742	722	888	774	14,7%
51	385	368	409	374	409	384	6,5%
52	1046	1048	1020	1098	1098	1053	4,3%
53	655	612	602	606	655	618,75	5,9%
54	982	921	929	944	982	944	4,0%
55	275	255	246	230	275	251,5	9,3%
56	411	405	416	392	416	406	2,5%
57	1208	1178	1146	1168	1208	1175	2,8%
58	122	127	131	142	142	130,5	8,8%
59	1092	1073	1050	1103	1103	1079,5	2,2%
60	804	800	774	792	804	792,5	1,5%

61	838	833	805	785	838	815,25	2,8%
62	766	709	708	705	766	722	6,1%
63	843	856	837	879	879	853,75	3,0%
64	121	131	119	124	131	123,75	5,9%
65	921	915	913	952	952	925,25	2,9%
66	966	960	1019	963	1019	977	4,3%
67	682	685	697	734	734	699,5	4,9%
68	226	273	234	266	273	249,75	9,3%
69	1207	1203	1227	1225	1227	1215,5	0,9%
70	1003	969	991	944	1003	976,75	2,7%
71	115	113	132	113	132	118,25	11,6%
72	621	589	571	603	621	596	4,2%
73	490	394	418	466	490	442	10,9%
74	966	971	955	985	985	969,25	1,6%
75	830	794	797	791	830	803	3,4%
76	799	807	894	803	894	825,75	8,3%
77	900	936	893	1046	1046	943,75	10,8%
78	927	916	915	941	941	924,75	1,8%
79	969	1001	963	1017	1017	987,5	3,0%
80	1056	1003	974	1032	1056	1016,25	3,9%
81	882	900	839	859	900	870	3,4%
82	943	938	935	976	976	948	3,0%
83	1222	1176	1155	1142	1222	1173,75	4,1%
84	1099	1081	1045	1105	1105	1082,5	2,1%
85	534	569	504	547	569	538,5	5,7%
86	585	631	571	577	631	591	6,8%
87	1013	1095	1038	1064	1095	1052,5	4,0%
88	823	830	801	830	830	821	1,1%
89	632	589	563	634	634	604,5	4,9%
90	1057	1039	1017	1051	1057	1041	1,5%
91	1130	1250	1203	1222	1250	1201,25	4,1%
92	946	901	887	872	946	901,5	4,9%
93	1304	1254	1187	1209	1304	1238,5	5,3%
94	834	811	798	811	834	813,5	2,5%
95	1118	1076	1154	1101	1154	1112,25	3,8%
96	930	915	898	899	930	910,5	2,1%
97	177	140	137	133	177	146,75	20,6%
98	1136	1143	1132	1202	1202	1153,25	4,2%
99	1081	1072	1132	1083	1132	1092	3,7%
100	94	97	112	92	112	98,75	13,4%

De lo anterior, se obtienen varias estadísticas de interés:

- El 90% de las veces, el gap se encuentra por debajo del 10%. Esto quiere decir que la mayoría de las veces la eficiencia de la heurística se encuentra por encima del 90%.
- El 68% de las veces, el gap se encuentra por debajo del 5%.



- Sólo el 2% de las veces, el gap estuvo por encima del 15%.

Lo anterior revela el alto nivel de rendimiento de la heurística, aun cuando fue comparada con la solución más relajada posible.

El gap promedio fue de 4.9%, con una desviación estándar del 4.5%. Comparando la media y la desviación, se puede presumir una alta variabilidad en los resultados; sin embargo, ya se aclaró que el 90% de los resultados tiene un gap inferior al 10%.

Por tanto, se espera que la mayoría de las veces, el algoritmo tenga un rendimiento que se encuentre por encima del 90%. Con la herramienta HeuroJob 1.0, los interesados en programar sistemas de producción Job Shop podrán tener soluciones de alta calidad en muy poco tiempo.

Desde el punto de vista estadístico, se concluye que la heurística es eficiente en más del 90% de las veces, teniendo un gap inferior al 10% respecto a la cota ya mencionada. Esto implica que el comportamiento real de la heurística es mejor al descrito, pues se ha comparado con un valor que realmente es menor al óptimo de cada instancia. Cabe aclarar que estos resultados son válidos solo para instancias de tamaño pequeño. Para instancias grandes, donde la eficiencia baja (menor al 60% según pruebas piloto realizadas) los resultados arrojados por la heurística pueden emplearse como base para el desarrollo de otros algoritmos tales como algoritmos genéticos.

## **12. RECOMENDACIONES Y FUTURAS INVESTIGACIONES**

La heurística propuesta, aunque muestra ser eficiente, debe ser puesta a prueba con un mayor número de trabajos y un mayor número de máquinas. Sin embargo, esto escapa al alcance de este proyecto, ya que en las PYMES la cantidad de estaciones de trabajo suele ser reducida, razón por lo cual el algoritmo propuesto muestra ser prometedor. Por otro lado, se podrían proponer modificaciones futuras en la estructura de funcionamiento del procedimiento, como también podría ser usado como generador de soluciones iniciales para algoritmos más sofisticados.

## REFERENCIAS BIBLIOGRAFICAS

- [1] B. J. Lageweg, J. K. Lenstra, and A. H. G. R. Kan, "Job-Shop Scheduling by Implicit Enumeration," *Manage. Sci.*, vol. 24, no. 4, pp. 441–450, 1977.
- [2] J. Adams, E. Balas, and D. Zawack, "The Shifting Bottleneck Procedure for Job Shop Scheduling," *Manage. Sci.*, vol. 34, no. 3, pp. 391–401, 1988.
- [3] S. DAUZERE-PERES and J.-B. LASSERRE, "A modified shifting bottleneck procedure for job-shop scheduling," *Int. J. Prod. Res.*, vol. 31, no. 4, pp. 923–932, 1993.
- [4] J. Carlier and E. Pinson, "An Algorithm for Solving the Job-Shop Problem," *Manage. Sci.*, vol. 35, no. 2, pp. 164–176, 1989.
- [5] P. Brucker, B. Jurisch, F. B. Mathematik, U. Osnabriick, and D.-O. Germany, "A branch and bound algorithm for the job-shop problem \*," no. 92, 1994.
- [6] C. Bierwirth, "A generalized permutation approach to job shop scheduling with genetic algorithms," *Operations-Research-Spektrum*, vol. 17, no. 2, pp. 87–92, 1995.
- [7] Y. Chen, "A Genetic Algorithm for Job-Shop Scheduling," vol. 5, no. 3, pp. 269–274, 2010.
- [8] L. Gao, G. Zhang, L. Zhang, and X. Li, "Computers & Industrial Engineering An efficient memetic algorithm for solving the job shop scheduling problem," *Comput. Ind. Eng.*, vol. 60, no. 4, pp. 699–705, 2011.
- [9] R. Thamilselvan, "Integrating Genetic Algorithm , Tabu Search and Simulated Annealing For Job Shop Scheduling Proble," vol. 48, no. 5, pp. 42–54, 2012.
- [10] M. Kurdi, "Computers & Industrial Engineering A new hybrid island model genetic algorithm for job shop scheduling," *Comput. Ind. Eng.*, vol. 88, pp. 273–283, 2015.

- [11] D. Applegate and W. Cook, "A Computational Study of the Job-Shop Scheduling Problem," *ORSA J. Comput.*, vol. 3, no. 2, pp. 149–156, 1991.
- [12] E. Nowicki and C. Smutnicki, "An Advanced Tabu Search Algorithm for the Job Shop Problem," *J. Sched.*, vol. 8, no. 2, pp. 145–159, 2005.
- [13] J. Frausto-solis, "A New Algorithm That Obtains an Approximation of the Critical Path in the Job Shop Scheduling Problem A New Algorithm That Obtains an Approximation of the Critical Path in the Job Shop Scheduling Problem," no. June 2017, 2006.
- [14] G. R. Weckman, C. V Ganduri, and D. A. Koonce, "A neural network job-shop scheduler," *J. Intell. Manuf.*, vol. 19, no. 2, pp. 191–201, 2008.
- [15] S. Yang, D. Wang, T. Chai, and G. Kendall, "An improved constraint satisfaction adaptive neural network for job-shop scheduling," *J. Sched.*, vol. 13, no. 1, pp. 17–38, 2010.
- [16] S. Song, J. Ren, and J. Fan, "Improved Simulated Annealing Algorithm Used for Job Shop Scheduling Problems," in *Advances in Electrical Engineering and Automation*, A. Xie and X. Huang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 17–25.
- [17] X. Qiu and H. Y. K. Lau, "An AIS-based hybrid algorithm for static job shop scheduling problem," pp. 489–503, 2014.
- [18] K. Akram, P. Graduate, K. K. Assistant, and A. Zeb, "Fast simulated annealing hybridized with quenching for solving job shop scheduling problem," vol. 49, pp. 510–523, 2016.