



**Formulación y validación de una heurística para minimizar el máximo tiempo de finalización en el problema de scheduling Job-shop Multi-recurso con rutas lineales.**

**Alejandro González Saavedra**

**Facultad de Ingeniería  
Programa de Ingeniería Electrónica**

Ibagué, 2019

**Formulación y validación de una heurística para minimizar el máximo tiempo de finalización en el problema de scheduling Job-shop Multi-recurso con rutas lineales.**

**Alejandro González Saavedra**

Trabajo de grado que se presenta como requisito parcial para optar al título de:

**Ingeniero Electrónico**

Director:

Dr. Andrés Alberto García León  
Profesor Universidad de Ibagué

**Facultad de Ingeniería**  
**Programa de Ingeniería Electrónica**  
Ibagué, 2019

## Dedicatoria

*Este trabajo principalmente se lo dedico a DIOS, por otorgarme la sabiduría necesaria para culminarlo; también, por permitirme no desfallecer y llegar a este punto tan importante en mi formación profesional.*

*A mi madre Norma, porque siempre está presente en mi camino apoyándome y animándome en cualquier circunstancia; por ofrecerme con tanto esfuerzo una buena educación. Todo lo que soy se lo debo a ella...Porque es mi motivación día a día.*

*A mi abuela que me ofrece su apoyo incondicional, su entusiasmo y perseverancia en tiempos difíciles; su hermosa compañía me anima a ser una mejor persona, siempre de la mano de Dios y nuestra mamita la Santísima Virgen María.*

*A mi hermano, tíos y primos, gracias por su ayuda ilimitada.*

*A todas las personas que hicieron parte de una u otra manera en la realización de esta investigación; así mismo, en mi desarrollo personal y profesional. Muchas gracias...*

## **Agradecimientos**

A Dios y a María Santísima, por concederme el don de la vida, por otorgarme los medios necesarios para la culminación de este proyecto, superando todos los obstáculos que se presentaron en este largo camino, y haber llegado al final de este proyecto.

A mi familia, porque es el apoyo incondicional y desinteresado en cada una de las situaciones que enfrento a diario y porque con su gran amor me han brindado todo lo necesario para ser una persona íntegra.

Al Dr. Andrés Alberto García León por su apoyo y paciencia en la realización de este trabajo de grado.

## Resumen

En la actualidad las organizaciones están siendo obligadas a fomentar estructuras de operación flexible y que a su vez originen reducciones de costos, los cuales se revelan en el aprovechamiento de la maquinaria, los recursos y mano de obra. En la presente investigación, se introduce un problema que modela situaciones reales, el cual es de *scheduling Job-shop multi-recurso con rutas lineales - JSFMRL*, de este se despliegan dos sub-problemas basados en procesos de búsqueda local: asignación y secuenciación. El objetivo de ésta investigación es diseñar un instrumento computacional el cual permita optimizar el Makespan en el problema de *scheduling Job-shop multi-recurso con rutas lineales - JSFMRL*. Este instrumento computacional se diseñó en el software MATLAB R2014a; obteniendo, un resultado positivo en la disminución del mayor tiempo de finalización o tardanza total de los pedidos (Makespan), con un porcentaje promedio de rendimiento de la heurística del 21.1%.

**Palabras clave:** Job-Shop, heurística, Makespan, *JSFMRL*, asignación, secuenciación.

## Abstract

At present, organizations are being forced to promote structures of flexible operation that in turn cause cost reductions, which are revealed in the use of machinery, resources and labor. In this research, we introduce a problem that models real situations, which is scheduling Job-shop multi-resource with linear routes - JSFMRL, from these two sub-problems are based on local search processes: allocation and sequencing. The objective of this research is to design a computational instrument which allows to optimize the Makespan in the scheduling problem Job-shop multi-resource with linear routes - JSFMRL. This computational instrument was designed in MATLAB R2014a software; obtaining, a positive result in the decrease of the greater time of finalization or total delay of the orders (Makespan), with an average percentage of performance of the heuristic of 21.1%.

**Keywords:** Job-Shop, heuristics, Makespan, *JSFMRL*, assignment, sequencing.

# Contenido

	Pág.
Lista de figuras .....	VII
Lista de tablas .....	VIII
Lista de gráficas.....	IX
Lista de ecuaciones.....	X
Introducción .....	11
<b>1. Capítulo 1. Formulación del problema .....</b>	<b>13</b>
1.1 Descripción del problema Job-shop Multi-recurso con rutas lineales - JSFMRL	13
1.2 Grafo para el problema Job-shop Multi-recurso con rutas lineales - JSFMRL ...	14
1.3 Criterios a optimizar .....	17
1.4 Complejidad del Job-shop Multi-recurso con rutas lineales - JSFMRL .....	17
<b>2. Capítulo 2. Componentes del proceso de búsqueda .....</b>	<b>19</b>
2.1 Factibilidad del movimiento de operaciones críticas.....	19
2.1.1 Inversión de arcos críticos.....	20
2.1.2 Reasignación de un recurso de una operación crítica .....	21
2.2 Estimaciones de tiempos de finalización de pedidos.....	22
2.2.1 Estimación del movimiento al invertir $(x, y)$ .....	23
2.2.2 Estimación de reasignar el recurso $L$ de $\epsilon$ a $L'$ entre $j$ y $k$ .....	23
2.3 Desarrollo de la heurística .....	24
2.4 Solución inicial para Job-shop multi-recurso con rutas lineales - JSFMRL .....	25
<b>3. Capítulo 3. Resultados .....</b>	<b>29</b>
3.1 Solución Inicial para una instancia propuesta .....	29
3.2 Optimización del makespan .....	33
<b>4. Conclusiones y recomendaciones .....</b>	<b>35</b>
4.1 Conclusiones .....	35
4.2 Recomendaciones .....	36
<b>A. Anexo: Archivos de las mejores soluciones de la heurística.....</b>	<b>37</b>
<b>Referencias bibliográficas.....</b>	<b>38</b>

# Lista de figuras

	Pág.
<b>Figura 1.</b> Ejemplo de grafos disyuntivo y conjuntivo para el JSP. ....	15
<b>Figura 2.</b> Asignación de máquinas en una operación.....	16
<b>Figura 3.</b> Asignación de recursos en una operación. ....	20
<b>Figura 4.</b> Inversión de un arco crítico $(x, y)$ . ....	21
<b>Figura 5.</b> Predecesores y sucesores inmediatos de una operación $x$ .....	22
<b>Figura 6.</b> Grafo de la instancia propuesta. ....	26
<b>Figura 7.</b> Asignación de máquinas a las operaciones. ....	26
<b>Figura 8.</b> Secuenciación para la instancia propuesta. ....	27
<b>Figura 9.</b> Trazado de arcos conjuntivos para la secuencia.....	28
<b>Figura 10.</b> Solución inicial para una instancia propuesta. ....	29
<b>Figura 11.</b> Ruta crítica para la instancia propuesta. ....	30
<b>Figura 12.</b> Arcos y colas máximas para la instancia propuesta.....	31
<b>Figura 13.</b> Mejor solución encontrada para una instancia propuesta. ....	34
<b>Figura 14.</b> Pantalla de la mejor solución generada por el software. ....	34

## Lista de tablas

Pág.

**Tabla 1.** Comparación entre la solución inicial de referencia y la heurística.....32

## Lista de gráficas

Pág.

**Gráfica 1.** Porcentaje de rendimiento de las instancias y de la heurística. ....33

## Lista de ecuaciones

	Pág.
<b>Ecuación 1.</b> Estimación de $C_i$ al invertir un arco crítico $(x, y)$ . .....	23
<b>Ecuación 2.</b> Estimación de $C_i$ al reasignar $L$ de $\epsilon$ a $L'$ entre $j$ y $k$ . .....	24

# Introducción

Los procesos productivos a lo largo de la historia se han acoplado a las necesidades que implantan las circunstancias socio-culturales; hace algunas décadas la necesidad por satisfacer grandes demandas llevó a los procesos productivos a industrializarse y encaminarse a una producción acelerada de un servicio o bien determinado. En la actualidad, la administración de Operaciones promueve a las organizaciones a la obtención de artículos de máxima calidad con precios cada vez más bajos. Para lograr este propósito, las organizaciones deben desdoblarse en estructuras organizativas capaces de maximizar simultáneamente la productividad y el servicio al cliente.

El presente trabajo de grado tiene como objetivo proponer una heurística de búsqueda local para minimizar el mayor tiempo de finalización de pedidos ( $C_{max}$ ) en el problema de scheduling Job-shop multi-recurso con rutas lineales - *JSFMRL*. El cual, modela condiciones reales de los procesos industriales en donde existe un conjunto de máquinas que trabajan de forma simultáneamente para elegir un sub-conjunto de ellas y solucionar cada una de las operaciones de todos los pedidos de un respectivo trabajo. Por ende, se dilatan los resultados obtenidos en la investigación de García, Dauzère & Yazid (2015) basados en la búsqueda de soluciones innovadoras para el problema de Job-shop Flexible para la validación de la eficiencia de la heurística, 35 de las 68 instancias de Dauzère Pérès (1998) han sido consideradas; las cuales, se manejaron de forma aleatoria para un resultado más ecuánime.

Según T' Kindt & Billaut, la complejidad para solucionar los problemas de Scheduling es un proceso productivo y está determinada por la cantidad de máquinas vinculadas, el número de pedidos o de tareas, la flexibilidad de la operación, que se da en la capacidad de que pueda ser procesadas en varios tipos de máquinas o usar diferentes máquinas simultáneamente y el flujo que sigue el proceso de transformación (T' Kindt & Billaut, 2006). El problema de Scheduling es estimado computacionalmente como NP-hard. Es decir, requiere de esfuerzo y consumo computacional al no ser factible de ser solucionado en tiempos racionales, incluso en problemas con pocas máquinas y pedidos y sin consideraciones de flexibilidad (Brucker, 2007).

La manera de representar un problema de Scheduling ha sido el grafo disyuntivo y la forma para codificarlo ha sido tratada en diferentes maneras por ejemplo las matrices utilizadas por la programación genética y otras heurísticas inspiradas en la naturaleza y la física (colonia de hormigas, enjambre de abejas y recocido simulado). El valor computacional para la acumulación de soluciones es característico si se utilizan matrices. La búsqueda local disminuye este problema, en la medida que solo guarda la solución actual y el resto de soluciones solo son definidas por un conjunto de funciones de estimación; lo cual, disminuye el impulso en el proceso de optimización. Por ende, un proceso de búsqueda local es eficaz en la medida que las estimaciones planteadas



mitigan la diferencia entre una solución real y la evaluada a través de valores de referencia.

Este trabajo de grado procura disminuir la complejidad computacional en el desarrollo del problema; de igual manera, se proponen las siguientes preguntas para la investigación:

¿Qué elementos se deben considerar para determinar una solución inicial de calidad?

¿Cuáles deben ser las estrategias del proceso de búsqueda local para minimizar el makespan?

La construcción de este se fundamenta en un proceso de búsqueda local y para ello se propusieron tres objetivos específicos. El primer objetivo específico consiste en formular un algoritmo para determinar una solución inicial de calidad. El segundo objetivo es establecer funciones de estimación para el movimiento de operaciones críticas considerando el Makespan, y finalmente se culmina con una validación de las etapas de mejoramiento y una diversificación en el proceso de búsqueda local al minimizar el Makespan.

El documento está compuesto por tres capítulos: En el primer capítulo se plantea el problema de Scheduling Job-Shop Multi-recurso con rutas lineales con la definición del criterio a optimizar (Makespan). En el segundo capítulo, se plantea la formulación matemática para optimizar criterios en el problema; en el tercer y último capítulo se determina el proceso de búsqueda local y finalmente los experimentos computacionales que validan los capítulos anteriores.

# 1. Capítulo 1. Formulación del problema

En este primer capítulo se dará a conocer la representación del problema, en primera instancia se determina el problema; posteriormente, se detallará la representación por medio de grafos (disyuntivos y conjuntivos). Luego, las funciones a optimizar y en último lugar, la complejidad para la solución.

## 1.1 Descripción del problema Job-shop Multi-recurso con rutas lineales - JSFMRL

Para discernir el problema de la presente investigación, se pretende inicialmente definir el problema clásico de *Scheduling Job-shop* “*Job-shop Scheduling problem -JSP*”. *JSP* es considerado como uno de los problemas de optimización combinatoria más complejos respecto a los problemas de programación de producción.

En el problema se tiene un conjunto de  $n$  pedidos  $J = \{J_1, \dots, J_n\}$  en un conjunto  $M$  de  $m$  máquinas que están siempre disponibles para procesar los pedidos. Cada máquina sólo puede realizar una operación a la vez. Cada pedido consiste en una secuencia de operaciones fija denominada ruta. No se permite la prioridad de operaciones, lo que significa que una operación no puede ser interrumpida. Los tiempos de procesamiento de las operaciones son enteros, conocidos e incluyen los tiempos de alistamiento. Cada pedido  $J_i$  tiene una fecha de liberación  $r_i$  que significa que  $J_i$  no se puede iniciar antes de esta fecha, un peso  $w_i$  que expresa la importancia de  $J_i$ , y una fecha de vencimiento  $d_i$  que especifique el tiempo antes de que  $J_i$  debe ser entregado. El objetivo del problema es determinar un programa, es decir, la secuencia (orden) de las operaciones en las máquinas y los tiempos de finalización  $J_i$  para optimizar un objetivo (García León, Dauzère Pérès, & Yazid, 2015).

Según, García, Dauzère & Yazid (2015) el *JSP* permite representar configuraciones del escenario real, la tendencia moderna en la producción conlleva a la flexibilidad; por ejemplo, en las operaciones al poder ser realizada por varias máquinas de un grupo de posibilidades. Este supuesto no puede ser verificado por el *JSP*. Por lo tanto, la integración de esta flexibilidad en el *JSP* conduce a un modelo más general que es considerado por el Job-Shop Flexible multi-recurso con rutas lineales “*JSFMRL*”.

*JSFMRL* es una extensión del *JSP* en la que cada operación requiere por lo menos una máquina para ser procesada. La máquina o las máquinas dependiendo de las características de la operación son seleccionadas de un subconjunto de  $M$ . Para obtener una solución factible del *JSFMRL*, las operaciones deben ser asignadas y secuenciadas en máquinas. La asignación consiste en seleccionar, para cada operación, la(s) máquina(s) que realizará(n) la operación mientras que la secuenciación trata de obtener el orden de las operaciones en cada una de las máquinas seleccionadas. El *JSFMRL*



tiene como objetivo optimizar un criterio o múltiples criterios (García León, Dautère Pérès, & Yazid, 2015).

## 1.2 Grafo para el problema Job-shop Multi-recurso con rutas lineales - JSFMRL

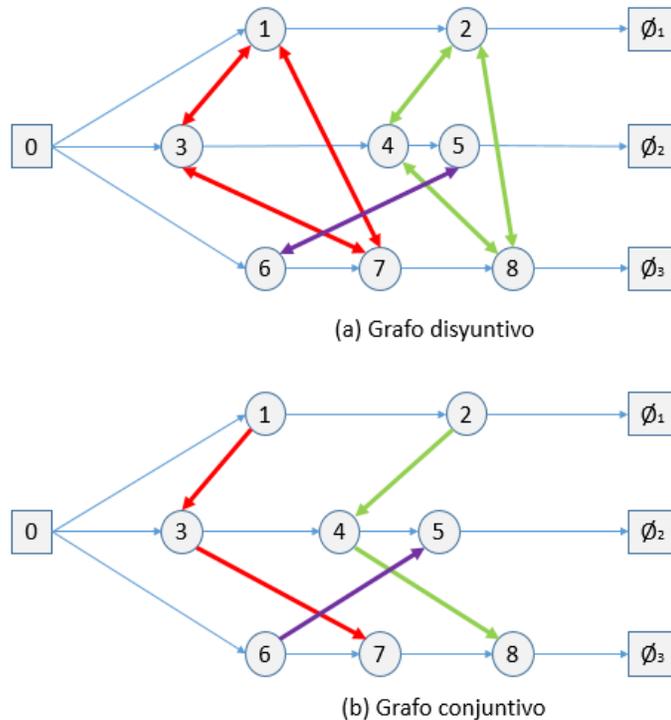
Originalmente el grafo disyuntivo se desarrolló por Roy B. y Sussmann B. (1964) y considerablemente manejado más adelante para disipar los problemas clásicos de programación Job-shop. Una ventaja importante de este grafo es su capacidad para modelar de manera eficiente diferentes restricciones y características de los problemas de programación de talleres. En esta sección, se explica cómo se usa este grafo para modelar el problema con criterios regulares (Torres Tapia, 2018).

El gráfico se representa como  $G = (V; A; E)$ , donde  $V$  es el conjunto de nodos asociados a las operaciones de pedidos, un nodo ficticio 0 que representa el inicio de cada trabajo, y  $n$  nodos  $\phi_i$ , cada uno de los cuales representa el tiempo de finalización de un trabajo  $J_i$ . El conjunto de arcos conjuntivos  $A$  contiene arcos que conectan dos operaciones consecutivas de cada trabajo, el nodo 0 y cada primera operación de cada trabajo, y la última operación de cada  $J_i$  con el nodo  $\phi_i$ . Finalmente,  $E_k$  es el conjunto de arcos disyuntivos entre pares de operaciones asignados a cualquier máquina  $k$  y  $E = \bigcup_{k \in M} E_k$ . El arco de 0 a la primera operación de un trabajo  $J_i$  tiene una longitud igual a la fecha de liberación  $r_i$  de  $J_i$ . Cualquier arco conjuntivo o disyuntivo restante tiene una longitud igual al tiempo de procesamiento de la operación desde la que comienza (García León, Dautère Pérès, & Yazid, 2015).

En la Figura 1 se pueden visualizar los grafos con que se simboliza el problema de JSP y en este caso un problema con tres pedidos (arcos horizontales) con nodos terminales  $\phi_i$ . Para el problema el pedido 1 tiene dos operaciones; las cuales, son simbolizadas por los nodos 1 y 2. En relación a los nodos 3,4 y 5 forman parte del pedido 2 y los otros para el pedido 3. En el grafo de la parte superior (disyuntivo), se visualiza el conflicto en la secuencia de las operaciones, los arcos con doble flecha; como se ilustran en la figura, los de color rojo, con un algoritmo se debe determinar la dirección o la secuencia de operaciones para optimizar algún criterio. Al solucionar el conflicto se consigue un grafo conjuntivo (grafo de la parte inferior) el cual debe asegurar la no existencia de un ciclo en la solución.

Según García, Dautère, & Yazid (2015); una selección  $S_k$ , que corresponde a un programa del JSP para una tarea determinada, se obtiene fijando una dirección a cada arco disyuntivo en  $E$ . La selección es factible si el gráfico inducido no conduce a un ciclo. Vale la pena mencionar que el grafo contiene muchos arcos redundantes que deben eliminarse para garantizar que cada nodo tenga como máximo un predecesor y un sucesor en la máquina que se realice. Por lo tanto, todos los nodos correspondientes a las operaciones del JSP tendrán dos arcos de entrada y dos arcos de salida, excepto la

primera (o la última) operación en la secuencia de máquinas que solo tendrá un arco de entrada (o de salida).



**Figura 1.** Ejemplo de grafos disyuntivo y conjuntivo para el JSP.

**Fuente:** Autor.

JSFMRL adiciona dos tipos de flexibilidad a JSP: Multi-recurso y recurso flexible. Multi-recurso considera que para realizar una operación se requiere de una varias o maquinas simultáneamente, mientras que recurso flexible implica la selección de estas de un conjunto dado. Así mismo, el tiempo de procesamiento de la operación estará determinado por el mayor tiempo de operación de la máquina seleccionada (Torres Tapia, 2018).

Para esclarecer la información mencionada por Torres Tapia (2018), la Figura 2 ilustra un ejemplo de una operación que requiere tres máquinas (recursos). Para este ejemplo, la primera máquina debe ser elegida entre  $M_1$ ,  $M_2$  o  $M_3$  (rectángulo superior en a) y el tiempo de procesamiento está establecido por el número inferior entre paréntesis. En la segunda máquina la selección se efectuará entre  $M_1$ ,  $M_2$  o  $M_4$  y para la tercera entre  $M_2$  o  $M_4$ . Es importante recalcar que solo se presenta una máquina de cada tipo. El rectángulo de la derecha (b) enseña la asignación que se ha realizado en el cual se seleccionaron las máquinas  $M_2$ ,  $M_1$  y  $M_4$  respectivamente; además, el tiempo de procesamiento de la operación es de 3 unidades.



En JSFMRL adquirir una solución factible, comprende solucionar el problema de asignación en las operaciones y de igual manera, la secuencia en las máquinas sin originar ciclos. En relación al grafo, en cada operación deben ingresar y salir de una operación cuantos recursos estén asignados. El tiempo de inicio de la operación será el de mayor tiempo de finalización de todos sus predecesores contiguos. Una vez haya culminado la solución de un grafo, se puede obtener información que favorecerá a la construcción de heurísticas.

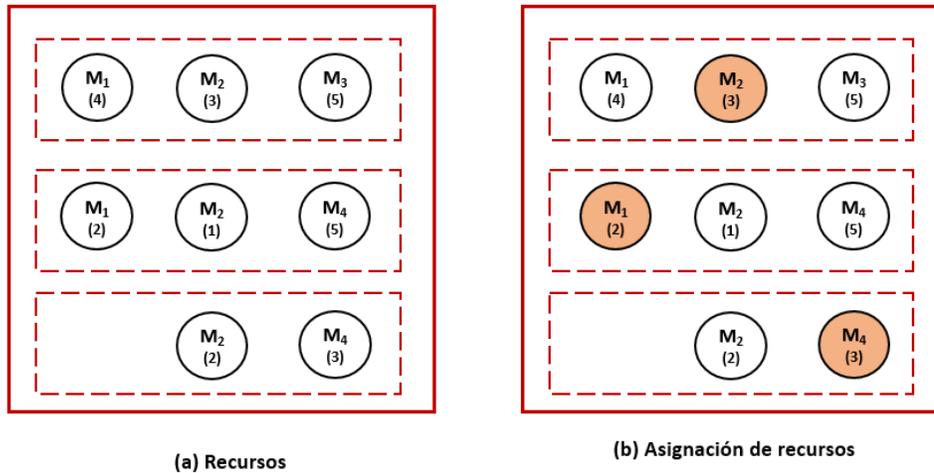


Figura 2. Asignación de máquinas en una operación.

Fuente: Autor.

El tiempo de inicio  $h_x = L(0, x)$  de un nodo  $x$ , comúnmente llamado cabeza, viene dado por la longitud de la trayectoria más larga desde 0 o nodo de inicio hasta  $x$ . Por lo tanto, el tiempo de finalización  $C_x$  de la operación  $x$  es  $h_x + p_x$ , donde  $p_x$  es el tiempo de la operación y el tiempo de finalización  $C_i$  de un trabajo  $J_i$  es igual a  $h_{\phi_i}$ . El nivel  $l_x$  de la operación  $x$  representa la cantidad máxima de arcos desde el nodo 0 a  $x$ . La cola  $q_x^i = L(x, \phi_i) - p_x$  de una operación  $x$  es la longitud máxima desde la terminación de  $x$  hasta el nodo ficticio  $\phi_i$  si existe una ruta de  $x$  a  $\phi_i$  y  $-\infty$  de lo contrario, se denota por  $q_x$  la cola clásica de  $x$  que consiste en la longitud máxima desde la terminación de  $x$  hasta cualquier nodo  $\phi_i$ , es decir  $q_x = \max_{1 \leq i \leq n} q_x^i$ . La ruta más larga desde el nodo 0 al nodo  $\phi_i$  se llama ruta crítica de 0 a  $\phi_i$  y su longitud es igual a  $C_i$ . Cada operación  $x$  que pertenece a esta ruta crítica es crítica según  $J_i$  y satisface  $h_x + p_x + q_x^i = C_i$ . Cada arco  $(x, y)$  que pertenece a la ruta crítica de 0 a  $\phi_i$  es crítico si las operaciones  $x$  y  $y$  están asignadas a la misma máquina. Un bloque es una secuencia de operaciones críticas asignadas a la misma máquina (García León, Dauzère Pères, & Yazid, 2015). Es relevante indicar que para el caso del presente problema, si  $(x, y)$  son operaciones consecutivas, y se ejecutan en la misma máquina pertenecientes a diferentes pedidos, todos los recursos que sean comunes se convertirán en arcos críticos.

### 1.3 Criterios a optimizar

Según Torres Tapia (2018), uno de los principales aspectos de la función de operaciones es lograr el uso eficiente de las máquinas con el fin de mejorar la productividad e interactuar con marketing para mejorar el servicio al cliente. La consideración de la productividad y el servicio al cliente requieren de funciones objetivas basadas en los tiempos de finalización de los pedidos. Diversas funciones objetivas han sido consideradas y entre los objetivos más comunes se encuentran los regulares. Un criterio es regular si es una función creciente de los tiempos de finalización ( $C_i \forall i \in J$ ) de los pedidos, es decir, siempre es óptimo comenzar y completar los trabajos tan pronto como sea posible (véase, por ejemplo, (Mati, Dauzère-Pérès, & Lahlou, 2011)).

El criterio regular más analizado es el Makespan ( $C_{max}$ ) que consiste en minimizar el tiempo máximo de finalización de todos los pedidos. Sin embargo, minimizar otros criterios capturan factores críticos que afectan la rentabilidad y; por consiguiente, su competitividad. En particular la importancia de los pedidos y los clientes con criterios ponderados, como el número ponderado de trabajos tardíos y la suma ponderada de los tiempos de finalización de los pedidos (Cheng, W.; Zhang, R., 2011). Para la presente investigación se ha considerado la máxima tardanza ( $T_{max}$ ), en la cual la tardanza de un pedido está dada por  $T_i = \max(0, C_i - d_i)$  y por ende  $T_{max} = \max T_i, \forall i \in J$ .

En el tratamiento de la heurística hallar una solución la cual, su Tardanza máxima sea cero, conlleva que la solución a una programación respecto a la entrega de pedidos sea más efectiva. Sin embargo, es importante mencionar que, si se valora un escenario justo a tiempo, la diferencia entre la finalización del pedido y la fecha de entrega debe ser minimizada.

En la investigación realizada por T'kindt & Billaut (2006), se distingue a  $C_{max}$  y  $T_{max}$  como criterios "minimax" ya que corresponde al valor máximo de un conjunto de funciones a minimizar y son los más encontrados comúnmente en la literatura.

### 1.4 Complejidad del Job-shop Multi-recurso con rutas lineales - JSFMRL.

El *JSP*, es un problema combinatorio, que para solucionarlo al optimizar un criterio se requiere de un esfuerzo computacional que muestra las diferentes secuencias factibles sobre las máquinas. Si se considera la flexibilidad, en el caso de que una operación se pueda realizar en una única máquina seleccionada de un subconjunto es *NP-hard* "Non-deterministic Polynomial-time hardness", incluso para el caso particular con dos máquinas y tres pedidos, cuando el *Makespan* se minimiza (Brucker P. , 2007). Muy pocos casos especiales pueden ser resueltos polinomialmente, en la investigación realizada por Brucker & Schile (1990), se formuló un algoritmo polinomial denominado enfoque geométrico para programar dos pedidos en los que los tiempos de



procesamiento de las operaciones no dependen de la máquina seleccionada; es decir, con máquinas relacionadas, el algoritmo se puede utilizar para minimizar cualquier criterio regular y se aplica cuando se permite la prioridad de operaciones. Este enfoque geométrico se ha extendido posteriormente en la investigación de Hurink, Jurisch & Thole (1995), para tratar las fechas de lanzamiento y las restricciones de precedencia entre las operaciones.

En el documento de Mati & Xie (2004), los autores exhiben que con dos pedidos es *NP-Hard* “*Non-deterministic Polynomial-time hardness*”, cuando los tiempos de procesamiento de las operaciones dependen de la máquina seleccionada; es decir, con máquinas no relacionadas. También, generalizan el enfoque geométrico para minimizar cualquier criterio regular con dos pedidos en los que solo uno de ellos es flexible.

La *consideración de flexibilidad en JSP* es más engorroso debido a que hay que tomar dos decisiones, estas decisiones son los problemas anteriormente mencionados, asignación y secuenciación. El *NP-Hardness* es de los casos especiales del *JSP* el cual, sugiere las dificultades para solucionar este problema en casos generales con cualquier número de pedidos y de máquinas. En conclusión, la solución del JSFMRL se consolida en solucionar el desarrollo de heurísticas eficientes que producen soluciones de calidad en tiempos mínimos computacionales.

Según la investigación “A batch-oblivious approach for Complex Job-Shop Scheduling problems” de Knopp (2017), la modelación de este tipo de problemas por métodos de optimización como por ejemplo el uso de la Programación Lineal requerirá de tiempos exponenciales para su solución y que con la tecnología computacional vigente no se ha podido determinar con exactitud su duración, lo cual no contribuye a la toma de decisiones eficientes.

## **2. Capítulo 2. Componentes del proceso de búsqueda**

En este segundo capítulo se detallan los lineamientos que encaminan a construir un enfoque de búsqueda local. Para esta asistencia, el objeto del proceso de búsqueda local se fundamenta en establecer a partir de una solución simbolizada en el grafo la factibilidad del movimiento de operaciones y los tiempos de finalización de los pedidos sin hacer transformaciones. Si se realizase alguna transformación no tendría sentido la representación de este tipo de enfoques.

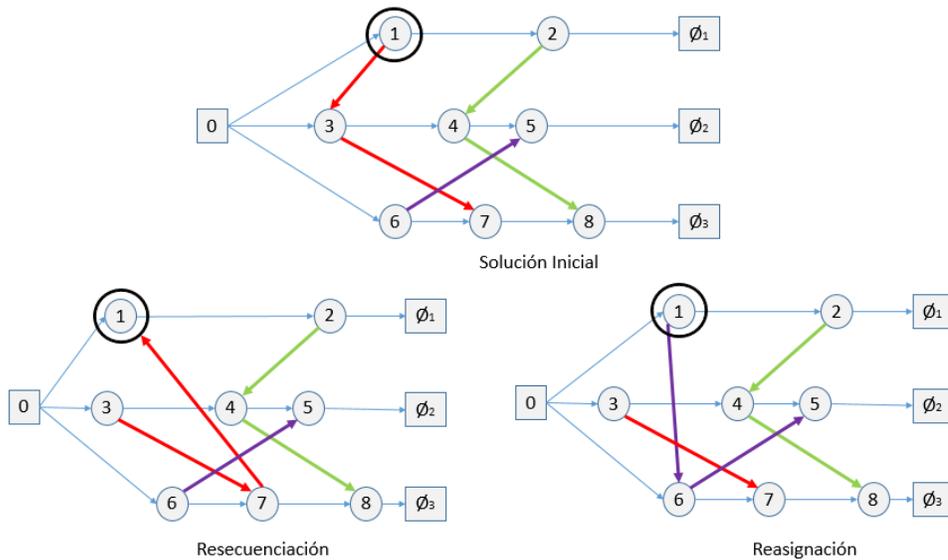
### **2.1 Factibilidad del movimiento de operaciones críticas**

En el JSFMRM dos tipos de movimientos que pueden ejecutarse considerando operaciones críticas, estos se denominan resecuenciación y reasignación. Evidentemente, estos movimientos son explicados sobre los recursos que se estén empleando en la operación y por los cuales se pueden sustituir. La resecuenciación se fundamenta en cambiar la posición de una operación en la secuencia de la máquina en que fue estipulada y la reasignación es la sustitución de un recurso por otro, siempre y cuando sea factible (García León, Dautère Pérès, & Yazid, 2015).

En la Figura 3 se puede visualizar un grafo conjuntivo; en el cual, se describe los movimientos de operaciones críticas, la resecuenciación y la reasignación en un JSP flexible en la asignación de máquinas, en este se estima que la operación 1, encerrada en un círculo es crítica y esta secuenciada en la máquina representada por los arcos rojos. La primera operación en la máquina es la operación 1, si se considera la resecuenciación en el grafo inferior izquierdo esta operación crítica se pasa a ubicar en la tercera posición (ver la secuencia de los arcos rojos). El grafo inferior derecho, enseña la reasignación; donde se asume que esta operación puede realizarse en la máquina que representa el color violeta, y en este caso pasaría a ser la primera operación (ver secuencia de los arcos violetas). Cabe mencionar que este movimiento no debe originar ciclo.

García, Dautère & Yazid (2015), mencionan en su investigación, que determinar este tipo de inserciones factibles sin hacer transformaciones en un grafo es computacionalmente costoso. Sin embargo, en la literatura diferentes proposiciones se han realizado para validar el movimiento conociendo los parámetros de una solución. De igual manera, se resalta el test HTA que demuestra la no existencia de ciclos al mover una operación crítica considerando los tiempos de inicio de operaciones, los números de arcos y las colas máximas de las operaciones.

En la presente investigación, se consideran dos tipos de movimientos, la reasignación de recursos según el número de arcos y la inversión de arcos críticos.



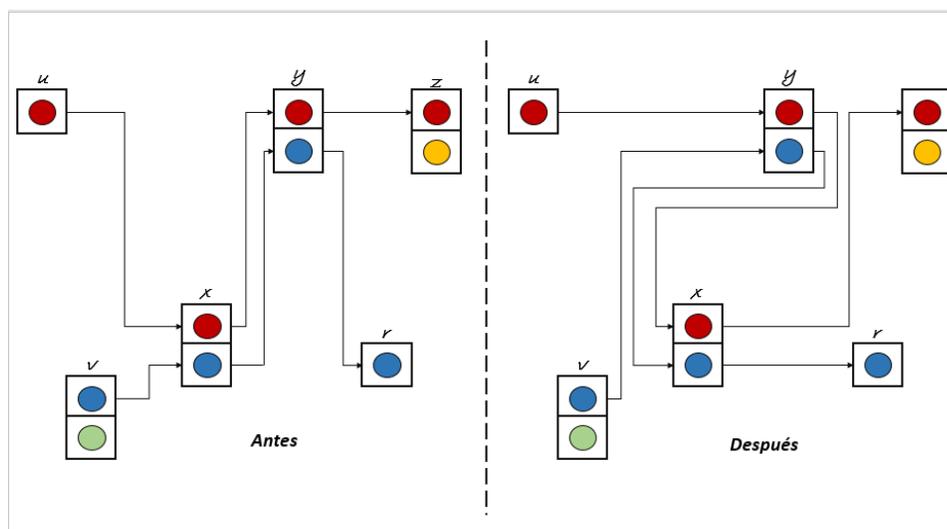
**Figura 3.** Asignación de recursos en una operación.

**Fuente:** Autor.

### 2.1.1 Inversión de arcos críticos

Uno de los movimientos permitidos sin requerimiento de validar su factibilidad en JSP, es la inversión de arcos críticos (Cheng, W.; Zhang, R., 2011). Para el caso de JSFMRL, esta inversión debe considerar la existencia de recursos en común entre las operaciones ( $x$ ,  $y$ ).

En la Figura 4 se evidencia el cambio en los arcos conjuntivos al modificar el orden en la asignación de recursos en  $x$  y  $y$  y representados con colores. Se observa, que en la situación de la derecha los arcos apuntan de  $y$  a  $x$  y los arcos que provienen de  $u$  y  $v$  que son los predecesores en los recursos para  $x$  se dirigen a  $y$ . De igual manera, los arcos que salen de  $x$  no están direccionados a  $y$  sino a los sucesores de los recursos críticos de  $y$  ( $z$  y  $r$ ).



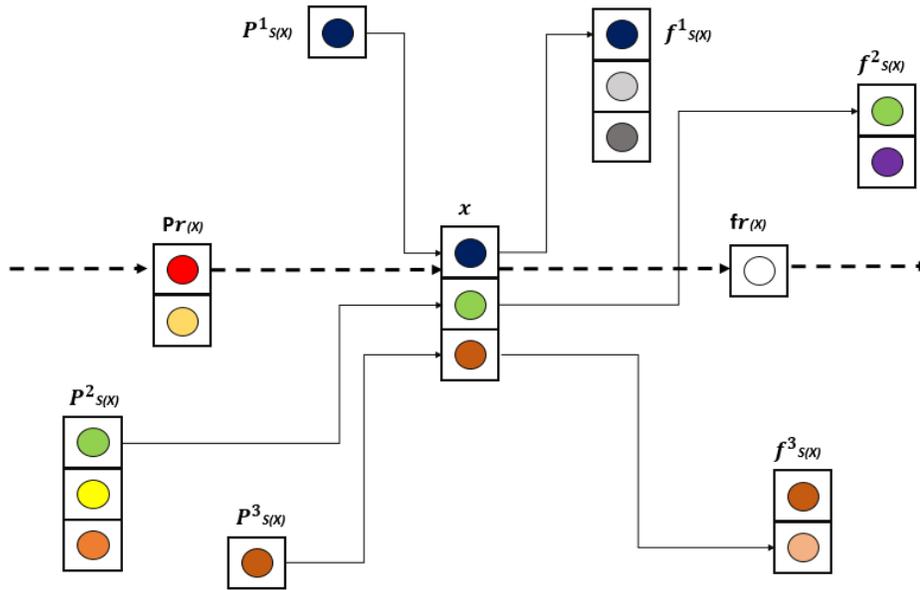
**Figura 4.** Inversión de un arco crítico  $(x, y)$ .

**Fuente:** Autor.

### 2.1.2 Reasignación de un recurso de una operación crítica

La cantidad de arcos que intervienen en la solución de JSFMRL, afecta directamente la complejidad, ya que el proceso de búsqueda local debe asegurar la no existencia de ciclos. En este caso se propone reasignar un recurso  $L$  de la operación  $\epsilon, \epsilon = x \vee \epsilon = y$  al recurso  $L'$  entre las operaciones  $j$  y  $k$ . Para validar la reasignación, los conceptos de predecesores y sucesores de la operación  $x$  son definidos (Torres Tapia, 2018).

Sea  $P_x$  (respectivamente  $F_x$ ), el conjunto de inmediatos predecesores (respectivamente sucesores) de la operación  $x$ . Sea además  $P_{r(x)}$  (respectivamente  $F_{r(x)}$ ) el inmediato predecesor en secuencia en ruta del pedido a que pertenece  $x$ . Cuantos recursos tenga la operación  $x$ , será el número de predecesores y sucesores en secuencia de los mismos que tendrá la operación y se representa como  $P_{s(x)}^k$  (respectivamente  $F_{s(x)}^k$ ) (Torres Tapia, 2018). En la Figura 5 se determina lo anteriormente exhibido con tres recursos ilustrados en colores para el procesamiento de la operación  $x$ .



**Figura 5.** Predecesores y sucesores inmediatos de una operación  $x$ .

**Fuente:** Autor.

Al tener precisión sobre los sucesores y predecesores, el proceso de búsqueda tiene como objetivo determinar la reasignación del recurso  $L$  de una operación crítica  $\epsilon$ , al recurso  $L'$  entre las operaciones  $j$  y  $k$ . De esta manera, no se forman ciclos si no existe una ruta entre los sucesores de  $\epsilon$  (excepto por el recurso en el que este asignado) y  $j$ , y entre  $k$  y los predecesores de  $\epsilon$  (excepto por el recurso en el que este asignado).

Al cumplirse la anterior condición lleva a la siguiente proposición matemática observada en la investigación realizada por García, Dauzère & Yazid (2015):

**Proposición 1.** La reasignación del recurso  $L$  a  $L'$  entre operaciones  $j$  y  $k$  es factible para una operación  $\epsilon$ ,  $\epsilon = x$  ó  $\epsilon = y$ . Además, no afecta a  $q_k^i$ , si:

1.  $l_j \leq \min_{b \in B} \{l_b\}$
2.  $l_k \geq \max_{p \in P_\epsilon} \{l_p\}$ ,  $k \neq p$

**Demostración:** Para la primera condición, se establece el conjunto  $B = \{F_\epsilon - F_{s(\epsilon)}^L\}$  Si  $j$  tiene menor número de arcos jamás podrá haber una ruta entre  $b$  y  $j$ . Respecto la segunda condición si  $k \neq p$  y tiene más arcos, tampoco existirá una ruta.

## 2.2 Estimaciones de tiempos de finalización de pedidos

Para estimar los tiempos de finalización de pedidos  $\hat{C}_i \forall i \in J$ , se debe asegurar que esta estimación cumpla las condiciones de un límite inferior "lower bound". Es decir, el valor

estimado debe ser menor igual que el real. Para los dos tipos de movimientos se han propuesto conjuntos de ecuaciones (Torres Tapia, 2018).

### 2.2.1 Estimación del movimiento al invertir (x, y)

Las transformaciones que se desarrollan en el grafo conjuntivo al invertir la ruta (x, y) están puntualizadas en la Figura 4. La Ecuación 1 representa la estructura para establecer el valor de la estimación  $\hat{C}_i \forall i \in J$ .

$$\hat{C}_i = \begin{cases} C_i & \text{si } q_c^i = q_k^i = -\infty \\ \max\{L_1, L_2\} & \end{cases}$$

Con

$$\begin{aligned} L_1 &= h_{P_{xx}} + p_{P_{xx}} + p_{F_{xx}} + q_{F_{xx}} \\ L_2 &= \hat{h}_x + p'_x + \max\{p_k + q_k, p_{F_{rx}} + q_{F_{rx}}\} \end{aligned}$$

Donde  $p'_x$  es el tiempo de procesamiento de x luego del movimiento y

$$\hat{h}_x = \max\{h_j + p_j, h_{P_{rx}} + p_{rx}\}$$

**Ecuación 1.** Estimación de  $C_i$  al invertir un arco crítico (x, y).

**Fuente:** (Mati, Dauzère-Pérès, & Lahlou, 2011).

Como se puede visualizar en la Ecuación 1 dos condiciones afectan la estimación para la finalización de  $J_i$ . La primera se relaciona con la inexistencia de una ruta entre la operación x y  $\phi_i$ ; la cual, en este problema la finalización seguirá siendo la misma. En caso de que exista esta distancia, se obliga a evaluar tres longitudes representadas por  $L_1$ ,  $L_2$  y  $L_3$ .

$L_1$  afecta las rutas que se originan desde y sin considerar al nodo x, ni los recursos que son críticos  $k \in K$ , los cuales estarán conectados a x. En  $L_2$  se analizan las rutas que se transforman al invertir a x; es decir, todos los sucesores de x sin y más los recursos críticos desde los cuales había ruta inmediata desde y. Finalmente,  $L_3$  es el que se evalúan los nodos de todo el grafo que tienen igual número de arcos de x. Adicionalmente, se han considerado las estimaciones de los tiempos de inicio de x ( $h_x$ ) y de y ( $h_y$ ) (García León, Dauzère Pérès, & Yazid, 2015).

### 2.2.2 Estimación de reasignar el recurso L de $\epsilon$ a $L'$ entre j y k

La Ecuación 2 representa la reasignación del recurso L de  $\epsilon$  a  $L'$  entre j y k.

$$\hat{C}_i = \begin{cases} C_i & \text{if } q_\epsilon^i = q_k^i = -\infty \\ \max\{L_1, L_2\} & \text{En caso contario} \end{cases}$$

$$L_1 = h_{\mathcal{P}_{s(\epsilon)}^l} + p_{\mathcal{P}_{s(\epsilon)}^l} + p_{\mathcal{F}_{s(\epsilon)}^l} + q_{\mathcal{F}_{s(\epsilon)}^l}^i$$

$$L_2 = \hat{h}_\epsilon + p'_\epsilon + \max\{p_k + q_k^i, \max_{b \in B} \{p_b + q_b^i\}\}$$

$$\hat{h}_\epsilon = \max\{\hat{h}_j + p_j, \max_{c \in C} \{h_c + p_c\}\}$$

$$\hat{h}_j = \begin{cases} \max_{p \in \{\mathcal{P}_{j-\epsilon}\}} \{h_p + p_p\} & \text{if } j = \mathcal{F}_{s(\epsilon)}^L \\ h_j & \text{En caso contario} \end{cases}$$

$$B = \{\mathcal{F}_\epsilon - \mathcal{F}_{s(\epsilon)}^L\}$$

$$C = \{\mathcal{P}_\epsilon - \mathcal{P}_{s(\epsilon)}^L\}$$

**Ecuación 2.** Estimación de  $C_i$  al reasignar  $L$  de  $\epsilon$  a  $L'$  entre  $j$  y  $k$ .

**Fuente:** (Mati, Dauzère-Pérès, & Lahlou, 2011).

Dos condiciones en la Ecuación 2 afectan la estimación para la finalización de  $J_i$ . La primera se relaciona con la inexistencia de una ruta entre la operación  $x$  y  $K$  a  $\phi_i$  que en este problema la finalización seguirá siendo la misma. En caso de que exista distancia, se deben evaluar dos longitudes representadas por  $L_1$  y  $L_2$ .

La primera  $L_1$ , simboliza la ruta que juntará al predecesor y sucesor de  $\epsilon$  una vez sea reasignado.  $L_2$  analiza la ruta nueva que unirá a  $j$  con  $\epsilon$  y las que se evalúan al considerar a  $k$  y los sucesores en secuencia de recursos diferentes de  $L$  para  $\epsilon$ . Cabe mencionar, que la consideración de  $L_3$  resultaría ineficiente ya que el tiempo de procesamiento de  $\epsilon$  puede minimizar lo cual no cambia las propiedades del conjunto que contiene las operaciones con el mismo número de arcos (Torres Tapia, 2018).

### 2.3 Desarrollo de la heurística

Como objetivo principal la heurística deberá llegar a una solución óptima, en tiempos parcialmente cortos de computación. Se alcanzará el desarrollo al considerar una solución inicial y a partir de esta se identifica la(s) rutas críticas de los pedidos que afectan un criterio; por ejemplo, en la tardanza máxima del trabajo (Makespan) la ruta que lleve el pedido de mayor duración. En la(s) rutas críticas que se hallen, se establecen los arcos críticos, los cuales son los insumos para aplicar las Ecuaciones 1 y 2 considerando la Proposición de factibilidad. El resultado de esta heurística permite determinar las combinaciones o cambios factibles con los valores estimados. Finalmente, la heurística transformará y guardará el grafo el cual contenga la mejor solución la cual contiene el menor valor del criterio inicial.

Según Torres Tapia (2018), la heurística se desarrolla en dos etapas iterativas denominadas mejoramiento y diversificación. En mejoramiento se busca la mejor

estimación que mejore el criterio para ir transformando el grafo. Cuando no es posible mejorar el valor del criterio, empieza la diversificación la cual se realizan movimientos aleatorios en un intervalo  $[a, b]$  para continuar con una nueva etapa de mejoramiento y en caso de que se mejore el criterio se retorna a la etapa de mejoramiento. El proceso de ejecución está dado en tiempo y en caso de que se alcance un valor de criterio de cero, la heurística finaliza el proceso. Este se da en el caso de que la tardanza máxima sea cero.

En la culminación de la heurística, el resultado computacional arroja dos archivos de texto, el primer archivo corresponde a la solución inicial, y el segundo la mejor solución alcanzada. Seguidamente, se detallará el proceso para alcanzar la solución inicial.

## **2.4 Solución inicial para Job-shop multi-recurso con rutas lineales - JSFMRL**

En el capítulo 1, descripción del problema Job-Shop Multi-recurso con rutas lineales - *JSFMRL* se mencionaron dos procedimientos (asignación y secuenciación). En esta sección se describirán a fondo estos dos procedimientos.

El primer procedimiento, es la asignación de máquinas a las operaciones. La asignación consiste en seleccionar, para cada operación, la(s) máquina(s) que realizará(n) la operación. El segundo procedimiento, es secuenciación trata de obtener el orden de las operaciones en cada una de las máquinas seleccionadas. Cuando se hayan asignado las máquinas en cada operación, se determina el tiempo de inicio de la operación al evaluar el máximo tiempo de finalización de cada uno de sus predecesores, los que tenga por recursos (máquinas) más el predecesor por ruta (pedido). Una vez que se ha establecido el tiempo de inicio, se delinean arcos conjuntivos desde todos sus predecesores hacia la operación. Finalmente se calcula el tiempo de finalización de la operación al adicionar el tiempo de operación (procesamiento de la operación), el cual es definido por el mayor tiempo de máquina asignado. En cada operación se asignan las maquinas según el menor tiempo de procesamiento de cada sub-conjunto y teniendo en cuenta de que estas no se repitan.

Este procedimiento se detalla a continuación:

1. Los  $n$  pedidos son ordenados según su número de identificación de manera ascendente. Es decir, pedido 1, pedido 2, ... pedido  $n$ -ésimo. De igual manera, las operaciones se ordenan con un número de identificación de manera ascendente de izquierda a derecha. Es decir, operación 1, operación 2, ... operación  $n$ -ésima.

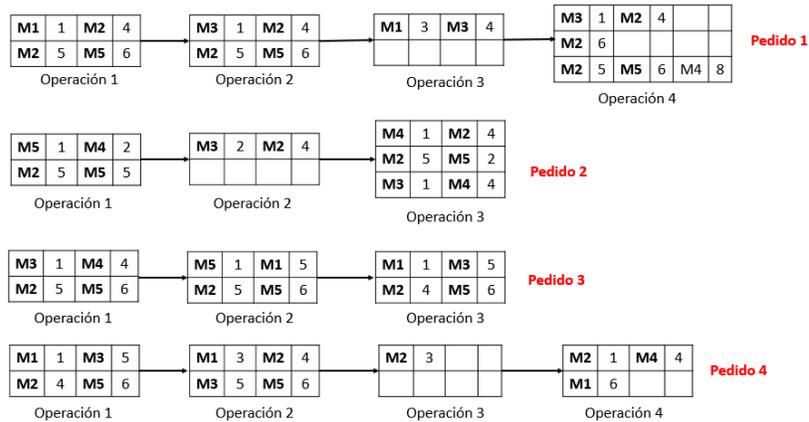


Figura 6. Grafo de la instancia propuesta.

Fuente: Autor.

- En cada pedido, se recorre la ruta de operaciones y se va asignando a las operaciones las máquinas con menor tiempo de procesamiento. Es importante, resaltar el caso de que en la operación solo exista una máquina para realizar la operación, la heurística asigna a la operación la única máquina existente.

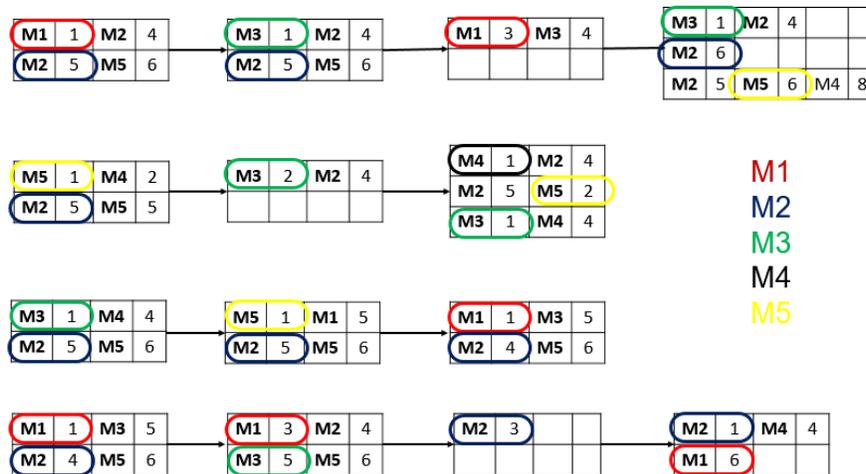


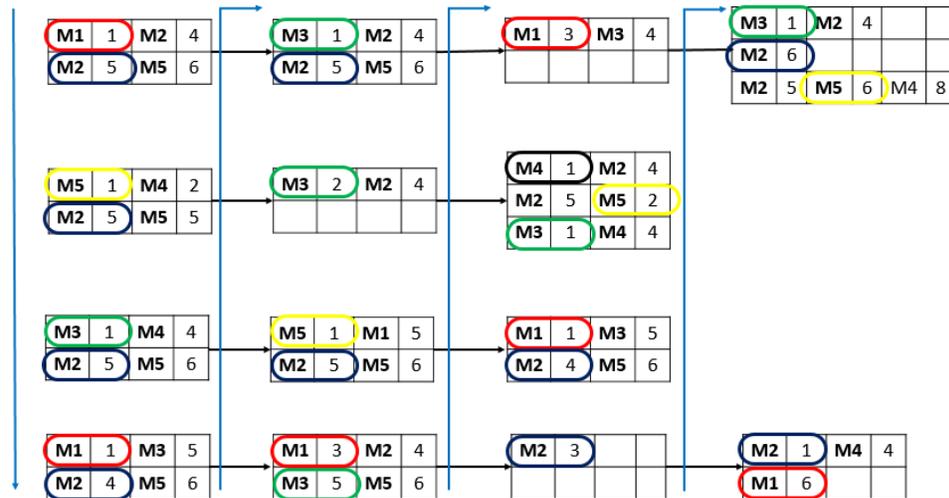
Figura 7. Asignación de máquinas a las operaciones.

Fuente: Autor.

Una vez asignadas todas las operaciones, el procedimiento de secuenciación es el siguiente:

- Se fija el punto de partida de todo el trabajo (Grafo), para este caso, el punto de partida es la primera operación del primer pedido (Ope 1.1). Luego se empieza

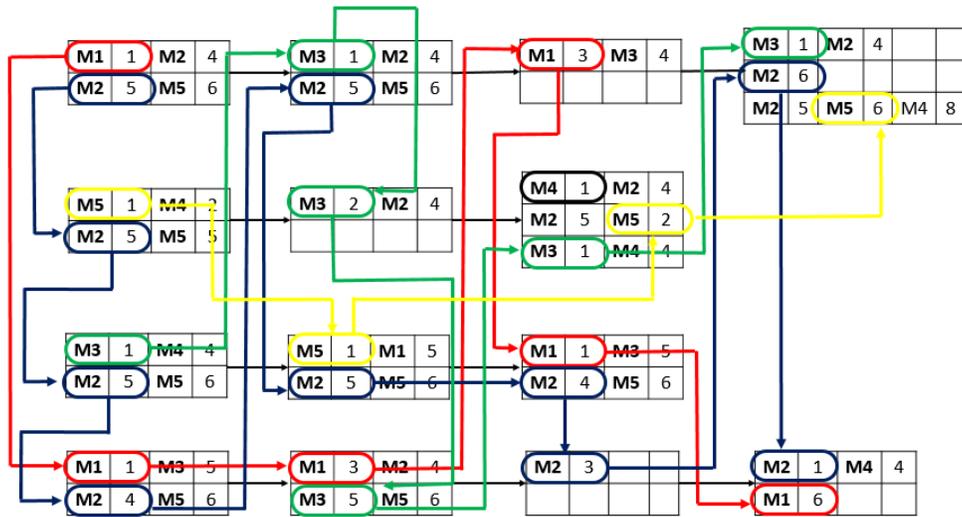
hacer un recorrido ascendente de pedidos, es decir, se recorren primero todas las primeras operaciones de todos los pedidos, y así sucesivamente, hasta finalizar con la última operación del último pedido.



**Figura 8.** Secuenciación para la instancia propuesta.

**Fuente:** Autor.

2. Los arcos conjuntivos se van agregando en la medida en que se van secuenciando las operaciones en las máquinas y vale la pena resaltar que el tiempo de inicio para una operación será el máximo entre la finalización de su predecesor inmediato por máquina y sobre la ruta del pedido.



**Figura 9.** Trazado de arcos conjuntos para la secuencia.

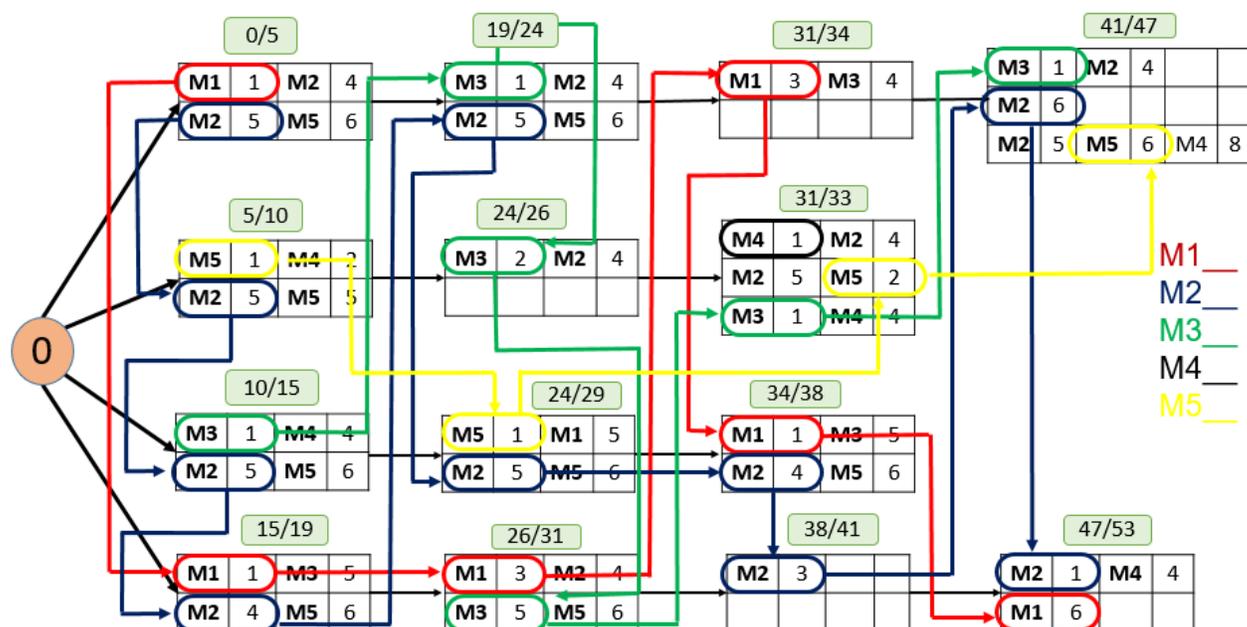
**Fuente:** Autor.

### 3. Capítulo 3. Resultados

Se ha detallado en los anteriores capítulos (Capítulos 1 y 2) el soporte matemático y teórico que sobrelleva el diseño de la heurística. En este capítulo, se procede a corroborar los resultados de los objetivos propuestos. La heurística fue desarrollada en el software MATLAB R2014a.

#### 3.1 Solución Inicial para una instancia propuesta

Para el problema Job-Shop Multi-recurso con rutas lineales - JSFMRL se tienen soluciones iniciales referentes para la minimización de la tardanza total del grafo (makespan). Por este motivo, se ha instaurado una instancia de 4 pedidos y 5 máquinas; y se ha aplicado el algoritmo de solución inicial estipulado en la Sección 2.4 y su solución inicial se visualiza en la Figura 10.



**Figura 10.** Solución inicial para una instancia propuesta.

**Fuente:** Autor.

Para la solución inicial propuesta, la primera operación asignada es la 1-1; es decir, la primera operación del primer pedido, la cual requiere de dos máquinas (recursos) y en el primero se selecciona M1 con un tiempo de procesamiento de una unidad (1) para el segundo recurso, se asigna M2 con un tiempo de procesamiento de 5. Esto indica que la

operación empieza en el tiempo 0 y termina 5 unidades después. Al finalizar la instancia propuesta, el valor de la tardanza total del trabajo (makespan) es 53.

Al calcular la tardanza total del grafo (makespan), se traza una ruta que inicia en la primera operación de la secuencia anteriormente mencionada, hasta el final de esta misma. Esta ruta es denominada "Ruta Crítica" y se puede visualizar en la Figura 11 con arcos sombreados. Como fue descrita en la sección 1.2 Grafo para el problema Job-Shop Multi-recurso con rutas lineales – JSFMRL.

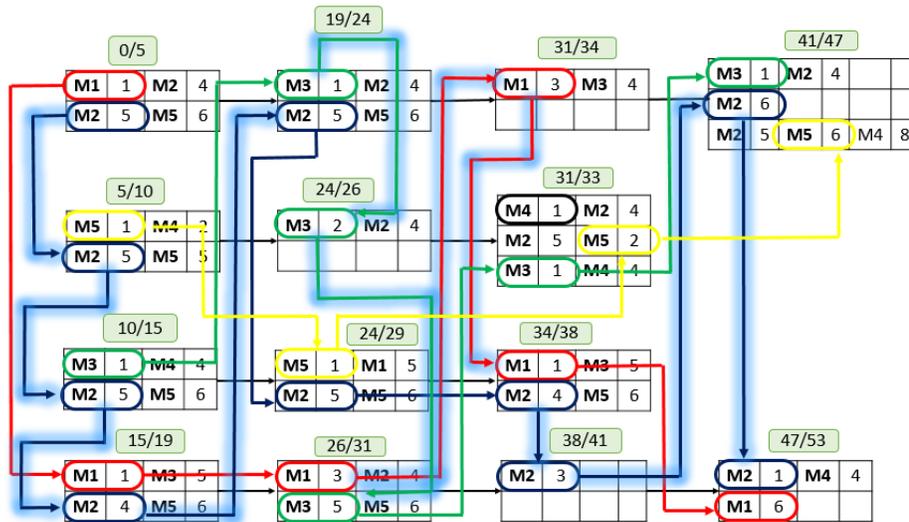
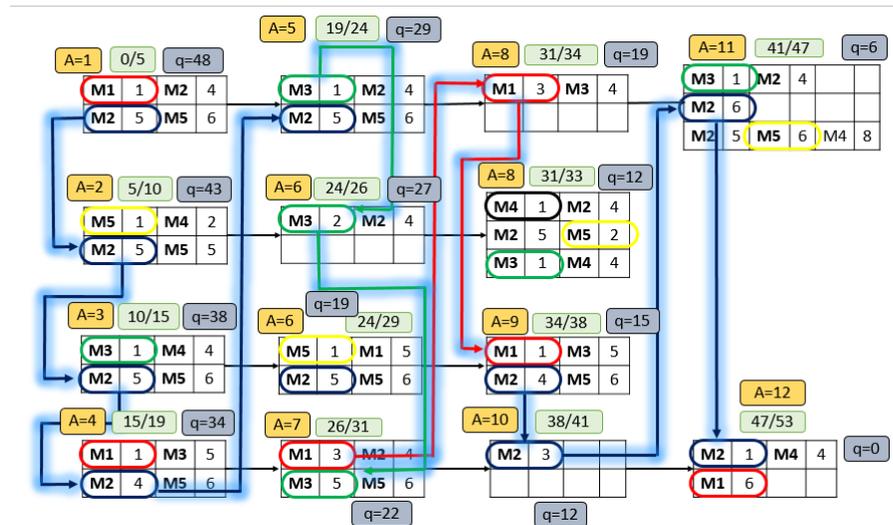


Figura 11. Ruta crítica para la instancia propuesta.

Fuente: Autor.

A partir de la ruta crítica, se calculan los arcos (A) y las colas máximas (q) de cada operación. Este procedimiento fue expuesto anteriormente en la sección 1.2 Grafo para el problema Job-Shop Multi-recurso con rutas lineales - JSFMRL.



**Figura 12.** Arcos y colas máximas para la instancia propuesta.

**Fuente:** Autor.

Seguidamente, fueron calculadas todas las soluciones iniciales basadas en la literatura de makespan y se registran en la Tabla 1 junto con la mejor solución de la heurística para cada instancia, las soluciones iniciales que se tienen de referencia (columna *Makespan*) y la que se obtiene aplicando la heurística (columna *Heurística*). Del presente análisis, se puede inferir que, con la realización de la heurística las soluciones iniciales (de referencia) han disminuido considerablemente.

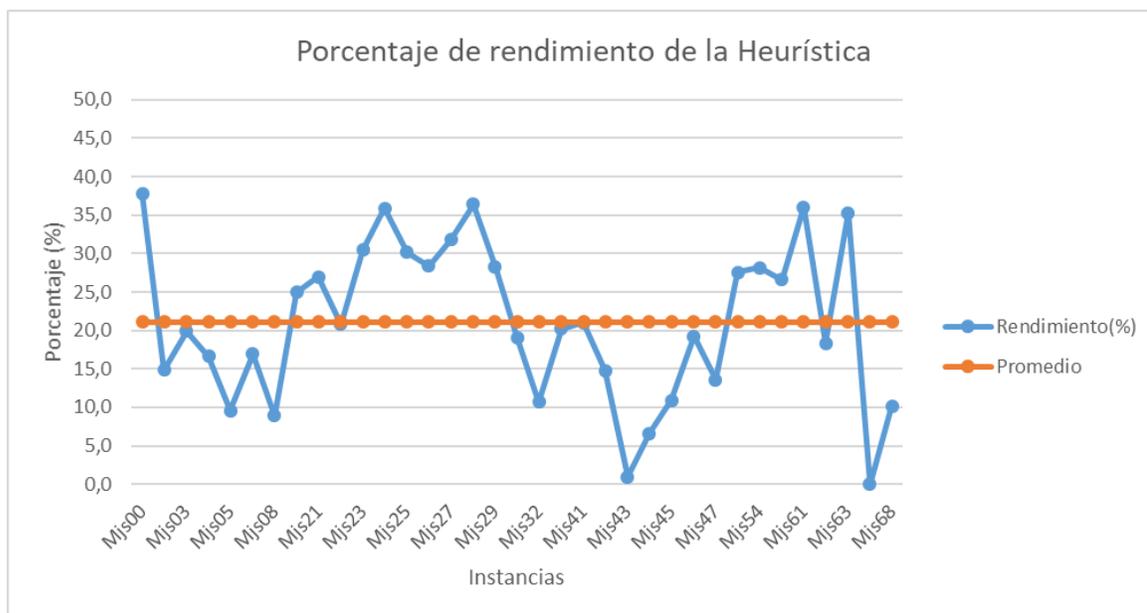
Esta disminución se puede concluir debido a que de las 35 instancias seleccionadas aleatoriamente de las 68 tomadas inicialmente de la investigación “Multi-resource Shop Scheduling with resources flexibility” realizada por Dautère, Roux & Lasserre (1998), 33 instancias tuvieron resultados satisfactorios y solamente 2 de ellas no tuvieron buen porcentaje de disminución (Mjs43 y Mjs64), como se evidencia en la tabla 1.

**Tabla 1.** Comparación entre la solución inicial de referencia y la heurística.

Instancia	Makespan	Heurística	Rendimiento (%)	Instancia	Makespan	Heurística	Rendimiento (%)
Mjs00	53	33	37,7	Mjs32	394	352	10,7
Mjs02	654	557	14,8	Mjs33	396	316	20,2
Mjs03	563	451	19,9	Mjs41	452	357	21,0
Mjs04	662	552	16,6	Mjs42	434	370	14,7
Mjs05	1030	931	9,6	Mjs43	317	314	0,9
Mjs06	921	765	16,9	Mjs44	423	395	6,6
Mjs08	1058	964	8,9	Mjs45	542	483	10,9
Mjs20	1923	1443	25,0	Mjs46	603	487	19,2
Mjs21	1647	1203	27,0	Mjs47	692	598	13,6
Mjs22	1819	1441	20,8	Mjs52	817	592	27,5
Mjs23	1918	1334	30,4	Mjs54	866	622	28,2
Mjs24	1833	1176	35,8	Mjs60	695	510	26,6
Mjs25	2927	2045	30,1	Mjs61	882	564	36,1
Mjs26	2258	1618	28,3	Mjs62	676	552	18,3
Mjs27	2783	1898	31,8	Mjs63	802	519	35,3
Mjs28	2788	1774	36,4	Mjs64	533	533	0,0
Mjs29	2435	1746	28,3	Mjs68	1054	947	10,2
Mjs31	409	331	19,1	-	-	-	-

**Fuente:** Autor.

De igual manera, en la tabla 1 fue calculado el rendimiento de la heurística para cada instancia seleccionada. El mayor rendimiento en la tabla es de 37.7%, siendo este el de la instancia Mjs00; y el peor rendimiento fue de 0% en la instancia Mjs64. A continuación, son ilustrados en la gráfica 1 los porcentajes de rendimiento de cada una de las instancias, y su debido promedio (21.1%).



**Gráfica 1.** Porcentaje de rendimiento de las instancias y de la heurística.

**Fuente:** Autor.

### 3.2 Optimización del makespan

A continuación, se ilustra la eficiencia de la heurística al minimizar el makespan de la instancia que se propuso en la sección 3.1 Solución Inicial, y que se referencio en la Figura 6, en donde el makespan presentaba un valor de 53 unidades. Aplicando la heurística, se alcanza un valor de 33 cuya solución es ilustrada en la Figura 13 en un tiempo de 5 minutos de ejecución computacional y la pantalla del archivo txt que ilustra la solución generada por el software Matlab R2014a.

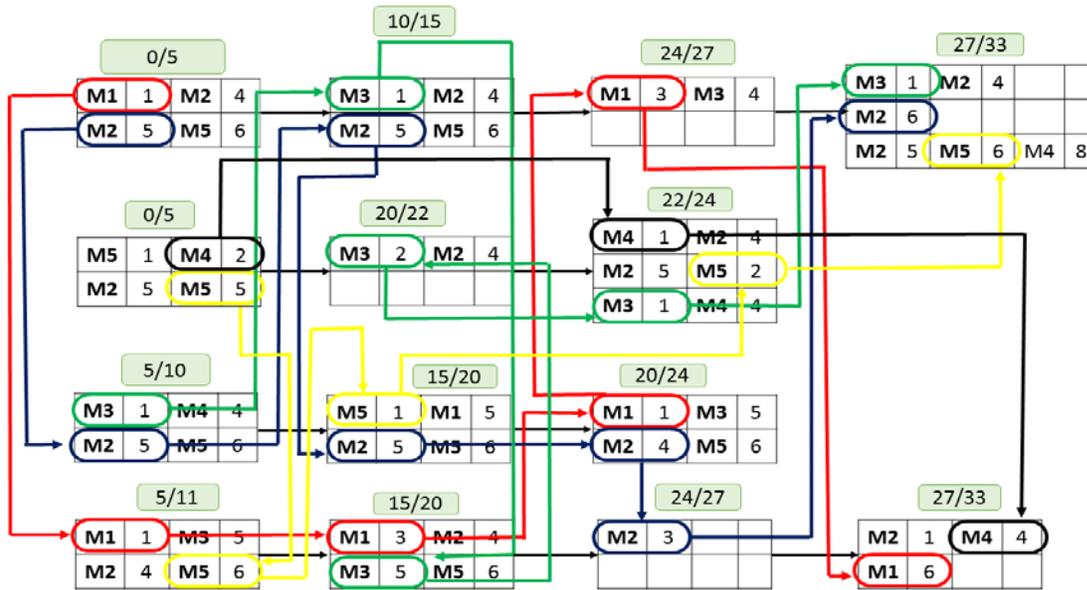


Figura 13. Mejor solución encontrada para una instancia propuesta.

Fuente: Autor.

```

Editor - C:\Users\User\Documents\MATLAB\TheBest.txt
present.m  metodoReasignacion.m  metodoResecuenciacion.m  solutionGrafo.m  transformarGrafo.m  writer.m  TheBest.txt  +
1 makespan: 33
2
3
4 Machine 1: (1, 1)->(0, 5)|| (4, 1)->(5, 11)|| (4, 2)->(15, 20)|| (3, 3)->(20, 24)|| (1, 3)->(24, 27)|| (4, 4)->(27, 33)||
5 Machine 2: (1, 1)->(0, 5)|| (3, 1)->(5, 10)|| (1, 2)->(10, 15)|| (3, 2)->(15, 20)|| (3, 3)->(20, 24)|| (4, 3)->(24, 27)|| (1, 4)->(27, 33)||
6 Machine 3: (3, 1)->(5, 10)|| (1, 2)->(10, 15)|| (4, 2)->(15, 20)|| (2, 2)->(20, 22)|| (2, 3)->(22, 24)|| (1, 4)->(27, 33)||
7 Machine 4: (2, 1)->(0, 5)|| (2, 3)->(22, 24)|| (4, 4)->(27, 33)||
8 Machine 5: (2, 1)->(0, 5)|| (4, 1)->(5, 11)|| (3, 2)->(15, 20)|| (2, 3)->(22, 24)|| (1, 4)->(27, 33)||
9
Command Window
best =
33
    
```

Figura 14. Pantalla de la mejor solución generada por el software.

Fuente: Autor.

---

## 4. Conclusiones y recomendaciones

### 4.1 Conclusiones

La presente asistencia de investigación ha diseñado una heurística para contribuir a dar soluciones óptimas e innovadoras a los problemas de Scheduling Job-Shop Multi-recurso con rutas lineales – FSJMRL, minimizando la tardanza total de un trabajo (Makespan). Esta heurística se representa como una herramienta computacional en el software de Matlab R2014a, en el cual, se ingresa un archivo de texto (extensión txt) con toda la información de un trabajo, es leído y procesado por el mismo para la determinación de movimientos factibles en operaciones críticas. La heurística genera al final un archivo de texto (extensión txt) con la solución optimizada del problema. Este archivo indica la forma en que debe ser secuenciada y asignada cada máquina del problema para optimizar la entrega del trabajo y satisfacer las necesidades del servicio al cliente. La heurística puede ser requerida en operaciones de la industria colombiana, las cuales requieran la utilización de máquinas simultáneas en procesos con configuraciones intermitentes. De igual manera, el diseño de esta herramienta satisface y/o alcanza los tres objetivos específicos planteados para la investigación, que a continuación se mencionan:

El objetivo 1 buscó formular un algoritmo para determinar una solución inicial óptima y de calidad para el criterio del makespan. Este algoritmo considera que el orden de los pedidos de un trabajo debe considerarse según las fechas de entrega; de igual manera, realizar una asignación apropiada de máquinas a las operaciones, las cuales serían las de menor tiempo de procesamiento.

El objetivo 2 buscó establecer funciones de estimación para el movimiento de operaciones críticas considerando el anterior criterio mencionado (makespan). Para el logro de este objetivo se debe extender a las contribuciones en el problema de Scheduling Job-Shop flexible al problema objeto de estudio, al considerar la inversión de arcos críticos (resecuenciación) y el número máximo de arcos en la reasignación sin perjudicar los parámetros en el problema. Se puede evidenciar este resultado con una proposición matemática y dos ecuaciones, descritas en el capítulo 2.

El objetivo 3 buscó validar las etapas de mejoramiento y diversificación en el proceso de búsqueda local al minimizar el criterio de makespan. La consecución de este se valida al resolver las instancias para el criterio. Finalmente se coopera al enfoque con la determinación de los tiempos de entrega, que para un futuro se pueden corroborar en un ambiente industrial.

Estos objetivos fueron alcanzados óptimamente y se puede apreciar, en la minimización del criterio makespan en la instancia propuesta para dar explicación a la heurística diseñada.

## 4.2 Recomendaciones

Como recomendaciones para emprender nuevas investigaciones, sería adecuado desarrollar lo siguiente:

1. Para mejorar el desempeño de la heurística se debe mejorar los procesos de diversificación.
2. Identificar los procesos productivos en la región (Tolima), en los cuales, se pueda aplicar o ejecutar ajustes para su respectivo mejoramiento de eficiencia.
3. Ampliar enfoques para los modelos de optimización multi-objetivos.
4. Para la consecución de nuevos enfoques investigar e indagar rutas no lineales en los procesos.

## **A.Anexo: Archivos de las mejores soluciones de la heurística.**

En la presente investigación, se anexan los archivos de las mejores soluciones de la heurística en algunas instancias, arrojadas por el software de Matlab R2014a como archivos de extensión txt.

Los archivos adjuntos son de las siguientes instancias: Mjs21, Mjs24, Mjs28, Mjs61 y Mjs63. Estos archivos se adjuntarán en una carpeta, junto con la asistencia de investigación. Cada archivo arrojado por el software se denomina como “TheBest”, y seguidamente el nombre de la instancia. Por ejemplo, la mejor solución de la instancia Mjs21 es TheBest mjs21.

## Referencias bibliográficas

Brucker. (2007). *Scheduling Algorithms*. Springer.

Brucker, P.; Schlie, R. (1990). Job-shop scheduling with multi-purpose machines. *Computing*, 45(4), 369-375.

Cheng, W.; Zhang, R. (2011). A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective. *Computers & Operations Research*, 38(5), 854-867.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C.;. (2001). *Introduction to algorithms* (Vol. 2). MIT press Cambridge.

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

Dauzère-Pérès, S. R. (1998). Multi-resource shop scheduling with resource Flexibility. *European Journal of Operational Research*(107), 289-305.

Deb, K. (1989). Genetic Algorithms in Multi-Modal Function Optimization. Master's Thesis. Tuscaloosa: AL; University of Alabama.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. WILEY.

Deb, k. et. al. (2001). *Multi-objective optimization using evolutionary algorithms*. John Wiley and Sons Chichester.

Deb, K., Pratap, A., Agarwal, S., & T. , M. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. In IEEE (Ed.).

Deb, K.; Agrawal, S.; Pratap, A; and Meyarivan, T.;. (2000a). A fast and elitist multi-objective genetic algorithm: NSGA\_II. *Technical Report 200001*. Indian Institute of Technology, Kanpur: Kanpur Genetic Algorithms laboratory(KanGAL).

Ehrgott, M. (1997). *Multiple criteria optimization: classification and methodology*. Shaker.

Fattahi, P. (2009). *A Hybrid Multi Objective Algorithm for Flexible Job Shop Scheduling*. Proceedings of World Academy of Science.

---

Fonseca, C. M. Fleming, P. J. (1996). *On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers*. Proceeding PPSN IV Proceedings of the 4th International Conference on Parallel Problem Solving from Nature.

García León, A., Dauzère Pérès, S., & Yazid, M. (2015). Minimizing regular criteria in the Flexible Job-shop scheduling problem. *Multidisciplinary International scheduling Conference: Theory & Applications*, (págs. 443-456). Prague.

Genova, K., & Guliashki, V. (2015). A survey of solving approaches for multiple objective flexible job shop scheduling problems. *Cybernetics and Information Technologies*, 15(2), 3-22.

Hansen, M. P. Jaszkiwicz, A. (1998). Evaluating the quality of approximations to the non-dominated set. *IMM Technical Report IMM-REP-1998-7*.

Hurink, J.; Jurisch, B.; Thole, M. (1995). Tabu search for the job-shop scheduling problem with multi-purpose machines. *OR Spectrum*, 15(4), 205-215.

Kacem, I. Hammadi, S. and Borne, P. (2002). *Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic* (Vol. 60). (M. a. Simulation, Ed.)

Knopp, S. Y. (2017). A batch-oblivious approach for Complex Job-Shop scheduling problems. *European Journal of Operational Research*, 263, 50-61.

Knowles, J. D. Corne, D. W. (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Journal Evolutionary Computation Volume 8 Issue 2*, 149-172.

Liu, H. Abraham, A. Choi, O. and Moon, S. H. (2006). *Variable Neighborhood Particle Swarm Optimization for Multi-objective Flexible Job-Shop Scheduling Problems* (Vol. 4247). Springer.

Mati, Y., Dauzère-Pérès, S., & Lahlou, C. (2011). A general approach for optimizing regular criteria in the job-shop scheduling problem. *European Journal of Operational Research*, 212(1), 33-42.

Mati, Y.; Xie, X. (2004). The complexity of two-job shop problems with multi-purpose unrelated machines. *European Journal of Operational Research*, 152(1), 159-169.

qing Li, J. ke Pan, Q. and Liang, Y.-C. (2010). *An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems* (4 ed., Vol. 54).

Roy, B. Sussmann, B. (1964). Les problemes d'ordonnancement avec contraintes disjonctives. Note ds,9.

Schott, J. R. (1995). *Fault tolerant design using single and multicriteria genetic algorithm optimization*. Boston, Massachusetts Institute of Technology. Dept. of Aeronautics and Astronautics: Massachusetts Institute of Technology.

Singer, M. P. (1998). A computational study of branch and bound techniques for minimizing the total weighted tardiness in job-shops. (IIE, Ed.) *IIE Transactions*, 30(2), 109-118.

T' Kindt, V., & Billaut, J. C. (2006). *Multicriteria scheduling: theory, models and algorithms*. Springer Science & Business Media.

Torres Tapia, W. F. (2018). *Formulación y validación de una heurística para minimizar la tardanza máxima en el problema de Scheduling Job-shop flexible multirecurso con rutas lineales*. Ibagué - Tolima.: Universidad de Ibagué (Trabajo de grado).

Veldhuizen, D. A. (1999, junio). Multiobjective Evolutionary Algorithms: Classifications. Ph. D. thesis, Graduate School of Engineering of the Air Force Institute of Technology, Air University.

Vilcot, G. (2007). Algorithmes approchés pour des problèmes d'ordonnancement multicritères de type job shop flexible et job shop multiresource. Autre [cs.OH]. Université François Rabelais - Tours, 2007. Français.

Vilcot, G. and Billaut, J.-C. (2008). A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem. *European Journal of Operational Research*, 398-411.

Vilcot, G. and Billaut, J.-C. (2011). A tabu search algorithm for solving a multicriteria flexible job shop scheduling problem. *International Journal of Production Research*(49(23)), 6963–6980.

Zitzler, E. (1999). Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Switzerland: Ph. D. thesis, Swiss Federal Institute of Technology (ETH) Zurich.

Zitzler, E. a. (1998b). multi objective optimization using evolutionary algorithms. *A comparative case study. In parallel problem solving from Nature V (PPSN-V)*, 292-301.