

Nota de Aceptación:

Proyecto de Diseño Aprobado. en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana Cali para optar el título de Ingeniero Industrial.

Camilo Roche.

HERNÁN CAMILO ROCHA NIÑO
Decano de la Facultad de Ingeniería y Ciencias

Jorge Enrique Álvarez Patiño

JORGE ENRIQUE ÁLVAREZ PATIÑO
Director Carrera Ingeniería Industrial

Daniel Morillo Torres

DANIÉL MORILLO TORRES
Director(a) Proyecto de Diseño

Álvaro Figueroa Cabrera

ÁLVARO FIGUEROA CABRERA
Jurado 1

Hector Fabio Bonilla Londoño

HECTOR FABIO BONILLA LONDOÑO
Jurado 2

Diciembre 10 de 2020, Santiago de Cali

PROYECTO DE DISEÑO II

VALENTINA ARCINIEGAS CORTÉS

JHOAN ARLEY DUQUE OTABO

SARA LUCÍA LONDOÑO BARRETO

JUAN ESTEBAN MARÍN CARVAJAL

DIRECTOR

DANIEL MORILLO TORRES

Para el curso de PROYECTO DE DISEÑO II se realizó la revisión de la tercera entrega quedando validada y sometida a correcciones. El documento hasta la fecha contiene las recomendaciones suministradas por el director del proyecto DANIEL MORILLO TORRES.

Muchas gracias.

A handwritten signature in black ink, appearing to read 'Morillo', with a stylized flourish extending to the right.

DIRECTOR DEL PROYECTO



Diseño de un Método de Solución para el Flexible Job Shop con Tiempos de Alistamiento Dependientes de la Secuencia.

Valentina Arciniegas Cortés^{a,c}, Jhoan Arley Duque Otabo^{a,c} Sara Lucía Londoño Barreto^{a,c}, Juan Esteban Marín Carvajal^{a,c}

Daniel Morillo Torres^{b,c}

^aEstudiante de Ingeniería Industrial

^bProfesor, Director del Proyecto de Grado, Departamento de Ingeniería Civil e Industrial

^cPontificia Universidad Javeriana, Cali, Colombia

Resumen en español

El Flexible Job Shop con Tiempos de Alistamiento Dependientes de la Secuencia (SDST-FJSP) es, al igual que el *Job Shop Scheduling Problem* (JSP), un problema de programación de trabajos. Sin embargo, el SDST-FJSP tiene en cuenta supuestos adicionales, tales como los tiempos de alistamiento dependientes de la secuencia y la multifuncionalidad de las máquinas, que permiten darle mayor cercanía a la realidad. Este problema ha sido catalogado como *NP-hard*, lo que ha despertado gran interés entre los investigadores, pues su alta complejidad hace que un método de solución exacta no sea una propuesta viable debido a la gran magnitud de los tiempos computacionales. Así pues, en este trabajo se proponen dos métodos para abordar el SDST-FJSP. El primero, es un modelo matemático empleando MILP (*Mixed-integer Linear Programming*), si bien este es un método exacto, permite una comprensión más profunda del problema. El segundo, es un algoritmo genético (AG) en el cual se empleó el método *Tournament* para el operador de cruce y el método *Swap Mutation* para el operador de mutación. Posteriormente, se realiza una comparación del desempeño obtenido por cada uno de los métodos, encontrando, como era de esperarse, mejores resultados en el AG para un mismo tiempo de ejecución.

Palabras claves: Flexible Job Shop, Tiempos de alistamiento, Algoritmo genético, Modelo matemático.

Abstract

The Flexible Job Shop with Sequence-Dependent Setup Times (SDST-FJSP) is like the Job Shop Scheduling Problem (JSP), a job scheduling problem. However, the SDST-FJSP considers additional assumptions, such as sequence-dependent setup times and the multipurpose of the machines, which allow it to be closer to reality. This problem has been classified as NP-hard, which has aroused great interest among researchers, because its high complexity means that an exact solution method is not a viable proposal due to the great magnitude of computational times. Thus, in this work, two methods are proposed to address the SDST-FJSP. The first is a mathematical model using MILP (*Mixed Integer Linear Programming*), although this is an exact method, it allows a deeper understanding of the problem. The second is a genetic algorithm (GA) in which the Tournament method was used for the crossover operator and the Swap Mutation method for the mutation operator. Afterward, a comparison of the performance obtained by each of the methods is made, finding, as expected, better results in the GA for the same execution time.

TABLA DE CONTENIDO

I.	PROJECT CHARTER.....	4
II.	DEFINIR	6
	A. Contexto y Justificación	6
	B. Descripción formal del Flexible Job Shop Scheduling Problem con tiempos de alistamiento dependientes de la secuencia (SDST-FJSP)	9
	C. Grupos de interés.....	9
	C. Requerimientos.....	12

1) Restricciones de diseño (Factibilidad).....	12
2) Especificaciones de diseño (características).....	13
3) Leyes, normas y estándares.....	13
III. MEDIR.....	15
A. Plan de recolección de datos.....	16
B. Exploración del mercado.....	18
1) Casos aplicados.....	18
2) Análisis del entorno.....	20
3) Estado actual del sistema.....	20
IV. ANALIZAR.....	22
A. Análisis de Oportunidad.....	22
B. Revisión de literatura.....	24
C. Exploración de ideas y selección de alternativa.....	27
D. Objetivos.....	29
Objetivo general:.....	29
Objetivos específicos:.....	29
E. Plan de trabajo (PdT).....	29
V. DISEÑAR.....	31
A. Desarrollo del diseño de la solución.....	31
1) Modelo Matemático.....	31
1) Algoritmo genético.....	33
B. Validación del diseño propuesto.....	¡Error! Marcador no definido.
A. Medición de los impactos.....	42
B. Estandarización de la solución – POE’S (plan de control).....	46
C. Conclusiones.....	¡Error! Marcador no definido.
D. Recomendaciones.....	47

Índice de Tablas

TABLA I. PROJECT CHARTER.....	4
TABLA II. GRUPOS DE INTERÉS.....	10
TABLA III. REQUERIMIENTOS DE LEYES, NORMAS Y ESTÁNDARES DE LOS ACTORES DE INTERÉS.....	14
TABLA IV. PLAN DE RECOLECCIÓN DE DATOS.....	17
TABLA V. RESUMEN DE CASOS EXITOSOS PARA EL FLEXIBLE JOBSHOP.....	19
TABLA VI. ANÁLISIS DOFA/FODA.....	20
TABLA VII. INDICADORES DE DESEMPEÑO A MEDIR.....	21
TABLA VIII. RESUMEN DE KPI'S PARA SDST_HUDATA.....	21

TABLA IX. RESUMEN DE REGLAS DE DESPACHO	23
TABLA X. CLASIFICACIÓN DE LOS MÉTODOS DE SOLUCIÓN	24
TABLA XI. RESUMEN DE LITERATURA DEL FJSP.....	25
TABLA XII. CRITERIOS DE CALIFICACIÓN.....	27
TABLA XIII. PONDERACIONES DE LOS CRITERIOS DE DECISIÓN	28
TABLA XIV PONDERACIONES FINALES POR ALTERNATIVA.....	29
TABLA XV. PLAN DE TRABAJO.....	30
TABLA XVI.....	38
TABLA XVII VALIDACIÓN DISEÑO 24 MIN DE EJECUCIÓN	38
TABLA XVIII VARIACIÓN PROBABILIDAD DE MUTACIÓN.....	39
TABLA XIX. TABLA DE ANEXOS	53

Índice de Figuras

Fig. 1 Diagrama de Gantt para la regla de secuenciación FIFO. Fuente: Tomado de [4].	7
Fig. 2 Diagrama de Gantt empleando un algoritmo genético multi-objetivo. Fuente: Tomado de [4].	8
Fig. 3 Crecimiento del interés por parte de investigadores a lo largo del tiempo en el FJSP. Fuente: Tomado de [5].	8
Fig. 4 Investigación del FJSP por cada una de las áreas de conocimientos. Fuente: Tomado de [5]......	9
Fig. 5 Matriz de poder vs interés	11
Fig. 6 Modelo de instancia SDST_HUData.....	16
Fig. 7 Pseudocódigo del proceso de programación del ERD.....	18
Fig. 8 Diagrama de Gantt de la regla de despacho ERD.....	22
Fig. 9 Diagrama de Gantt de la Instancia La01 programada con regla de despacho ERD.....	23
Fig. 10 Diagrama de Gantt de la Instancia La01 programada con regla de despacho SPT.....	23
Fig. 11 Diagrama de Gantt de la Instancia La01 programada con regla de despacho LPT.	23
Fig. 12 Selección de candidatos como padres.....	36
Fig. 13 Ejemplo método de la ruleta selección de parejas.	36
Fig. 14 Operador de cruce para la selección de máquina. Fuente. tomado de [41].....	¡Error! Marcador no definido.
Fig. 15 Ejemplo operador de cruce con PPX.....	37
Fig. 16 Operador de mutación con Swap Mutation	37
Fig. 17. Resultados por instancia con una probabilidad de mutación del 5%.....	40
Fig. 18. Resultados por instancia con una probabilidad de mutación del 30%	41
Fig. 19. Resultados por instancia con una probabilidad de mutación del 30%	41
Fig. 20 Manual de Usuario.	47

I.PROJECT CHARTER

TABLA I.
PROJECT CHARTER

Descripción (<i>Business case</i>)		Planteamiento del problema (<i>Problem statement</i>)		
<p>Empresas de distintos sectores como textil, automotriz, metalmecánico, entre otros, se caracterizan por tener un sistema de producción tipo <i>Job Shop</i> (JSP), debido a su modalidad de trabajo bajo pedido. Además, la multifuncionalidad de las máquinas que se usan hace que la programación de la producción sea un problema tipo <i>Flexible Job Shop</i> (FJSP). En este proyecto se abordará este problema en su variante SDST-FJSP, donde se tienen en cuenta los tiempos de alistamiento dependientes de la secuencia. Este, al ser una extensión del JSP, se cataloga como <i>NP-Hard</i>, lo que implica que el tiempo computacional usado en encontrar la solución óptima del problema no se encuentra aún acotado de forma polinomial, de modo que conforme el tamaño de las instancias aumenta, el problema se vuelve intratable para los métodos de solución exactos. Por lo tanto, se hace fundamental la búsqueda de métodos alternativos más eficientes que encuentren soluciones cercanas al óptimo en tiempos computacionales aceptables para el usuario, de modo que, se mejore el desempeño de la compañía en factores como:</p> <ul style="list-style-type: none"> • Disminución de las demoras en las entregas. • Disminución del tiempo de ocio. 		<p>Este problema se presenta, especialmente, cuando los trabajos que se deben realizar son muy diversos en sus características de procesamiento y los tiempos de alistamiento dependen de la secuencia. Esta situación es descrita por el problema del <i>Flexible Job Shop Scheduling</i> (FJSP). Donde, cada trabajo se compone de un conjunto de actividades que debe ejecutarse en uno de varios modos distintos, es decir, se dispone de distintas máquinas que pueden realizar la actividad. Además, se tiene un tiempo de procesamiento asociado al modo de ejecución y un tiempo de alistamiento dependiente de la secuencia.</p> <p>Adicionalmente, se tendrá en consideración los siguientes supuestos:</p> <ul style="list-style-type: none"> • Los trabajos deben ser procesados sin interrupciones. • Cada máquina puede procesar máximo un trabajo a la vez. • Todos los trabajos están disponibles para ejecutarse en el tiempo 0. • Las máquinas operan independientes unas de otras. • Los trabajos son independientes unos de otros. • La recirculación está permitida. 		
Impacto de los actores (<i>Stakeholder's business needs</i>)		Restricciones	Especificaciones	Marco legal
<p>La implementación permitirá aumentar la productividad de distintas industrias como lo son las farmacéuticas, textiles, metalmecánicas, entre otras, en las que se caracterizan por la multifuncionalidad de las máquinas y la producción bajo pedido, mediante la adecuada programación de la producción y la reducción del tiempo que tarda dicha programación.</p>		<ul style="list-style-type: none"> • Tiempos de alistamiento. • Reglas de despacho. • Mantenimiento de las máquinas. • Lineamientos de la Universidad. • Búsqueda de información. • Horas laborales permitidas. 	<ul style="list-style-type: none"> • Disminuir el <i>makespan</i>. • Metodología DMADV y propuesta para Evaluación de Proyectos de Diseño enfocados en Investigación Aplicada con base en DMADV. • Fichas de trabajo. • Fuente de información confiable. 	<ul style="list-style-type: none"> • Normas establecidas por las empresas. • Normas IEEE. • Artículo 27.1. Decreto 486 del 2000 CAN. • Código sustantivo de trabajo. Artículo 160 (horas diurnas y nocturnas). Código sustantivo de trabajo. Artículo 167 (Tiempo de descanso). Código sustantivo de trabajo. Artículo 161 (Jornada máxima). Artículo adicionado por el artículo 22 de la Ley 50 de 1990 (Máximo horas extras).
Indicadores de Desempeño (<i>KPI's</i>)				
Variable	Actualidad	Meta		
<i>Makespan</i> (C_{max})	Para las instancias SDST_HUData con las reglas de despacho ERD y SPT el <i>makespan</i> promedio es de 1501,25 y 1669 respectivamente.	Disminución del <i>Makespan</i> en al menos un 10%.		
MRE (<i>Mean Relative Error</i>)	Para las instancias SDST_HUData con las reglas de despacho ERD y SPT el MRE promedio es de 82% y 104% respectivamente.	Disminución del MRE. El cual se espera que sea un valor tan cercano a cero como sea posible.		
Objetivo general (<i>Goal statement</i>)				

Diseñar un método de solución para el problema de *Flexible Job Shop* con tiempos de alistamientos dependientes en la secuencia, con el fin de disminuir el *Makespan*

Objetivos específicos (Project scope)

- Identificar los componentes de diseño y ejecución de la alternativa seleccionada mediante la revisión sistemática de literatura.
- Diseñar un modelo matemático o algoritmo heurístico o metaheurístico, que dé solución al problema SDST-FJSP.
- Validar el método de solución propuesto, a través de la medición de los indicadores de desempeño y compararlo con otros métodos en la literatura en base a las instancias SDST-HUdata.
- Estandarizar la implementación del método propuesto mediante la elaboración de un manual de usuario.

Plan de Trabajo (Project Plan)

Equipo de trabajo (Team members)

Actividad	Fecha Inicio	Fecha Fin	Área IISE	Nombre	Rol
Desarrollar un estudio sistemático de la alternativa seleccionada, categorizando sus diversos componentes	27/08/2020	04/09/2020	11. Information Engineering	Programadores ALPHA	Técnico de procedimientos Buscador de información Iniciador Coordinador
Describir cada componente, en base al problema de estudio FJSP	07/09/2020	15/09/2020	11. Information Engineering	Programadores ALPHA	Técnico de procedimientos Buscador de información Iniciador Coordinador
Sintetizar la información clave de la alternativa seleccionada, con sus componentes de diseño y ejecución	07/09/2020	11/09/2020	11. Information Engineering	Jhoan Duque, Valentina Arciniegas, Juan Esteban Marín.	Coordinador Buscador de información Técnico de procedimientos
Efectuar la formulación matemática del problema SDST-FJSP.	16/09/2020	13/10/2020	2. Operations research & analysis	Sara Londoño	Iniciador
Realizar experimentación del modelo matemático, identificando su desempeño de resolución.	14/10/2020	22/10/2020	2. Operations research & analysis	Sara Londoño	Iniciador
Desarrollar el diseño de la alternativa seleccionada de acuerdo con las condiciones y restricciones del sistema.	14/09/2020	09/10/2020	2. Operations research & analysis	Jhoan Duque, Valentina Arciniegas, Juan Esteban Marín.	Coordinador Buscador de información Técnico de procedimientos
Evaluar el rendimiento de la alternativa seleccionada en cada una de las instancias SDST-Hudata	12/10/2020	23/10/2020	11. Information Engineering	Jhoan Duque, Valentina Arciniegas, Juan Esteban Marín.	Coordinador Buscador de información Técnico de procedimientos
Comparar el desempeño de la alternativa seleccionada con diversos métodos de la literatura del FJSP	26/10/2020	02/11/2020	11. Information Engineering	Jhoan Duque, Valentina Arciniegas, Juan Esteban Marín.	Coordinador Buscador de información Técnico de procedimientos
Modificar el ingreso de las variables de entrada a través de un <i>dashboard</i>	03/11/2020	12/11/2020	11. Information Engineering	Juan Esteban Marín Jhoan Duque	Técnico de procedimientos Coordinador
Realizar una guía detallada para el ingreso de las variables y la interpretación de la solución	03/11/2020	18/11/2020	11. Information Engineering	Sara Londoño Valentina Arciniegas	Iniciador Buscador de información

II. DEFINIR

A. Contexto y Justificación

Muchas empresas de manufactura en algún momento se han preguntado cómo es la mejor forma de utilizar los recursos que poseen, ya que generalmente, estos son limitados. Algunos de estos recursos pueden ser: el personal, la maquinaria, las pistas de aterrizaje en un aeropuerto, las unidades computacionales, entre otros [1]. Este problema se presenta, especialmente, cuando los trabajos que se deben realizar son muy diversos en sus características de procesamiento. Esta situación es descrita por el problema de *Job Shop Scheduling* (JSP), donde se tiene un conjunto finito de trabajos que requieren que una cierta secuencia de operaciones sea realizada en un grupo de máquinas o recursos específicos asumiendo que existe una secuencia de procesamiento para cada uno de los trabajos. El objetivo consiste en determinar el orden en que se realizarán las operaciones de los trabajos dentro de cada máquina, mientras se minimiza el *makespan* [2] (tiempo que tardan en estar listos todos los trabajos, es decir, cuando el trabajo más tardío finaliza), la tardanza u otras medidas de desempeño.

A continuación, se presentan algunos ejemplos en industrias de manufactura en las cuales se evidencia el problema de JSP [1]:

Ejemplo 1: Considere una empresa que produce bolsas, a partir de rollos de papel y cuyo proceso de producción consta de 3 operaciones básicamente:

- Impresión del logo.
- Pegado de un lado de la bolsa.
- Costura de los extremos.

Para cada una de las operaciones se tiene un conjunto de máquinas, con diferentes características cada una, como la variedad de color que pueden imprimir, el tamaño de bolsa que pueden fabricar, la velocidad de procesamiento. Del mismo modo, cada una de las órdenes de producción tiene sus propias especificaciones, incluidas la fecha de entrega prometida y la cantidad a producir. En este caso, si se presentan entregas tardías, la empresa será penalizada; de modo que su objetivo es minimizar el total de penalizaciones.

Por otro lado, también cabe resaltar que los tiempos de configuración de las máquinas están fuertemente ligados a la orden que se estaba procesando anteriormente. Por ejemplo, si dos órdenes consecutivas tienen el mismo tamaño o colores similares, entonces el tiempo de alistamiento será menor que el requerido cuando estas características difieren.

Ejemplo 2: El proceso de fabricación de semiconductores con los chips de memoria y microprocesadores, se hace en instalaciones altamente especializadas. Donde el proceso de producción consta de cuatro fases: Fabricación de obleas, sonda de obleas, ensamble o empaque, y pruebas finales.

La fabricación de obleas es la fase más compleja, tecnológicamente hablando. Capas de metal y el material de las obleas se construyen por patrones sobre obleas de silicio o arseniuro de galio para producir los circuitos, en donde cada capa requiere la siguiente serie de operaciones:

- Limpieza.
- Oxidación, deposición y metalización.
- Litografía.
- Grabado.
- Implantación de iones.
- Eliminación de fotorresistencias.
- Inspección y medición.

Debido a que consta de varias capas, cada oblea debe someterse a estas operaciones varias veces. Por lo tanto, hay una cantidad significativa de recirculación en el proceso. Además, algunas máquinas requieren preparaciones para los trabajos entrantes; el tiempo de configuración a menudo depende de las especificaciones del lote recién completado y el lote que está por comenzar. Al igual que en el ejemplo anterior, cada pedido tiene su propia fecha de lanzamiento y fecha de envío comprometida. Por lo tanto, el objetivo del planificador es cumplir con tantas fechas de envío como sea posible, mientras se maximiza el rendimiento.

Otros sectores donde la producción es tipo *Job Shop* son: farmacéutico, textil, automotriz, metalmecánico, entre otros. Por ejemplo, en la industria metalmecánica se realizan piezas o productos terminados, con características muy específicas provenientes

de órdenes de producción y para realizarlas se deben ejecutar una secuencia de operaciones en una gran variedad de máquinas como: tornos, fresadoras, pulidoras, máquinas CNC, entre otras. Es por todas estas características, que se clasifica como una estructura *Job Shop*, pues a diferencia del *Flow Shop*, en donde la producción es continua y sobre productos estándar, aquí se hace bajo pedido y de forma específica para cada cliente. Esta versatilidad que caracteriza los productos de la metalmecánica, sumado a la multifuncionalidad de las máquinas con la que se suele trabajar, hace que la planeación más allá de ser un problema de JSP se catalogue como un *Flexible Job Shop Scheduling Problem (FJSP)*, pues este caso particular, se caracteriza porque cada una de las máquinas pueden o no realizar varios tipos de tareas.

Adicionalmente, otras características de esta industria, como lo mencionó [3], incluye que actualmente la secuenciación de los trabajos en cada una de las máquinas se hace de forma empírica por parte de los operarios que laboran en el taller, o basándose en reglas de secuenciación simples, tal como lo es FIFO (*First In, First out*) la cual cabe resaltar que no representa el óptimo en la secuenciación de trabajos [4]. También, se presenta que los tiempos de alistamiento entre una operación y otra varían, dependiendo del orden en el que las operaciones se realizan.

Esto se identifica como una oportunidad de mejora, ya que, el método que se implementa actualmente no se preocupa por realizar una asignación óptima, en donde la utilización de los recursos y el cumplimiento a los clientes podría mejorar. Por ejemplo, en Fig. 1 se implementa la regla de secuenciación FIFO para una instancia, y se observa que el *Makespan* es 1410, mientras que en la Fig. 2 se emplea un algoritmo genético multi-objetivo para la misma instancia y el *Makespan* mejora a 992, lo que indica que un cambio en el método de secuenciación conduce a cambios en el tiempo que toma completar todas las tareas.

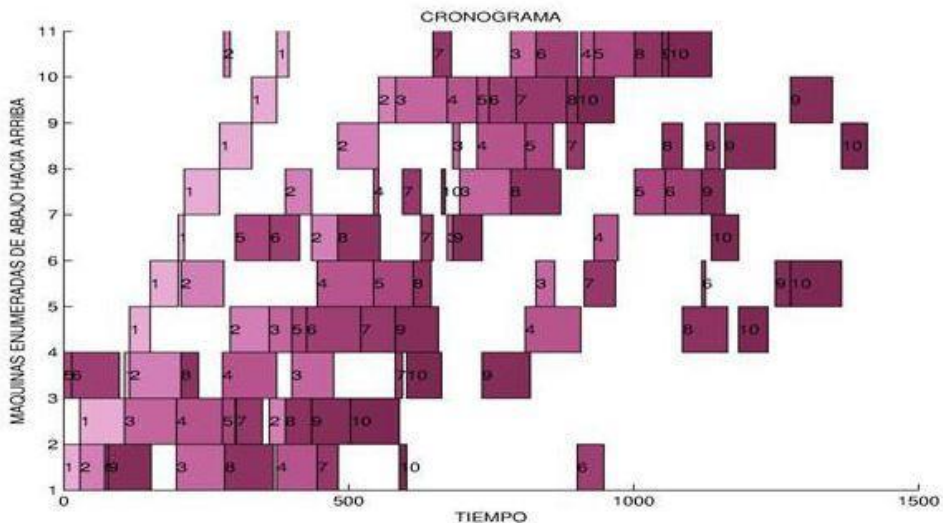


Fig. 1 Diagrama de Gantt para la regla de secuenciación FIFO. Fuente: Tomado de [4].

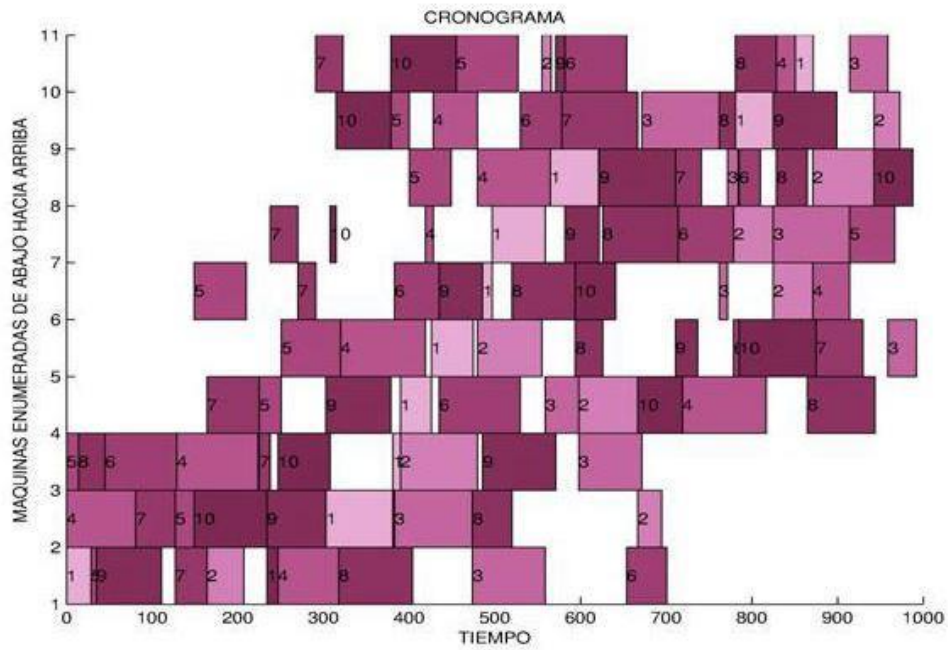


Fig. 2 Diagrama de Gantt empleando un algoritmo genético multi-objetivo. Fuente: Tomado de [4].

En este campo, a pesar de que se han propuesto diversos métodos de solución, entre métodos de solución exactos y heurísticas y metaheurísticas que aproximan el óptimo, los resultados aún no son lo suficientemente buenos cuando se habla de instancias grandes. Esta tarea por supuesto, no ha sido sencilla, pues su característica combinatoria, hace que se catalogue como *NP-Hard* para más de dos máquinas [4], lo que ha atraído en las últimas tres décadas la atención de los investigadores de múltiples áreas a nivel mundial, como lo muestra la Fig. 3 y la Fig. 4.

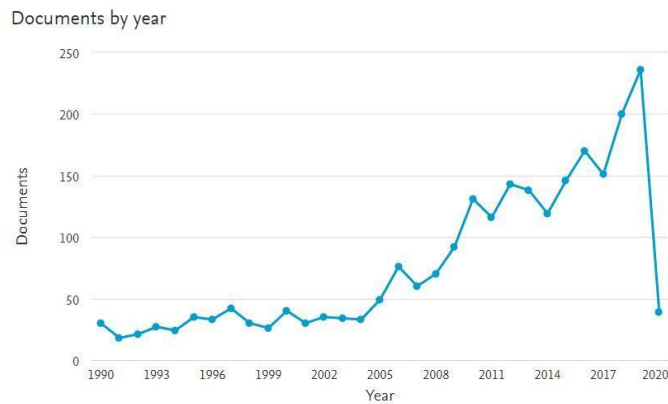


Fig. 3 Crecimiento del interés por parte de investigadores a lo largo del tiempo en el FJSP. Fuente: Tomado de [5].

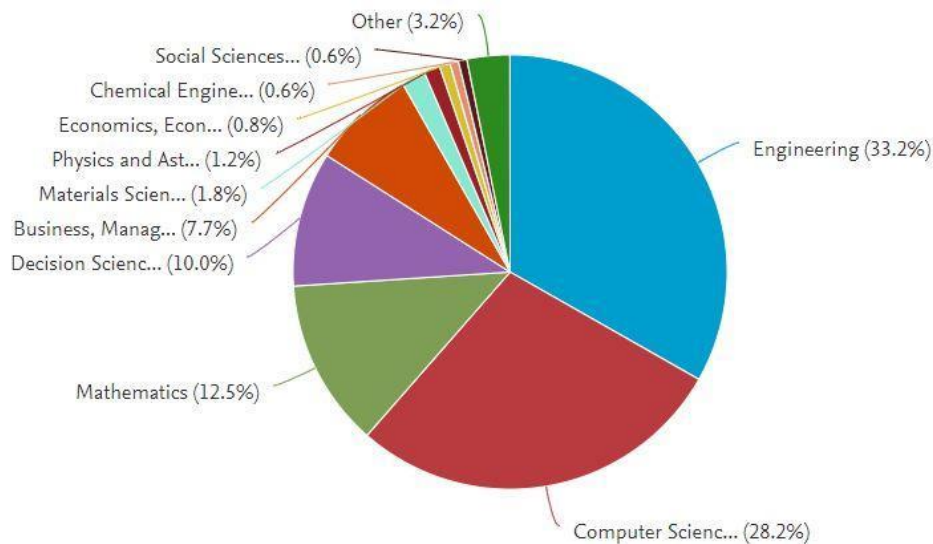


Fig. 4 Investigación del FJSP por cada una de las áreas de conocimientos. Fuente: Tomado de [5].

B. Descripción formal del Flexible Job Shop Scheduling Problem con tiempos de alistamiento dependientes de la secuencia (SDST-FJSP)

El SDST-FJSP considera un conjunto de n trabajos $J = \{J_1, \dots, J_j, \dots, J_n\}$, los cuales deben ser procesados en un conjunto $M = \{M_1, \dots, M_r, \dots, M_m\}$ de m máquinas o recursos. Donde, cada trabajo J_j se compone de un conjunto de a_j actividades $\{1, \dots, i, \dots, a_j\}$ y cada actividad i del trabajo j puede ejecutarse en $k_{ij} \leq m$ modos distintos (máquinas que pueden procesar el trabajo). Así pues, $\left(\left\{\theta_{j11}, \dots, \theta_{j1k_{1j}}\right\}, \dots, \left\{\theta_{ji1}, \dots, \theta_{jik_{ij}}\right\}, \dots, \left\{\theta_{ja_j1}, \dots, \theta_{ja_jk_{a_jj}}\right\}\right)$ representa los modos de ejecución de la secuencia de actividades, siendo θ_{jis} el s -ésimo modo de ejecución de la actividad i del trabajo j . Además, se tiene un tiempo de procesamiento $p_{\theta_{jis}}$ asociado al modo de ejecución y S_{rjl} , el tiempo de alistamiento (*setup times*) en la máquina r , al pasar del trabajo j al trabajo l .

Adicionalmente, en este proyecto se tendrá en consideración los siguientes supuestos:

- Los tiempos de alistamiento que dependen de la secuencia son conocidos y constantes.
- Los tiempos de procesamiento, pueden o no, variar dependiendo del modo de ejecución.
- Las actividades de los trabajos deben ejecutarse sin interrupciones.
- Cada máquina puede procesar máximo un trabajo a la vez.
- Todos los trabajos están disponibles para ejecutarse en el tiempo 0.
- La recirculación está permitida, es decir que un mismo trabajo puede pasar más de una vez por una máquina.

C. Grupos de interés

La identificación de las partes interesadas del proyecto, se definen mediante una lluvia de ideas de los integrantes del equipo, teniendo en cuenta la literatura del libro de Michael Pinedo [1], para ello se abordan características como el impacto e interés que tiene el actor en el desarrollo del proyecto, realizando la siguiente lista de actores interesados:

- La universidad: Institución de Educación Superior que registra y da soporte al desarrollo del proyecto.
- Programadores ALPHA: Equipo de trabajo encargado de desarrollar el proyecto.
- Director del proyecto: Profesor a cargo del proyecto.
- Industrias tipo *Job Shop*: Empresas de manufactura con un modelo de producción tipo JSP.
- Semilleros de investigación: Grupos enfocados en el estudio y análisis de una determinada área de conocimiento.
- Área de planificación de la producción en una empresa de fabricación: Sector encargado de realizar la planeación de producción
- Área de reserva y/o asignación de recursos en una empresa de servicios: Sector encargado de la planeación de recursos.

- Gerente de un sistema ERP: Persona a cargo del sistema de planificación de recursos empresariales.
- Ministerio de Trabajo: Autoridad administrativa que desarrolla acuerdos y promueve el empleo digno para todos los colombianos [6].
- Ministerio de las TICs: Entidad que se encarga de diseñar, adoptar y promover las políticas, planes, programas y proyectos del sector de las Tecnologías de la Información y las Comunicaciones [7].

Con la identificación de las partes interesadas, se realiza una clasificación, mediante la matriz poder vs interés, ya que permite identificar la influencia y atención que tiene la parte interesada sobre el proyecto. Posteriormente, se establecen cuatro categorías o clasificaciones: promotores, latentes, defensores y apáticos. Estas Clasificaciones, son realizadas de acuerdo con la importancia e influencia del actor. Por lo tanto, en la categoría de promotores, se describen los actores de mayor grado de interés e impacto. En la siguiente categoría, los latentes, son aquellos que tienen alto poder, pero su interés es bajo. En la penúltima clasificación, los defensores, son los actores que tienen mucho interés más poco poder. Por último, los apáticos, tienen poco interés y poco poder.

Para realizar la identificación de las partes interesadas en las cuatro categorías, es necesario realizar una tabla, donde se describe el poder e interés, con un rango de uno a seis, donde uno es bajo y seis es alto. Adicionalmente se describe el interés e impacto.

TABLA II.
GRUPOS DE INTERÉS.

N°	Parte interesada	Interés (Influencia)	Nivel	Poder (Impacto)	Nivel
a	La universidad.	Desarrollo de un proyecto con altos estándares de calidad.	3	Aprobación del proyecto.	6
b	Programadores ALPHA.	Desarrollar un método de solución en la programación de los trabajos en las máquinas, aplicable al sector industrial de metalmecánica, y ser referencia en la literatura científica del FJSP.	6	Realización o no del proyecto.	6
c	Director del proyecto.			Acompañamiento, soporte y ayuda para la realización del proyecto.	5
d	Industrias tipo <i>Job Shop</i>	Disminuir los tiempos de trabajo, según el anexo 1.	6	Aplicación y validación del proyecto	3
e	Semilleros de investigación.	Aumento de la literatura científica en el tema de investigación desarrollado	5	Revisión del resultado	2
f	Área de planificación.	Solución de problemas relacionados con el tiempo de procesamiento de los trabajos.	5	Aplicación de la solución, en la secuenciación de trabajos en las máquinas.	3

N°	Parte interesada	Interés (Influencia)	Nivel	Poder (Impacto)	Nivel
g	Área de reserva y/o asignación de recursos.	Solución de problemas relacionados con el tiempo de trabajo ejecutado de un servicio específico.	5	Aplicación de la solución, en la asignación de recursos.	3
h	Gerente de un sistema ERP.	Solución de problemas relacionados a la gestión de recursos para la ejecución de trabajos.	4	Aplicación de la solución para los sistemas MPS Y MRP.	2
i	Ministerio de las TICS.	Cumplir los reglamentos y regulaciones para el manejo de datos.	2	Sanción del proyecto realizado si se incumple con los requerimientos.	3
j	Ministerio del trabajo.	Cumplir con los tiempos de trabajo mínimos para un operario.	2	Sanción por el incumpliendo de los horarios del trabajador.	2

Con los niveles de influencia y poder, se realiza la matriz de interesados, clasificando a cada una de las partes interesadas en las cuatro categorías. De esta forma, se divide el plano en los cuatro cuadrantes correspondientes, donde el eje x es el nivel de interés, y el eje y es el nivel de poder. Finalmente, se obtiene la Fig. 5.

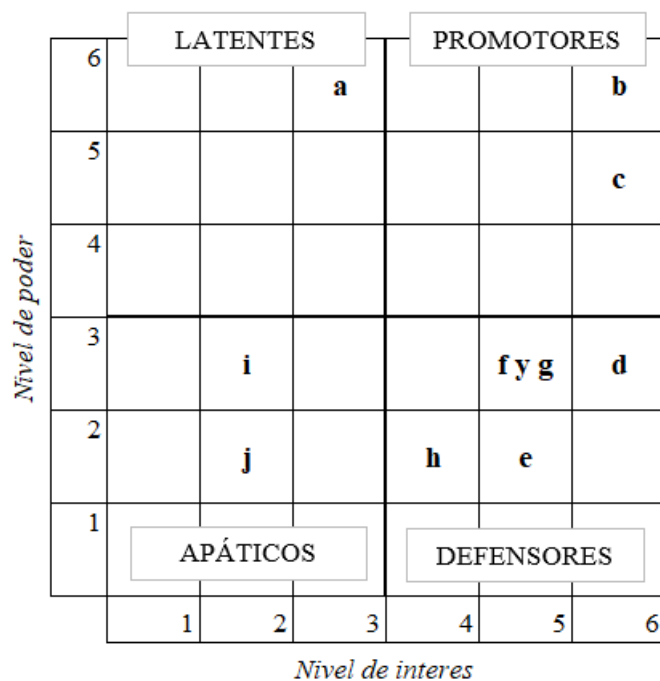


Fig. 5 Matriz de poder vs interés

La clasificación realizada permite identificar el trato con cada uno de los actores de interés. Iniciando con los promotores, siendo aquellos quienes desarrollan el proyecto, es decir, el equipo “Programadores ALPHA” y el director del proyecto. Luego, en el grupo de latentes, se encuentra la universidad, actor que debe ser informado constantemente del desarrollo del proyecto. Posteriormente, en el grupo defensores, se encuentran los actores que pueden aplicar el resultado del proyecto, siendo un grupo bastante diverso, debido al amplio campo de acción en los sectores industrial, servicios e investigación. A este grupo de defensores es de vital importancia reportarles la información del proyecto, de acuerdo con el siguiente orden de priorización, descrito de mayor a menor:

- Industrias tipo *Job Shop*.
- Área de planificación.
- Área de reserva y/o asignación.
- Semilleros de Investigación.
- Gerente de un sistema ERP.

Por último, los apáticos, deben ser monitorizados. En este caso, se debe realizar un análisis constante de los movimientos y cambios en los decretos de los ministerios del trabajo y las TICS, que pueda afectar el desarrollo del proyecto.

C. Requerimientos

Con el objetivo de identificar las restricciones y satisfacer las necesidades de los actores de interés, se hace indispensable identificar los requisitos de cada uno de ellos y así determinar qué tipo de restricciones impiden el desarrollo de estos mismos, teniendo en cuenta el diseño (factibilidad y sus características), leyes, normas y estándares que puedan afectar el desarrollo del proyecto correctamente.

1) Restricciones de diseño (Factibilidad).

Durante el desarrollo del proyecto pueden existir factores que limiten llevar a cabo la implementación de una propuesta, como solución al problema SDST-FJSP. En el diseño de factibilidad, los principales requerimientos difieren según el actor de interés. A continuación, se presenta una lluvia de ideas de posibles restricciones de diseño en la factibilidad.

- Costos y procesos de producción: factor relevante para calcular la utilidad de la empresa.
- Lineamientos de la universidad: características necesarias que debe presentar el documento, tales como la etapa definir del proyecto.
- Fechas de entrega: tiempo de finalización de cada parte del desarrollo del proyecto.
- Reuniones: mínimas por grupo y con el director del proyecto.
- Tiempos de alistamiento dependientes de la secuencia: después de haber ejecutado una operación (trabajo *j*) se requiere realizar un ajuste de herramientas, cambiar pastillas, limpieza, etc. De manera que la máquina se encuentre lista y en óptimas condiciones para realizar la siguiente operación. Pueden ser separables y no separables:
 - Separables: permite una holgura mayor que cero entre la finalización del alistamiento y el inicio de la siguiente operación.
 - Inseparables: después de haber finalizado el alistamiento, inmediatamente debe iniciarse la actividad correspondiente.
- Tiempo de ejecución: Intervalo de tiempo en que se ejecuta el algoritmo.
- Una vez una operación es iniciada en una máquina, esta no podrá interrumpirse. Es decir, parará solo al final.
- Capacidad del software: según el tipo de algoritmo y la velocidad del software del computador, será posible hallar una respuesta acorde con los requerimientos de la empresa.
- Reglas de despacho: FIFO, SPT, EDD, etc.
- Información verídica.
- Número de máquinas: conjunto de máquinas necesarias para realizar cada conjunto de trabajos.
- Número de operarios: necesarios para realizar cada alistamiento y control del proceso.
- Capacidad de cada máquina: para realizar tantas operaciones en un conjunto de máquinas disponibles.
- Velocidad de la máquina: para realizar cada conjunto de trabajos, con el objetivo de minimizar el *Makespan*.
- Secuenciación: programación de todas las operaciones en todas las máquinas buscando una solución factible.
- Mantenimiento de las máquinas: Planear el cronograma de mantenimiento de máquinas con el fin de no parar todas las máquinas el mismo día, así no interrumpir la producción.
- Tiempo de producción: Tiempo total de proceso en elaborar un producto determinado incluyendo tiempos de espera, de preparación, operación y de transferencia.
- Tiempos de alistamiento: importante para aumentar la productividad. El tiempo de alistamiento está determinado por la relación que hay entre las operaciones de todos los trabajos.

- Horas laborales permitidas: las horas necesarias para tener el control y supervisión del proceso no puede exceder las horas laborales permitidas.

2) Especificaciones de diseño (características).

El diseño de este proyecto debe estar relacionado con los requisitos de las partes interesadas, por tanto, se deben seguir unos ciertos lineamientos para poder cumplir en mayor medida con cada uno de ellos. La principal característica para el desarrollo del proyecto es la necesidad de disminuir al máximo el *Makespan*. Ya que un sistema SDST-FJSP, se caracteriza por la capacidad de realizar distintas operaciones en un conjunto de máquinas compatibles, se hace necesario determinar aquellas especificaciones de diseño necesarias para la realización de este proyecto, como, por ejemplo:

- Tipo de productos: depende para la correcta asignación del conjunto de máquinas compatibles con dicho producto.
- Estándares de calidad.
- Presupuesto: la inversión necesaria para el correcto desarrollo de un algoritmo inteligente capaz de resolver el problema de SDST-FJSP.
- Asignación de máquinas: Selección de las máquinas en las que debería realizarse las operaciones de cada trabajo.
- Interrupción de trabajos: Los tiempos entre trabajos no se pueden interrumpir.
- Metodología DMADV.
- Cronograma: tener un orden adecuado para la realización de las actividades necesarias para el desarrollo del proyecto.
- *Team charter*: Roles y tareas que debe realizar cada integrante del equipo.
- Actas de reunión: control de reuniones realizadas en grupo y con el director de tesis.
- *Makespan*.
- Fuentes de información confiables.
- Tipo de máquina.
- Tipo de operación.
- Inventarios: tener materia prima disponible en todo momento.
- Fichas de trabajo: para tener un control de las horas realizadas por cada operario.

A continuación, se presentan las restricciones de diseño (factibilidad y características), dependiendo de cada grupo de interés, determinando el impacto que puede generar cada una de ellas en el desarrollo del proyecto.

3) Leyes, normas y estándares.

Actualmente, el derecho de autor se aplica sobre todas las formas en las que se puede dar a exponer o expresar una idea. En la era digital, en el caso de software, la ley colombiana, asume este como una obra literaria permitiendo que el código fuente de un programa esté cubierto por la ley de Derechos de Autor.

La propiedad industrial, se enfoca en la protección de aquellas ideas que pueden ser aplicadas en cualquier sector productivo o de servicio. Para poder obtener este tipo de protección en Colombia, se requiere de un registro formal en la superintendencia de industria y comercio, pero este solo será válido en un transcurso de unos años por motivo de poder asegurar la monopolización y explotación económica de este mismo [8].

En ingeniería, cuando se habla de estándares de calidad de un software, se refiere a la utilización de metodologías para el correcto diseño y programación, con el fin de tener mayor confiabilidad, eficiencia y eficacia. A continuación, se presentan algunas posibles leyes, normas y estándares que hay que tener en cuenta para el desarrollo de este proyecto:

- Artículo 27.1 del acuerdo sobre los Aspectos de Propiedad Intelectual Relacionados con el Comercio (ADPIC), en tanto que impide la patentabilidad de este tipo de creaciones. Esto afecta el desarrollo de la industria creativa del software y como ejemplo se analiza el caso de la industria de los videojuegos [9].
- Ley 23 de 1982 Nivel Nacional: Derechos de autor [10].
- Normas IEEE: elaboración de citas y bibliografías para hacer uso ético de la información y así, evitar el plagio.
- Norma ISO/IEC 25000 (*System and Software Quality Requirements and Evaluation*): es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software [10]
- Propiedad industrial: se dedica a la protección de las creaciones industriales (invenciones, modelos de utilidad, diseños), y de los signos distintivos (marcas, nombres, lemas, enseñas, denominaciones de origen), que son protegidos a través de patentes, registro de diseños industriales, registros de marcas y demás. Es la Decisión 486 de 2000 de la Comunidad Andina de Naciones –CAN– Régimen Común Sobre Propiedad Industrial [8].

- Política de la empresa: la empresa que decida y disponga de la necesidad de implementar FJSP-SDST cuenta con normas internas.
- Norma ISO/IEC 9126: Esta norma evalúa los productos de software, indica características de calidad y lineamientos para su uso.
- Estándar ISO/IEC 14598: Uso de metodología base para evaluación del producto software.
- Código sustantivo de trabajo artículo 167: Horas de trabajo durante cada jornada.
- Código sustantivo de trabajo, artículo 160: Trabajo ordinario y nocturno.
- Código sustantivo de trabajo, artículo 161: Duración máxima de jornada ordinaria de trabajo.
- Ley 50, artículo 22: horas extras máximas.
- Decreto 1078 de 2015: Decreto único reglamentario del sector TIC.

TABLA III.
REQUERIMIENTOS DE LEYES, NORMAS Y ESTÁNDARES DE LOS ACTORES DE INTERÉS.

N°	Grupos de Interés	VOC (Requisitos del cliente)	Restricciones de diseño	Especificaciones de diseño	Leyes, Normas, Estándares	
					Legislación y requisitos aplicables	Importancia o efecto
a	La universidad	Desarrollo de un proyecto con altos estándares de calidad.	Lineamientos.	Metodología DMADV y propuesta para Evaluación de Proyectos de Diseño enfocados en Investigación Aplicada con base en DMADV.	Normas IEEE.	Acreditación ABET.
b	Programadores ALPHA	Desarrollar un método de solución en la programación de los trabajos en las máquinas, aplicable al sector industrial de metalmecánica, y ser referencia en la literatura científica del FJSP.	Fechas de entrega. Reuniones mínimas con el director de tesis. Búsqueda de información. Disponibilidad de tiempo del director de tesis para atender dudas del grupo.	Cronograma. Team Charter. Actas de reunión. Comunicación vía virtual.	Norma ISO/IEC. Reuniones mínimas por entrega con el director del proyecto, cinco (5).	Tener un orden y desarrollo adecuado del proyecto. Aplicabilidad del proyecto en distintos campos industriales con alta calidad.
c	Director del proyecto					
d	Industrias tipo <i>Job Shop</i>	Disminuir los tiempos de trabajo en el taller, según el anexo 1.	Tiempo de ejecución de un proceso. Costos de producción. Tiempos de alistamiento. Reglas de despacho.	Disminuir el <i>makespan</i> .	Normas establecidas por la empresa.	Mayor competitividad con respecto a otras industrias metalmecánicas. Cumplir con las fechas de entrega de todos los pedidos.
e	Seminarios de investigación.	Aumento de la literatura científica en el tema de investigación desarrollado	Información verídica.	Fuentes de información confiables.	Artículo 27.1. Decreto 486 del 2000 CAN.	Obtener información real y confiable. Protección de la información.
f	Área de planificación	Solución de problemas relacionados con el tiempo de procesamiento de los trabajos.	Capacidad de las máquinas. Número de máquinas. Velocidad de las máquinas. Secuenciación. Mantenimiento de las máquinas. Tiempo de producción	Tipo de máquina. Tipo de producto. Tipo de operación.	Normas establecidas por la empresa.	Datos necesarios para el desarrollo correcto de la planificación.

N°	Grupos de Interés	VOC (Requisitos del cliente)	Restricciones de diseño	Especificaciones de diseño	Leyes, Normas, Estándares	
					Legislación y requisitos aplicables	Importancia o efecto
			y alistamiento de máquinas. Secuenciación. Número de operarios.			
g	Área de planificación Área de reserva y/o asignación de recursos.	Solución de problemas relacionados con el tiempo de procesamiento de los trabajos. Solución de problemas relacionados con el tiempo de trabajo ejecutado de un servicio específico.	Capacidad de las máquinas. Número de máquinas. Velocidad de las máquinas. Secuenciación. Mantenimiento de las máquinas. Tiempo de producción y alistamiento de máquinas. Secuenciación. Número de operarios. tiempos de procesamiento. Materia prima disponible. Maquinas compatibles.	Tipo de máquina. Tipo de producto. Tipo de operación. Inventarios. Tipo de producto Tipo de máquina.	Normas establecidas por la empresa. Normas establecidas por la empresa.	Datos necesarios para el desarrollo correcto de la planificación. Asignar recursos adecuadamente.
h	Gerente de un sistema ERP.	Solución de problemas relacionados a la gestión de recursos para la ejecución de trabajos.	Capacidad del software. Presupuesto. Costos y procesos de producción.	Presupuesto.	Normas ISO/IEC	Visualizar el alcance del software y su calidad.
i	Ministerio de las TICS.	Cumplir los reglamentos y regulaciones para el manejo de datos.	Diseño y desarrollo del software.	Leyes y normas.	Decreto 1078 de 2015	Incrementar y facilitar el acceso de todos los habitantes del territorio nacional a las Tecnologías de la Información y las Comunicaciones y a sus beneficios.
j	Ministerio de trabajo.	Cumplir con los tiempos de trabajo mínimos para un operario.	Horas laborales permitidas (diurna, nocturna, y extras).	Fichas de trabajo	Código sustantivo de trabajo. Artículo 160 (horas diurnas y nocturnas). Código sustantivo de trabajo. Artículo 167 (Tiempo de descanso). Código sustantivo de trabajo. Artículo 161 (Jornada máxima). Artículo adicionado por el artículo 22 de la Ley 50 de 1990 (Máximo horas extras).	Controlar y equilibrar el peso de trabajo de todos los trabajadores.

III.MEDIR

La fase de medición contiene dos partes fundamentales: plan de recolección de datos y exploración del mercado. La primera parte, desarrolla la elección de las bases de datos aplicadas al SDST-FJSP, identificando las variables de entrada, salida y de medición. La segunda parte, inicia con los resultados de las variables de medición del método actual, continuando con un resumen de casos exitosos aplicados al FJSP, finalizando, con un análisis FODA.

A. Plan de recolección de datos

En este apartado, se explicará de manera completa cómo se realizó la recolección de datos y se establecen los indicadores de desempeño con los que se evaluarán los resultados del proyecto en la etapa de verificar. Teniendo en cuenta la definición realizada en la sección II.BII.B, las variables de entrada son: El tiempo de procesamiento de la actividad i del trabajo j en la máquina r ($P_{\theta_{jmr}}$), el tiempo de configuración (*Setup Time*) de la máquina r del trabajo j que precede al trabajo l (S_{rjl}) y la secuencia de las actividades para el trabajo j (θ_{jmr}). Estas variables se encuentran definidas y operacionalizadas en la TABLA IV.

Para la adquisición de dichas variables, se procedió a realizar la búsqueda de casos prueba (llamadas en este documento como instancias) en la literatura científica propia del SDST-FJSP. En el acta del congreso de Angelo Oddi, Riccardo Rasconi, Amedeo Cesta y Stephen F. Smith del año 2011 [11] muestran un conjunto de instancias del SDST-FJSP citadas como SDST-HUData, las cuales añaden los tiempos de alistamiento a las primeras 20 instancias de edata (subconjunto de FJSSP HU-data) reportadas en [12]. Así pues, SDST-HUData consiste en 20 problemas de tamaño 10 X 5, 15 X 5, 20 X 5 y 10 X 10, donde se especifica las variables mencionadas anteriormente, incluyendo los diferentes modos en los que se puede ejecutar una actividad i del trabajo j . En la siguiente Fig. 6 se describe la información del modelo de la instancia SDST_HUData.

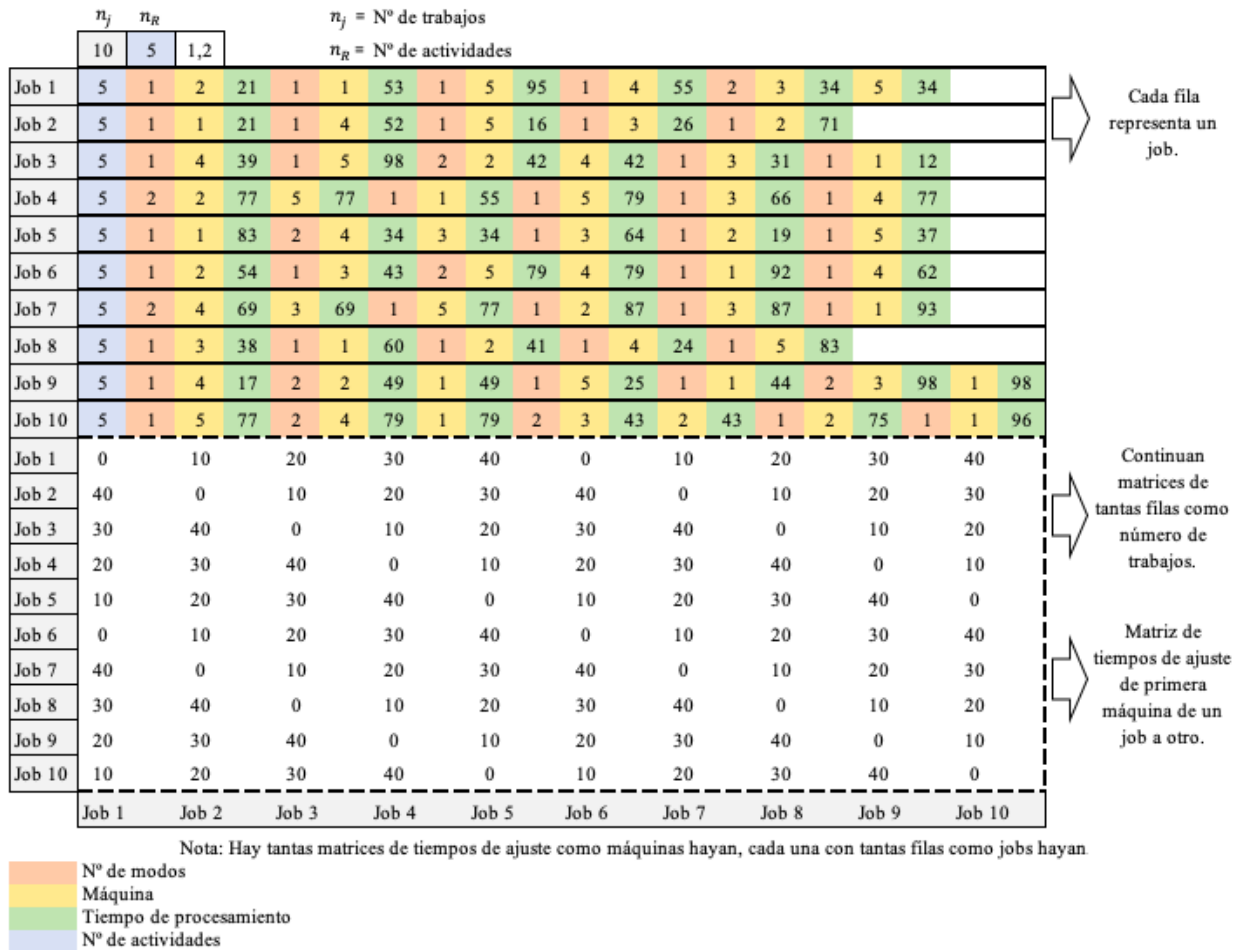


Fig. 6 Modelo de instancia SDST_HUData

A continuación, en la TABLA IV, se presenta una descripción completa de las variables a medir y como estas fueron recolectadas.

TABLA IV
PLAN DE RECOLECCIÓN DE DATOS.

Tipo	Variable	Definición	Utilidad para el proyecto	Método	Recursos de medición	Presentación
Entrada	Tiempo de procesamiento ($P_{\theta_{jmr}}$)	Tiempo de operación que tarda en realizarse la actividad i del trabajo j en la máquina r .	Datos de entrada requeridos para la ejecución de un método de solución.	Búsqueda de instancias o simulación de datos.	Lectura de las instancias SDST-HUData y/o simulación de los datos en C++.	En las instancias se presenta adimensional, sin embargo, podría tomar cualquier unidad de tiempo.
	Tiempo de configuración (S_{rjl}).	Tiempo que se tarda en ajustar la máquina r del trabajo j al trabajo l .	Datos de entrada requeridos para la ejecución de un método de solución.	Búsqueda de instancias o simulación de datos.	Lectura de las instancias SDST-HUData y/o simulación de los datos en C++.	En las instancias se presenta adimensional, sin embargo, podría tomar cualquier unidad de tiempo.
	Secuencia de las actividades $\left(\left\{ \theta_{j11}, \dots, \theta_{j1k_{1j}}, \dots, \right. \right.$ $\left. \left\{ \theta_{ja_j1}, \dots, \theta_{ja_jk_{a_jj}} \right\} \right)$	Orden de ejecución de las a_j actividades del trabajo j , en los k_{ij} modos distintos por actividad. Aquí se especifica las máquinas en la cuales se puede ejecutar cada actividad.	Datos de entrada requeridos para la ejecución de un método de solución.	Búsqueda de instancias o simulación de datos.	Lectura de las instancias SDST-HUData y/o simulación de los datos en C++.	En las instancias se presenta adimensional, sin embargo, podría tomar cualquier unidad de tiempo.
Salida	Secuencia de los trabajos en la máquina r .	Orden en el cual la máquina r procesará cada uno de los trabajos.	Hace parte de la caracterización de la solución propuesta por un método dado.	Diferentes métodos de solución para un SDST-FJSP.	Ejecución del método en C++.	Diagrama de Gantt y tabla resumen.
	Modo elegido (K_{ij}).	Pareja de tiempo de procesamiento y máquina en las que se puede ejecutar la actividad i del trabajo j .	Hace parte de la caracterización de la solución propuesta por un método dado.	Diferentes métodos de solución para un SDST-FJSP.	Ejecución del método en C++.	Diagrama de Gantt y tabla resumen.
	Tiempos de inicio. (TI_{ij}).	Tiempo de inicio de cada una de las actividades del trabajo j .	Hace parte de la caracterización de la solución propuesta por un método dado.	Diferentes métodos de solución para un SDST-FJSP.	Ejecución del método en C++.	Diagrama de Gantt y tabla resumen.
	Tiempos de finalización. (TF_{ij}).	Tiempo de finalización de cada una de las actividades del trabajo j .	Hace parte de la caracterización de la solución propuesta por un método dado. Y es empleado en el cálculo del <i>Makespan</i> como se detallará más adelante.	Diferentes métodos de solución para un SDST-FJSP.	Ejecución del método en C++.	Diagrama de Gantt y tabla resumen.

Posteriormente, se procedió a realizar la recolección de datos del tipo de programación implementado en la industria del sector metalmeccánico, el cual como se ha mencionado anteriormente, hace parte de los sectores que se podrían catalogar como una producción tipo FJSP. Por tanto, se realizaron entrevistas a 10 empresas de este sector, en el Valle del Cauca, el día 20 de marzo del 2020 y cuyos resultados se presentan en el anexo II. Así pues, con la información recolectada, se estableció el tipo de programación empleada era manual en el 100% de las empresas metalmeccánicas encuestadas, al hacer una comparación con las reglas de secuenciación básicas, se encontró que esta coincidía con la regla básica de secuenciación ERD (*Earliest Release Date*), caracterizada por programar los trabajos teniendo como prioridad por orden de llegada. Esta regla se describe en detalle por el diagrama de flujo del anexo III y el pseudocódigo mostrado en la Fig. 7.

```

//Algoritmo ERD
Sea j el conjunto de trabajos; TP los tiempos de procesamiento de la actividades; ST Tiempos de configuración
i el conjunto de actividades; m los modos de ejecución de las actividades; gap Brecha de tiempo disponible entre dos actividades;
TimeI el tiempo de inicio de la actividad i en el trabajo j; g es el conjunto de gaps
TimeF el tiempo de finalización de la actividad i en el trabajo j;
OrganizarTrabajos("De acuerdo a su orden de llegada(ERD)")
for (j = 0 to Jobs){
  if (j = 0){
    Programar actividades lo más pronto posible de acuerdo a su primer modo
  }
  else{
    for (i = 0 to actividades_j){
      for (m = 0 to modos_i){
        for (g = 0 to gap){
          if (i = 0 y (TP + ST) < TimeF("del anterior trabajo programado")){
            Programar actividad al inicio
          }
          else{
            if (TP + ST <= gap){
              Programar actividad en medio del gap
            }
            else{
              Programar actividad al finalizar el último gap
            }
          }
          if (TimeF_i < MejorModo){
            MejorModo = TimeF
          }
        }
      }
    }
    Programar el MejorModo
  }
}
}
}

```

Fig. 7 Pseudocódigo del proceso de programación del ERD.

B. Exploración del mercado

1) Casos aplicados

En esta sección se pretende resaltar la aplicabilidad de los diferentes métodos de solución en problemas tipo *Job shop* y *Flexible Job Shop*, para ello se hizo una revisión de casos exitosos aplicados a diferentes tipos de industria y se registran en la TABLA V.

TABLA V
RESUMEN DE CASOS EXITOSOS PARA EL *FLEXIBLE JOB SHOP*.

Título	Autores	Año	Variante del problema	¿Dónde se aplicó?	Resultados obtenidos	Fuente
<i>A dispatching algorithm for flexible job-shop scheduling with transfer batches: an industrial application</i>	Gema Calleja, Rafael Pastor	2013	FJSP con lotes de transferencia. Considerando como objetivo, minimizar la tardanza promedio.	Fábrica componentes metálicos para la industria automotriz	Los resultados de la implementación del algoritmo presentaron una mejora promedio de la tardanza promedio de 17.04% para la variante ordenada, y 22.30% para la variante aleatorizada con respecto al método de programación manual anterior utilizado en la planta.	[13]
<i>Multi-Objective Flexible Job-Shop Scheduling Problem in Steel Tubes Production</i>	Li Lin, Huo Jia-zhen	2009	FJSP con restricciones de capacidad, velocidad y mantenimiento de las máquinas, restricciones de inventario intermedio y multiobjetivo: reducir el retraso de las entregas, minimizar la inactividad de las máquinas y la interrupción en la producción.	<i>Baoshan Iron & Steel Complex</i> Fabrica de producción de tubos de acero.	(1) Mejora la calidad de la programación en comparación con el esquema de programación producido por el sistema manual. La mejora promedio del nuevo sistema es superior al 10% . (2) Reduce la pérdida de tiempo y aumenta la eficiencia para la programación.	[14]
<i>A genetic algorithm-based scheduler for multiproduct parallel machine sheet metal job shop</i>	Felix TS Chan, K.L. Choy, Bibhusban	2011	FJSP con tiempos de alistamiento dependientes	Taller mecanizado de chapas.	Este algoritmo da como resultado una reducción del 11.67% del <i>makespan</i> para el problema de un ejemplo de una industria.	[15]
Metodología multiobjetivo basada en un comportamiento evolutivo para programar sistemas de producción <i>Flexible Job Shop</i> . Aplicaciones en la industria metalmeccánica	Santiago Ruiz Herrera	2015	FJSP	Industria metalmeccánica	La implementación del algoritmo generado proporciona mejoras mayores al 10% en el <i>makespan</i> y costo total del recurso consumido.	[16]
<i>An adaptive neuro-fuzzy inference system for makespan estimation of flexible manufacturing system assembly shop: a case study</i>	Vineet Jain, Tilak Raj	2018	El problema de este estudio es encontrar la estimación del <i>makespan</i> por adelantado si el tiempo de procesamiento de las máquinas son conocidas, en un sistema de fabricación flexible.	Empresa multinacional india de automóviles.	Se obtiene un porcentaje promedio de precisión alcanzada del 95.97% , y un coeficiente de determinación y coeficiente de correlación del 0.9310 y 0.964 respectivamente.	[17]
<i>Solving Flexible Job-Shop Scheduling Problem with Transfer Batches, Setup Times</i>	M.Ortiz, D. Neira, G.Jiménez y H.Hernández	2016	Mejorar la tardanza de sistemas flexibles de trabajo estructurados, para fabricar una amplia gama de productos personalizados.	Industria de confección	Mejora del 61.1% en la tardanza promedio, una reducción del 19.2% en la desviación estándar de la tardanza, una reducción del 54.3% en los pedidos. También ofrece un mejor desempeño de las medidas de tardanza que las proporcionadas por PGDHS y HHS / LNS con una reducción de 2.63% y 1.77% correspondientemente.	[18]
<i>A hybrid genetic tabu search algorithm for solving job shop scheduling problems: a case study</i>	S. Meeran · M. S. Morshed	2011	JSP con objetivo: minimización del <i>makespan</i> .	Sector de las acerías y del sector de parabrisas de automóviles.	La compañía podría ahorrar tiempo y aumentar la productividad, también podría reducir alrededor del 75% de su tiempo de producción simplemente reduciendo el tiempo de inactividad de la máquina mediante la implementación del sistema propuesto.	[19]
<i>Flexible job-shop scheduling with flexible workdays, preemption, overlapping in operations and satisfaction criteria: an industrial application</i>	Jiae Zhang y Jianjun Yang	2016	El FJSP con jornadas flexibles, preferencia, superposición de operaciones y criterios de satisfacción. El objetivo es maximizar la satisfacción.	Empresa aeronáutica en China.	El grado de satisfacción combinado de esta solución optimizada es de 0,81 .	[20]

2) *Análisis del entorno.*

Con el fin de analizar las características del entorno del proyecto, en la TABLA VI se implementa una MATRIZ DOFA, la cual nos permite identificar las fortalezas y debilidades (desde las características externas), y las oportunidades y amenazas (desde las características externas), y así identificar las estrategias que alineen mejor los recursos y capacidades del proyecto.

TABLA VI.
ANÁLISIS DOFA/FODA.

<h1>MATRIZ DOFA</h1>	<p>Debilidades: D1: No se cuenta actualmente con un buen plan de implementación de capacitaciones en el sector para entendimiento del proyecto. D2: No se cuenta actualmente con un plan adecuado para fomentar un lenguaje técnico en el sector</p>	<p>Fortalezas: F1: El proyecto implementa el uso de la tecnología y la ingeniería, con el fin de optimizar los procesos industriales del sector metalmeccánico F2: Al implementar el proyecto, se reducen significativamente los costos y los tiempos del proceso. F3: El proyecto es útil para gerentes con problemas de tiempos de entregas de pedidos, o para empresas con desperdicios en sus recursos debido a la mala asignación de actividades en sus máquinas.</p>
<p>Oportunidades: -O1: La implementación del proyecto mejoraría significativamente la eficiencia de las actividades de la industria metalmeccánica -O2: El sector provee a muchos otros sectores industriales (desde la industria automotriz, hasta la industria aeronáutica)</p>	<p>Estrategia (D1-O1): Estandarizar estas capacitaciones en el sector, por medio de cursos concisos y cortos, bien sea con videos suministrados en la red, o con manuales guía incluidos al adquirir el método de mejora. (D2-O2): Implementar un lenguaje técnico en la industria metalmeccánica, para que se normalice el entendimiento de este método de mejora</p>	<p>Estrategia (F1-O1): Empezar con una campaña de <i>marketing</i> que impulse el método al conocimiento de más personas. O2: Analizar e implementar un gremio de sectores industriales para que se trabaje de una forma sistémica.</p>
<p>Amenazas: A1: El sector metalmeccánico no cuenta con la tecnología pertinente para la ejecución e implementación de la propuesta de solución al problema FJSP-SDST. A2: El sector metalmeccánico no cuenta con equipos avanzados requeridos para la realización del proyecto.</p>	<p>Estrategia (D1-A1): Realizar acompañamiento “prestado” por el gobierno, para implementar un capital destinado a renovación de equipos acompañados con la debida capacitación para el sector, para después hacer el debido retorno de pago. A2: Oportunidad para incentivar la creación e implementación de equipos eficientes en los diferentes mercados. Ser una motivación para ello.</p>	<p>Estrategia (F1-A1): Implementar un acompañamiento “prestado” gubernamental para comenzar con la implementación de equipos tecnológicos a la altura en el sector metalmeccánico.</p>

Esto, sumado a los casos exitosos para el *Flexible Job Shop* descritos anteriormente en la TABLA V, son evidencia de las oportunidades de mejora del sistema al implementar la ingeniería y la tecnología en los procesos industriales, sin importar el sector en el que se enfoque, siempre y cuando cumpla con los supuestos del modelo. Sin embargo, no debe dejarse de lado las amenazas o debilidades ante las que se enfrenta el proyecto y es importante idear medidas para superar adversidades tales como, la falta de capacitación de los operarios.

3) *Estado actual del sistema*

Con el fin de evaluar el desempeño del sistema en el estado actual, es decir, la regla ERD, se implementó un algoritmo en C++. Posteriormente, se planteó los KPI's descritos en la TABLA VII, los cuales permiten hacer una comparación futura del método actual con otros métodos de secuenciación, entre ellos, el diseño que se propondrá en este proyecto.

TABLA VII
INDICADORES DE DESEMPEÑO A MEDIR.

Variable	Objetivo	Descripción	Indicador	Actualidad	Meta
<i>Makespan</i> C_{max}	Entre más pequeño sea el <i>Makespan</i> , mejor es la solución que se propuesta, pues el proyecto ejecuta en un menor tiempo. Así pues, esta medida de desempeño permite evaluar la calidad de la solución propuesta.	Es tiempo en el que se completa el último trabajo que abandona el sistema. Por lo tanto, es el tiempo que toma completar todo el proyecto.	$\max C_j$ Donde C_j es el tiempo en el que el trabajo j abandona el sistema [1]. Por lo tanto, puede expresarse como $C_j = \max T_{Fij}$.	La TABLA VIII muestra los resultados obtenidos para las instancias SDST_HUData al implementar el método ERD. Además, el <i>makespan</i> promedio es de 1501,25.	Disminución del <i>Makespan</i> en al menos un 10%.
MRE (Mean Relative Error)	Este indicador hace una comparación entre el <i>Makespan</i> obtenido y una cota inferior (LB) del <i>Makespan</i> óptimo. De este modo, permite analizar qué tan cercana al valor de la cota, es una posible solución.	Error relativo promedio	$MRE = \left(\frac{C_{max} - LB}{LB} \right) (100)$ Donde LB es la cota inferior encontrada para la instancia dada. [21]	La TABLA VIII muestra los resultados obtenidos para las instancias SDST_HUData al implementar el método ERD. Además, el MRE promedio es de 82%.	Disminución del MRE. El cual se espera que sea un valor tan cercano a cero como sea posible.

Las cotas inferiores corresponden a valores que son menores o iguales al óptimo y pueden provenir de la relajación (sin la condición de integralidad) del modelo matemático. Su principal función es servir de punto referencia para determinar qué tan cercana es una solución al óptimo, cuando este es desconocido o es muy complejo de calcular.

Como se mencionó anteriormente, se implementó el método de secuenciación ERD y la regla de secuenciación básica SPT (*Shortest Processing Time*), la cual consiste en programar primero aquellos trabajos que tienen un tiempo de procesamiento más corto; ambas mediante un programa en C++, con el fin de evaluar el comportamiento actual del sistema. Dicho programa lee las 20 instancias de SDST_HUData y proporciona las variables de salida descritas en la TABLA IV con las cuales se calculan los KPI's reportados en la TABLA VIII.

TABLA VIII
RESUMEN DE KPI'S PARA SDST_HUDATA.

Instancia	Tamaño	LB	C_{max} SPT	MRE SPT	C_{max} ERD	MRE EDR
la01	10 x 5	609	1391	128.41%	1225	101.15%
la02	10 x 5	655	1173	79.08%	1081	65.04%
la03	10 x 5	550	1063	93.27%	966	75.64%
la04	10 x 5	568	1212	113.38%	1061	86.80%
la05	10 x 5	503	1199	138.37%	905	79.92%
la06	15 x 5	833	1699	103.96%	1367	64.11%
la07	15 x 5	762	1550	103.41%	1327	74.15%
la08	15 x 5	845	1766	108.99%	1398	65.44%
la09	15 x 5	878	1760	100.46%	1555	77.11%
la10	15 x 5	866	1873	116.28%	1486	71.59%
la11	20 x 5	1087	2025	86.29%	1788	64.49%
la12	20 x 5	960	1863	94.06%	1958	103.96%

Instancia	Tamaño	LB	C_{max} SPT	MRE SPT	C_{max} ERD	MRE EDR
la13	20 x 5	1053	2088	98.29%	2009	90.79%
la14	20 x 5	1123	2090	86.11%	2034	81.12%
la15	20 x 5	1111	2113	90.19%	2043	83.89%
la16	10 x 10	892	1794	101.12%	1680	88.34%
la17	10 x 10	707	1519	114.85%	1358	92.08%
la18	10 x 10	842	1871	122.21%	1685	100.12%
la19	10 x 10	796	1556	95.48%	1440	80.90%
la20	10 x 10	857	1775	107.12%	1659	93.58%
Promedio			1669	104%	1501.25	82%

Cabe aclarar que los valores reportados para el LB o cotas inferiores para las instancias fueron tomadas de [21], los cuales se diseñaron para las primeras 20 instancias de edata. Es decir, son las mismas SDST_HUData, pero sin incluir los tiempos de alistamiento. Por lo tanto, es probable que estos LB estén muy por debajo del óptimo.

Un ejemplo del método de secuenciación de la regla de despacho ERD, empleando una pequeña instancia de tamaño 4 X 5, se encuentra en la Fig. 8, donde se identifica, la secuencia de cada uno de los productos, en cada una de las máquinas, con un *makespan* de 82 minutos.

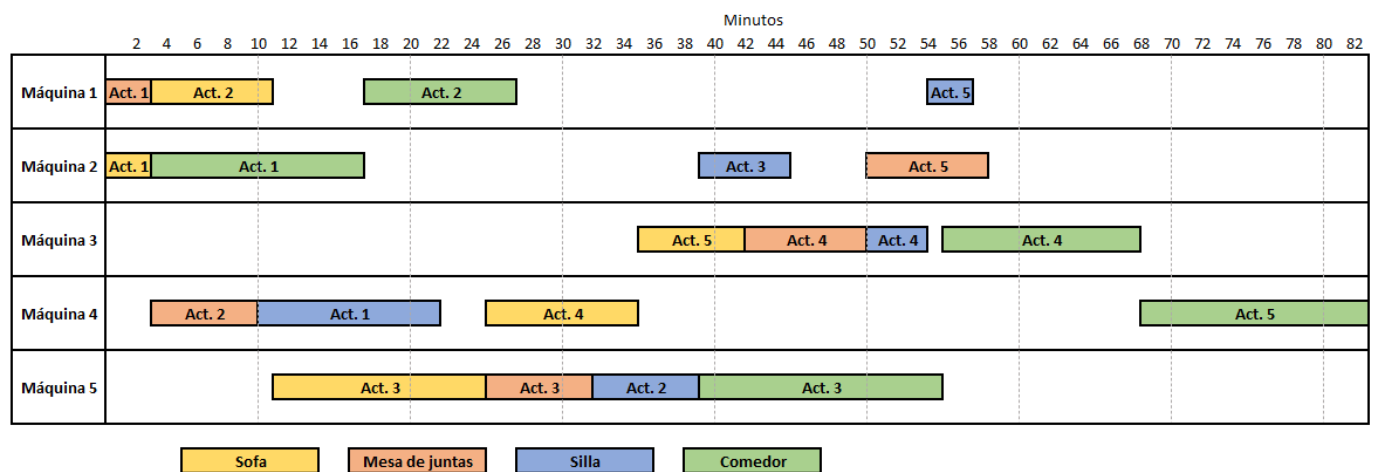


Fig. 8 Diagrama de Gantt de la regla de despacho ERD.

IV. ANALIZAR

A. Análisis de Oportunidad

En esta sección, se pretende mostrar que existe un problema de decisión de alta complejidad en el SDST-FJSP. Dicho lo anterior, se obtuvo tres posibles soluciones del espacio factible de la instancia *La01* del conjunto *SDST_HUData*, con el fin de mostrar como el *Makespan* depende de la forma en que se programen los trabajos. Sin pérdida de generalidad, en esta sección se presentarán tres reglas de despacho: ERD y SPT, descritas anteriormente, y, además, la regla LPT (*Longest Processing Time*) que consiste en programar los trabajos de acuerdo con el tiempo de procesamiento total de todas sus actividades (tiempo total del trabajo), iniciando de mayor a menor.

A continuación, en la TABLA IX, se presentan los resultados obtenidos al implementar cada una de las reglas de despacho mencionadas anteriormente en la instancia *La01*, la cual se compone de 10 trabajos y 5 máquinas (10 x 5). Posteriormente, se muestran los diagramas de Gantt correspondientes, donde las casillas de color gris, indicadas con ST, representan el tiempo de alistamiento dependiente de la secuencia, S_{rjl} , y las otras casillas el tiempo de procesamiento, $P_{\theta_{jmr}}$.

TABLA IX.
RESUMEN DE REGLAS DE DESPACHO

Regla de Despacho	Diagrama de Gantt	Secuencia para la programación	Makespan
ERD	Fig. 9	1, 2, 3, 4, 5, 6, 7, 8, 9, 10.	1225
SPT	Fig. 10	2, 3, 9, 5, 8, 1, 6, 4, 10, 7.	1391
LPT	Fig. 11; Error! No se encuentra el origen de la referencia.	7, 10, 4, 6, 1, 8, 5, 9, 3, 2.	1145

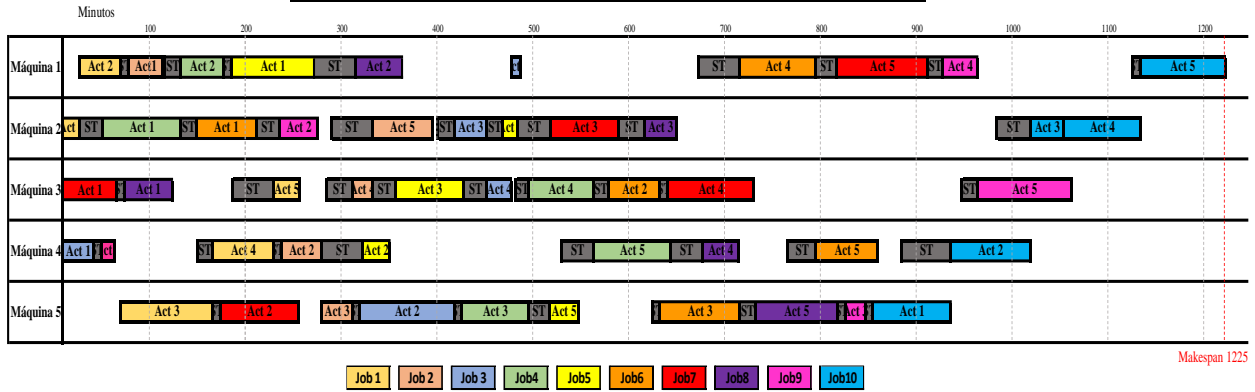


Fig. 9 Diagrama de Gantt de la Instancia La01 programada con regla de despacho ERD.

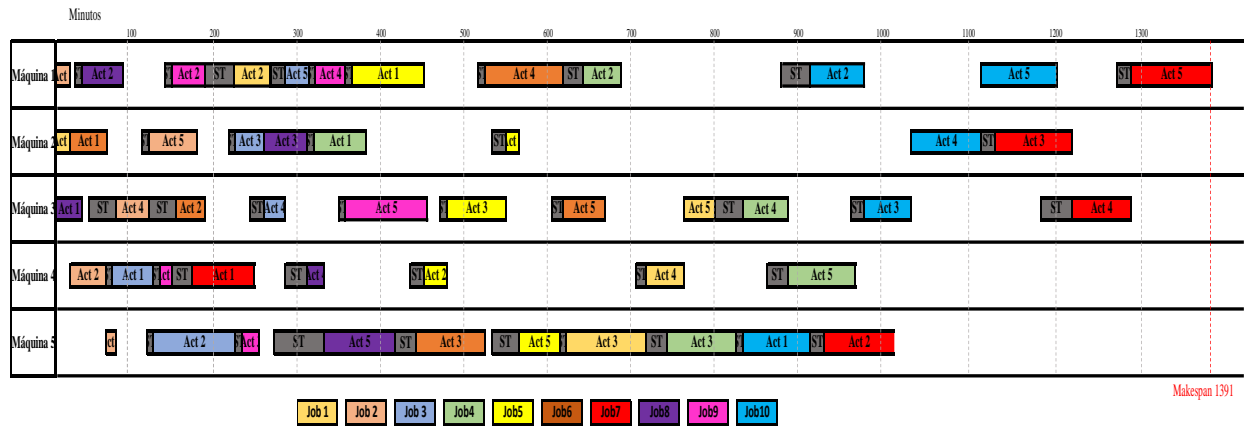


Fig. 10 Diagrama de Gantt de la Instancia La01 programada con regla de despacho SPT.

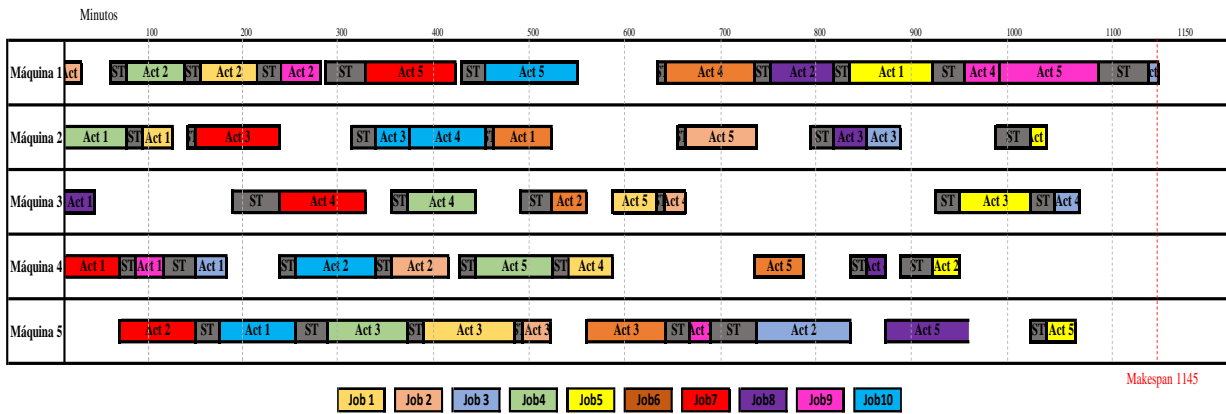


Fig. 11 Diagrama de Gantt de la Instancia La01 programada con regla de despacho LPT.

Los resultados anteriores dan un indicio del problema de decisión existente en el SDST-FJSP, pues se tiene cerca de $(n \times m)!$ opciones factibles, donde cada una tiene un *Makespan*, y se debe buscar entre las posibles soluciones la que proporciona el mínimo de ellos. Es decir, que para una instancia como la La01 con 10 trabajos y 5 máquinas se tiene un espacio de búsqueda de poco menos de $(10 \times 5)! = 3.04 \times 10^{64}$ soluciones factibles. Además, un pequeño cambio en el tamaño de la instancia se ve reflejado como un gran aumento en el espacio de solución, por ejemplo, suponga se tienen 11 trabajos, el número de soluciones factibles ascendería a aproximadamente $(11 \times 5)! = 1.27 \times 10^{73}$.

Adicionalmente, el FJSP al ser una extensión de JSP, se cataloga al igual que este último, como *NP-Hard* [22]. Sin embargo, cabe resaltar que el FJSP es más complejo pues además del problema de la secuenciación, cuenta con un subproblema de asignación, en donde se decide el modo de ejecución de algunas las actividades [23]. Más aún, al incorporar los tiempos de alistamiento dependientes de la secuencia S_{rjl} , el problema es fuertemente *NP-Hard*; incluso cuando se tiene sólo una máquina, salvo cuando los tiempos de alistamiento cumplen condiciones muy específicas [1]. Esta complejidad implica que el tiempo computacional usado en encontrar la solución óptima del problema no se encuentra aún acotado de forma polinomial, en palabras más sencillas, a medida que las instancias aumentan su tamaño, el tiempo computacional aumenta exponencialmente. Esto a su vez, hace que este tipo de problemas sea intratable para ser resuelto por métodos exactos, siendo necesario la implementación de métodos eficientes que permitan encontrar soluciones aproximadas (cercanas al óptimo) en un tiempo computacional razonable [24].

B. Revisión de literatura

Con el fin de comenzar a enrutar el proyecto, se debe determinar los diferentes métodos de solución para el problema objeto de estudio. Hay un campo muy amplio en cuanto a métodos se refiere, por lo que se realizará una comparación de los desempeños de los diferentes métodos de solución para enfocarse en aquel que presente mejores resultados con respecto a los demás métodos. En la TABLA X se describe de manera muy general la clasificación de los métodos de solución, empleados en la literatura científica, con sus respectivos ejemplos. En la TABLA XI se muestran los distintos casos de estudio que se observaron para evaluar y comparar los desempeños de cada uno tratándose de una forma mucho más detallada. Los criterios que se tuvieron en cuenta son las instancias que se usaron en determinado método, las dimensiones de estas instancias (número de trabajos y máquinas), el método de solución y sus respectivos resultados.

TABLA X.
CLASIFICACIÓN DE LOS MÉTODOS DE SOLUCIÓN

Métodos de solución	Definición general	Ejemplos
Método exacto	Método que garantiza la solución óptima por medio de la aplicación de técnicas matemáticas sujetas a características específicas (continuidad, linealidad, diferenciabilidad, etc.). Permiten desarrollar un procedimiento en base a teoremas matemáticos [25].	<ul style="list-style-type: none"> • Programación lineal • Ramificación y acotamiento • Programación dinámica

Métodos de solución	Definición general	Ejemplos
Métodos heurísticos	Es un método que se basa en la búsqueda de una solución, con una técnica basada en la experiencia que puede incluir acciones de búsqueda, descubrimiento, error y corrección. Así, se puede llegar a encontrar una solución que sea óptima o aproximada al óptimo en un espacio de posibilidades [26].	<ul style="list-style-type: none"> • Método basado en cuello de botella dinámico • Búsqueda local • Algoritmos <i>Greedy</i>
Métodos metaheurísticos	Método de solución para resolver problemas de optimización combinatoria en los cuales los heurísticos clásicos no son muy efectivos. Generalmente, están basados en conceptos derivados de la inteligencia artificial, la evolución biológica o mecanismos estadísticos, y en algunas ocasiones combinación de varios métodos (algoritmos híbridos), con el fin de abordar una amplia gama de problemas y tener una mayor flexibilidad [27].	<ul style="list-style-type: none"> • Búsqueda de vecindad variable (VNS) • Algoritmo genético (GA) • Búsqueda Tabú (TS)

TABLA XI.
RESUMEN DE LITERATURA DEL FJSP

Autores	Variante del problema	Instancias	Método de solución	Resultados
R. Tavakkoli-Moghaddam, M. Khalili & B. Naderi. 2008 [28]	SDST-JSP	Taillard con n y m así: (15, 20, 30, 50, 100 X 15, 20 respectivamente).	Este método metaheurístico, llamado EMSA, es una combinación de dos metaheurísticas: Mecanismo de tipo electromagnético (EM) y recocido simulado (SA). Fue comparado con IA, algoritmo genético (GA), algoritmo genético híbrido (HGA), SA, EMSA, algoritmo de mecanismo electromagnético (EMA)	El algoritmo propuesto EMSA, se compara con un IA, algoritmo genético (GA), algoritmo genético híbrido (HGA), SA. Obteniendo los mejores resultados entre los algoritmos con un índice de desviación relativa (RDI) del 9,47%.
Mohammad Saidi-Mehrabad; Parviz Fattahi. 2006 [29]	SDST-FJSP	(SSP, MSP y LSP) (2-15X3-6) experimentales para la industria de fabricación de piezas	Se propone un algoritmo de búsqueda tabú (TS) y lo comparan con el método de ramificación y acotamiento.	Resuelve muy cerca al óptimo los problemas de tamaño grande. Logra tiempos de CPU mejores en el 64% de las instancias. En el C_{max} , se desempeña igual o mejor en un 80% de las instancias
Guohui Zhang; Yifan Hu; Jinghe Sun; Wenqiang Zhang; 2020 [30]	Transportation Time, SDST-FJSP	FJSP Brandimarte (diferentes tamaños desde 10X6 hasta 20-15)	Algoritmo genético mejorado (IGA), con diversos métodos para mejorar la calidad y la diversidad de la población inicial, y un método cruzado con emparejamiento artificial para preservar las buenas soluciones y mejorarlas de manera efectiva.	El algoritmo se compara con un <i>Elitism GA</i> , <i>Enhanced GA</i> y <i>multi-objective grey wolf optimization (MOGWO)</i> . Los resultados del IGA propuesto son mejores que los de <i>Elitism GA</i> y <i>Enhanced GA</i> en la mayoría de los casos, y la mitad de los tres objetivos de optimización son mejores que MOGWO.
Miguel a. Gonzales, Camino R. Vela, Ramiro Varela. 2013 [21]	SDST-FJSP	SDST Hudata (diferentes tamaños desde 10X5 hasta 10X10)	Algoritmo genético híbrido con búsqueda tabú, con una estructura de vecindad incrustado (GA+TS).	Se compara el algoritmo híbrido (GA+TS), con un <i>Iterative Flattening Search (IFS)</i> , el GA y TS, obteniendo los siguientes MRE, respectivamente: 15.92%, 16.49%, 42.27% y 16.29%.
Ameni Azzouz; Meriem Ennigrou; Lamjed Ben Said. 2017 [31]	SDST-FJSP	SDST Hudata (diferentes tamaños desde 10X5 hasta 10X10)	Proponen un algoritmo híbrido basado en un algoritmo genético (GA) combinado con búsqueda local iterada (ILS). SAHA	Se compara el algoritmo propuesto SAHA, con un GA, un híbrido GA más un VNS y finalmente, con un SAHA Static, con sus respectivos MRE 34.78%, 38.69%, 37.42% y 38.89%.
Ameni Azzouz; Meriem Ennigrou; Lamjed Ben Said. 2017 [32]	SDST-FJSP	SDST Hudata (diferentes tamaños desde 10X5 hasta 10X10)	Algoritmo híbrido basado en el algoritmo genético (GA) y la búsqueda de vecindad variable (VNS) llamado HGA	El algoritmo propuesto HGA, se compara con un <i>Variable Neighborhood Search (VNS)</i> , TS y GA, a través de la desviación porcentual relativa (RPD) medida para comparar el rendimiento de los algoritmos, obteniendo un 0.54, 3.12, 4.15 y 1.40, respectivamente.
Ameni Azzouz; Meriem Ennigrou; Lamjed Ben Said. 2020 [33]	SDST-FJSP	SDST Hudata (diferentes tamaños desde 10X5 hasta 10X10)	Algoritmo evolutivo de dos niveles llamado (Bi-GTS)	El algoritmo propuesto Bi-GTS, se compara con un VNS, TS, GA y un <i>Hybrid genetic algorithm coupled with VNS algorithm (GTS)</i> . Los resultados de sus MRE respectivamente fueron: 8.17%, 37.31%, 20.35%, 32.61% y 12.76%.

Autores	Variante del problema	Instancias	Método de solución	Resultados
Sobeyko, Oleh Mönch, Lars. 2016 [34]	FJSP	FJSP Brandimarte MK (diferentes tamaños desde 10X6 hasta 20-15) + Conjunto de instancias de gran tamaño	Heurística de cuello de botella cambiante (SBH) con el enfoque de búsqueda local propuesto (ILS) y un enfoque de búsqueda de vecindad variable (VNS).	Para máquinas idénticas, el SBH supera en enfoques a ILS, para máquinas paralelas funcionan mejor los enfoques del ILS. Si hay suficiente tiempo de cómputo, el SBH, hibridado con enfoques LS, puede superar los enfoques basados en LS. Si no hay suficiente tiempo de cómputo, se debe elegir el enfoque de ILS.
Jun-Qing Li; Quan-Ke Pan; P. N. Suganthan. 2011 [35]	FJSP-STD	BRdata MK (diferentes tamaños, desde 10X6 hasta 20X15) Kacem instances (diferentes tamaños, desde 4X5 y 15X10)	Se propone una TS híbrido denominado TSPCB con una estructura de vecindad rápida para resolverlo. Además, enfoques y funciones disminuyen el tamaño de la estructura vecinal del problema y eliminan movimientos innecesarios e inviables. Se compara con una arquitectura genética aprendible (LEGA)	El algoritmo TSPCB supera al algoritmo de Kacem un 80%, al de GENACE en 50%, al de Brandimarte en un 90% y el algoritmo LEGA en un 77,7%. Para las instancias de 8x8 y 15x10, supera a optimización de enjambre de partículas (PSO)+ Recocido simulado (SA).
Mohsen Ziaee. 2013 [36]	FJSP	FJSP Brandimarte MK (10-20X4-15)	Se presenta una heurística simple y fácilmente extensible basada en un procedimiento constructivo. Se compara con 8 de los principales métodos	El algoritmo heurístico propuesto tiene un RPD promedio de 6.88 comparado con 0 para los otros algoritmos, su CPU es de 0.2 s en comparación con el resto (seguido de 4.978,80 s para el algoritmo de optimización de colonias de hormigas basado en el conocimiento KBACO).
Li-Ning Xing, Ying-Wu Chen, Peng Wang, Qing-Song Zhao, Jian Xiong. 2009 [37]	FJSP	Se implementan instancias específicas del problema con diferentes tamaños (4-20X5-13)	Mejora del algoritmo evolutivo (EA) desde una búsqueda de marco uniforme llamado Arquitectura de Búsqueda Heurística Basada en el Conocimiento (KBHSA). Se demuestra el rendimiento de esta arquitectura en la instanciación del KBACO	Los resultados muestran que el KBACO tiene un mejor <i>makespan</i> en 16 de las 31 instancias. En promedio, tiene un <i>makespan</i> menor que un KBACO sin programación en un 0,1311875 y un 0,0342 menor que un KBACO sin módulo de mejora de horario
Lei Wang, Jingcao Cai, Ming Li, Zhihu Liu. 2016 [38]	FJSP	Kacem y BRdata instances de diferentes tamaños (4,8,10,15X5,8,7,10)	Se propone una optimización mejorada de colonias de hormigas (IACO) para optimizar el <i>makespan</i> para FJSP, y para corregir la desventaja de la baja eficiencia computacional y el óptimo local.	Encuentra la mejor solución para el 10x10 T-FJSP en la segunda iteración en 5 s. IACO el mejor <i>makespan</i> igual o menor que el de otros algoritmos con casi las 10 Instancias BRdata. Supera a GA en un 30%, al KBACO en un 60%, al LEGA en un 70%, al TSPCB en un 40%, y es casi tan bueno como PVNS

La revisión de literatura realizada muestra un alto uso de metaheurísticas poblacionales, siendo los más utilizados los algoritmos genéticos, combinándolos con otros tipos de metaheurísticas, con el fin de mejorar su eficacia en la minimización del *makespan*. Utilizando diversos métodos para mejorar los componentes que hacen parte de los algoritmos genéticos, tales como: diversidad en la población inicial, mejoramiento en las estrategias de emparejamiento y selección. La utilización de los diversos métodos heurísticas y metaheurísticas, aunque no garantiza el resultado óptimo, puede encontrar una muy buena solución en un tiempo de cómputo razonable para instancias de mediano tamaño y algunas metaheurísticas robustas en instancias de un mayor tamaño. Este hecho, se evidencia en [29], donde se compara un método exacto ramificación y acotamiento, con una metaheurística llamada búsqueda Tabú, evidenciando que a medida que el tamaño de las instancias aumenta, el tiempo computacional del método exacto aumenta exponencialmente, mientras que en el algoritmo TS, se mantiene en unos rangos aceptables de tiempo. No obstante, este hecho será comprobado mediante la formulación matemática del problema SDST-FJSP, y posteriormente mediante experimentación, validando del desempeño de los métodos exactos.

Los algoritmos metaheurísticos más eficientes utilizados en la revisión de la literatura fueron el GA, TS y el IFS (*Iterative Flattening Search*), obteniendo un MRE por debajo del 20%. Sin embargo, los algoritmos híbridos como es el caso del algoritmo evolutivo de dos niveles, basado en una combinación de un GA y la búsqueda de vecindad variable (VNS) obtiene un mejor *makespan* con un MRE del 8.76%, y el híbrido del GA + VNS de un solo nivel, también obtiene un MRE bajo del 12.76%. Otra combinación eficaz es el GA más un TS, obteniendo un MRE de un 15.92%. De esta forma, se observa que la utilización de metaheurísticas resulta el método más eficaz y adecuada para el tamaño de instancias y el problema abordado por este proyecto.

Por lo tanto, de acuerdo con la investigación realizada, se observa dentro de las metaheurísticas más utilizadas para nuestro tipo de proyecto son: GA, TS, VNS y los algoritmos híbridos más eficientes: GA+TS y GA+VNS. De igual forma se realizará un análisis a través de un modelo matemático

C. Exploración de ideas y selección de alternativa

La exploración de ideas se llevará a cabo de acuerdo con las métodos más importantes identificadas en la sección anterior, teniendo en cuenta los criterios de evaluación descritos en la TABLA XII, realizando una comparación por pares de cada una de las alternativas y criterios, mediante un juicio de expertos realizado mediante una encuesta de calificación (ANEXO 4). Posteriormente se realiza un análisis multicriterio, con el fin de calcular las ponderaciones de los criterios y finalmente la alternativa con la mayor ponderación.

TABLA XII.
CRITERIOS DE CALIFICACIÓN

Criterio	Calificación				
	1	2	3	4	5
Tiempo computacional	Demasiado	Mucho	Moderado	Poco	Muy poco
Facilidad de implementación	Muy difícil	Difícil	Regular	Fácil	Muy fácil
Precisión en el resultado	Impreciso	Poco preciso	Moderado	Preciso	Muy preciso
Tamaño de las instancias	2x2	5x5	10x5	10x20	mayor a 10x20

Las alternativas seleccionadas como el GA, TS, VNS, y sus algoritmos híbridos con base en GA. Se diferencian en sus componentes de ejecución, siendo algunos evolutivos y otros de búsqueda local, estas diferencias y conceptualización se describe a continuación definiendo cada uno de los métodos seleccionados.

Algoritmo genético (GA):

El algoritmo genético (GA), inicialmente fue inspirado por el fenómeno natural “la teoría biológica de la evolución” formulado por Charles Darwin a mediados del siglo XIX [39]. Y desarrollado finalmente, en la Universidad de Michigan por el Dr. John Holland, sus estudiantes y amigos en la década de 1960. El GA es un método en el cual cada “individuo” de una nueva generación representa una solución factible codificada en un cromosoma con distintos algoritmos / parámetros para

ser evaluados por una función de aptitud. Los operadores de GA son mutación (el cambio de un bit del cromosoma elegido al azar) y cruce (el intercambio de fragmentos de cromosoma elegidos al azar) [40]. Los GA han sido aplicados a distintos problemas de optimización, algunos ejemplos son: el enrutamiento de cables, el diseño del circuito en los diseños VLSI (*Very Large Scale Integration*) y los problemas de programación del taller.

Búsqueda de vecindad variable (VNS):

Como propuesta para resolver problemas de optimización muy difíciles, los autores Mladenovis y Hansen propusieron este método el cual consiste en realizar cambios sistemáticos de las estructuras del vecindario dentro del procedimiento de búsqueda [41], tanto en la fase de descenso para poder encontrar un mínimo local, como en la fase de perturbación para salir del valle correspondiente [42]. Se han desarrollado diversas extensiones tales como: *a variable neighborhood descent* (VND), VNS básico (BVNS), VNS reducido (RVNS), entre otros. Diseñado originalmente para la solución aproximada de problemas de optimización combinatoria, se amplió para abordar programas enteros mixtos, programas no lineales y programas enteros no lineales enteramente mixtos recientemente [43].

Búsqueda Tabú (TS):

En 1986 el TS fue desarrollado en por Glover [44], señalando la aleatorización controlada de un SA, con el fin de escapar de los óptimos locales. Convirtiéndose en la década de los 90, en un algoritmo muy popular para resolver problemas de optimización de manera aproximada. Su principal característica es el uso de la memoria, que almacena información relacionada con el proceso de búsqueda [45].

Un TS, explora de manera determinística todo un vecindario de soluciones, cuando alcanza un óptimo local la búsqueda continúa seleccionando un candidato peor que la situación actual. Es decir, la mejor solución de un vecindario se selecciona, aunque este no sea mejor que la de la solución actual. De esta manera se pueden generar ciclos, visitando repetitivamente distintas soluciones. Para evitar los ciclos, se genera una estrategia llamada lista tabú, almacenando los distintos atributos de los movimientos realizados. Sin embargo, este hecho puede perder algo de la información sobre la memoria de búsqueda, rechazando soluciones que aún no se han generado. Finalmente, un TS, contiene mecanismos avanzados para hacer frente a la intensificación y diversificación de la búsqueda, almacenando en la memoria de mediano plazo, las soluciones elites, y al largo plazo, almacenando información sobre soluciones ya visitadas [45].

Metaheurísticas híbridas:

Los métodos metaheurísticos híbridos combinan los componentes de varios algoritmos. La principal motivación para fusionar dos algoritmos es aprovechar el carácter complementario que pueden tener las diferentes estrategias, siendo esta a su vez la clave en el diseño que permite maximizar el rendimiento en la resolución de problemas de optimización complejos [46]. Además, pueden presentarse múltiples combinaciones de metaheurísticas con otras metaheurísticas, métodos exactos, *constraint programming* o *machine learning*. En [45] describen las principales combinaciones y algunos usos de ellas:

- Metaheurísticas con Metaheurística.
- Metaheurísticas con métodos exactos de enfoques de programación matemática, se usan principalmente en investigación de operaciones.
- Metaheurísticas con *constraint programming*, se desarrollan principalmente en inteligencia artificial.
- Metaheurísticas con *machine learning* y técnicas de minería de datos.

Finalmente, con las diversas alternativas seleccionadas y los criterios de selección identificados, se aplica un proceso de jerarquía analítica (AHP), siendo un enfoque básico para la toma de decisiones [47]. Este proceso está realizado en el ANEXO 5, mediante el diseño de [48], determinando las ponderaciones para cada uno de los criterios, y luego por alternativas, con base a la calificación promedio dada por el juicio de expertos descrita en el ANEXO 4. De esta manera se obtiene la TABLA XIII, con los pesos porcentuales de importancia para cada uno de los criterios.

TABLA XIII.
PONDERACIONES DE LOS CRITERIOS DE DECISIÓN

Criterio	Prioridades
Tiempo computacional	8%
Facilidad de implementación	36%
Precisión en el resultado	36%

Criterio	Prioridades
Tamaño de las instancias	20%

Después de realizar el proceso AHP por criterios, se realiza de igual forma en cada una de las alternativas el proceso de jerarquización analítica, obteniendo la TABLA XIV TABLA XIII, con los porcentajes de prioridad, siendo la alternativa seleccionada, el método del algoritmo genético con un porcentaje del 23%.

TABLA XIV
PONDERACIONES FINALES POR ALTERNATIVA

Alternativa	Ponderación final
GA	23%
TS	21%
VNS	18%
GA+TS	19%
GA+VNS	19%

D. *Objetivos*

Objetivo general:

Diseñar un método de solución para el problema de Flexible Job Shop con tiempos de alistamientos dependientes de la secuencia, con el fin de disminuir el Makespan.

Objetivos específicos:

1. Identificar los componentes de diseño y ejecución de la alternativa seleccionada mediante la revisión sistemática de literatura.
2. Diseñar un modelo matemático o algoritmo heurístico o metaheurístico, que dé solución al problema SDST-FJSP.
3. Validar el método de solución propuesto, a través de la medición de los indicadores de desempeño y compararlo con otros métodos en la literatura en base a las instancias SDST-HUdata.
4. Estandarizar la implementación del método propuesto mediante la elaboración de un manual de usuario.

E. *Plan de trabajo (PdT)*

El desarrollo de los objetivos del proyecto y su respectivo desglose de actividades se describe mediante la TABLA XV, el cual enmarca el plan de trabajo del equipo Programadores ALPHA, con el fin de lograr el cumplimiento del objetivo general.

TABLA XV.
PLAN DE TRABAJO

Objetivo General						
Diseñar un método de solución para el problema de Flexible Job Shop con tiempos de alistamientos dependientes en la secuencia, con el fin de disminuir el Makespan.						
Objetivos específicos	Actividades	Área IISE	Herramienta	#	Entregable	Fecha
Identificar los componentes de diseño y ejecución de la alternativa seleccionada mediante la revisión sistemática de literatura.	Desarrollar un estudio sistemático de la alternativa seleccionada, categorizando sus diversos componentes	11. Ingeniería de la información	Clasificación de la información.	C	Informe descriptivo de la alternativa seleccionada	15/09/2020
	Describir cada componente, en base al problema de estudio FJSP.	11. Ingeniería de la información	Conceptos de visualización de datos	D		
	Sintetizar la información clave de la alternativa seleccionada, con sus componentes de diseño y ejecución	11. Ingeniería de la información	Informes de variables clave	D		
Diseñar un modelo matemático o algoritmo heurístico o metaheurístico, que dé solución al problema SDST-FJSP	Efectuar la formulación matemática del problema SDST-FJSP.	2. Análisis e investigación de operaciones	Enfoques de modelado	A	Diseño del método de solución propuesto	22/10/2020
	Realizar experimentación del modelo matemático, identificando su desempeño de resolución.	2. Análisis e investigación de operaciones	Aplicaciones LP	B		
	Desarrollar el diseño de la alternativa seleccionada de acuerdo con las condiciones y restricciones del sistema.	2. Análisis e investigación de operaciones	Algoritmo con restricciones	H		
Validar el método de solución propuesto, a través de la medición de los indicadores de desempeño y compararlo con otros métodos en la literatura en base a las instancias SDST-HUdata.	Evaluar el rendimiento de la alternativa seleccionada en cada una de las instancias SDST-HUdata	11. Ingeniería de la información	Análisis descriptivo	R	Informe de desempeño del método seleccionado	02/11/2020
	Comparar el desempeño de la alternativa seleccionada con diversos métodos de la literatura del FJSP	11. Ingeniería de la información	Evaluación de alternativas	L		

Objetivo General						
Diseñar un método de solución para el problema de Flexible Job Shop con tiempos de alistamientos dependientes en la secuencia, con el fin de disminuir el Makespan.						
Objetivos específicos	Actividades	Área IISE	Herramienta	#	Entregable	Fecha
Estandarizar la implementación del método propuesto mediante la elaboración de un manual de usuario.	Modificar el ingreso de las variables de entrada a través de una <i>dashboard</i>	11. Ingeniería de la información	Diseño de <i>dashboards</i>	D	Manual de usuario	18/11/2020
	Realizar una guía detallada para el ingreso de las variables y la interpretación de la solución	11. Ingeniería de la información	Especificaciones del programa	N		

Una descripción más detallada del PdT, se encuentra en el ANEXO 6, realizando la gestión del cronograma en MS Project, determinando el tiempo total y la duración de cada uno de los objetivos y actividades. De igual forma, se determina los recursos necesarios y las actividades críticas.

V.DISEÑAR

A. Desarrollo del diseño de la solución

Para el diseño de la propuesta de solución al SDST-FJSP, se tuvo en cuenta dos vertientes principales: la primera de ellas consta de un modelo matemático el cual pretende, más allá de brindar una solución al problema, servir en la comprensión de los supuestos y condiciones que lo caracterizan. Por otro lado, se diseñó un algoritmo genético, con el cual se busca proporcionar una mejor solución a la obtenida por el método ERD (método actual), en un tiempo aceptable.

Inicialmente, se describe el modelo matemático y en el ANEXO 7 se encuentra el respectivo código en AMPL. Luego, se aborda cada una de las etapas que componen el algoritmo genético y el proceso de implementación para proponer una solución al programador de los trabajos.

1) Modelo Matemático

Anteriormente, se hizo la definición del SDST-FJSP como un problema que considera un conjunto de N trabajos, los cuales se componen a su vez de un conjunto de operaciones que se deben ejecutar en una de las m máquinas disponibles, según los modos de ejecución posibles. En esta versión del FJSP se parte de los siguientes supuestos:

- Los tiempos de alistamiento que dependen de la secuencia son conocidos y constantes.
- Los tiempos de procesamiento, pueden o no, variar dependiendo del modo de ejecución.
- Las actividades de los trabajos deben ejecutarse sin interrupciones.
- Cada máquina puede procesar máximo un trabajo a la vez.
- Todos los trabajos están disponibles para ejecutarse en el tiempo 0.
- La recirculación está permitida, es decir que un mismo trabajo puede pasar más de una vez por una máquina.

No obstante, para hacer una caracterización formal del problema, se formuló el modelo matemático MILP (por sus siglas en inglés *Mixed Integer Linear Programming*) que se presenta a continuación, tomando como referencia el modelo presentado en [29].

Definición de conjuntos

- J : conjunto de trabajos $\{0, \dots, N\}$

- O : conjunto de operaciones y máquinas $\{1, \dots, m\}$

A pesar de que el problema del FJSP considera la posibilidad de que el número de operaciones y el número de máquinas sea distinto, e incluso que el número de operaciones varíe de un trabajo a otro, las instancias SDST-HUData consideran que hay tantas operaciones como máquinas y adicionalmente, que el número de operaciones por Job es el mismo. Por lo tanto, sin pérdida de generalidad, se considera un único conjunto para representarlas a ambas.

Definición de parámetros

De aquí en adelante, usaremos la notación R_{jh} para hacer referencia a la operación $h \in O$ del trabajo $j \in J$.

- S_{ijk} : tiempo de alistamiento en la máquina $i \in O$ al pasar del trabajo $j \in J$ al trabajo $k \in J$.
- P_{ijh} : tiempo de procesamiento de la operación $h \in O$ del trabajo $j \in J$ en la máquina $i \in O$.
- $a_{ijh} = \begin{cases} 1, & \text{si la operación } R_{jh} \text{ puede ejecutarse en la máquina } i \in O. \\ 0, & \text{en caso contrario.} \end{cases}$
- M : representa un número muy grande, en comparación con la magnitud de la instancia.

Definición de variables de decisión

- $Y_{ijh} = \begin{cases} 1, & \text{si la operación } R_{jh} \text{ se ejecuta en la máquina } i \in O. \\ 0, & \text{en caso contrario.} \end{cases}$
- $X_{ijhkl} = \begin{cases} 1, & \text{si la operación } R_{jh} \text{ precede a la operación } R_{kl} \text{ en la máquina } i \in O. \\ 0, & \text{en caso contrario.} \end{cases}$
- T_{jh} : tiempo de inicio de la operación R_{jh} .
- F_{jh} : tiempo de finalización de la operación R_{jh} .
- C : *makespan*.

Modelo algebraico de programación lineal entera mixta

Minimizar: $z = C$

Sujeto a:

$$F_{jm} \leq C, \quad \forall j \in J \quad (1)$$

$$T_{0h} = 0, \quad \forall h \in O \quad (2)$$

$$T_{jh} + Y_{ijh}(P_{ijh}) \leq F_{jh}, \quad \forall i, h \in O, \forall j \in J, \quad (3)$$

$$F_{jh} \leq T_{j,h+1}, \quad \forall j \in J, \forall h \in O \setminus \{m\} \quad (4)$$

$$\sum_{i \in O} Y_{ijh} = 1, \quad \forall j \in J, \forall h \in O \quad (5)$$

$$Y_{ijh} \leq a_{ijh}, \quad \forall i, h \in O, \forall j \in J \quad (6)$$

$$\sum_{j \in J} \sum_{h \in O} X_{ijhkl} = Y_{ikl}, \quad \forall i, l \in O, \forall k \in J \setminus \{0\} \quad (7)$$

$$\sum_{k \in J} \sum_{l \in O} X_{ijhkl} \leq Y_{ijh}, \quad \forall i, h \in O, \forall j \in J \quad (8)$$

$$X_{ijhjh} = 0, \quad \forall i, h \in O, \forall j \in J \quad (9)$$

$$F_{jh} + S_{ijk} \leq T_{kl} + (1 - X_{ijhkl})M, \quad \forall j \in J, \forall k \in J \setminus \{0\}, \forall h, l, i \in O \quad (10)$$

$$F_{jh} \leq T_{jh} + P_{ijh} + (1 - Y_{ijh})M, \quad \forall j \in J, \forall h, i \in O \quad (11)$$

$$X_{ijhkl} \text{ Binarias}, \quad \forall j, k \in J, \forall h, l, i \in O \quad (12)$$

$$Y_{ijh} \text{ Binarias}, \quad \forall j \in J, \forall i, h \in O \quad (13)$$

$$T_{jh} \geq 0, F_{jh} \geq 0, \quad \forall j \in J, \forall h \in O \quad (14)$$

$$C \geq 0. \quad (15)$$

La Ecuación **¡Error! No se encuentra el origen de la referencia.** se encarga de asignar el *makespan* como el tiempo de finalización del último trabajo que se completa. Por eso, C es mayor o igual a los tiempos de finalización de la última operación de cada trabajo, ya que, al minimizar, será asignado al más grande de los tiempos de finalización. La Ecuación **¡Error! No se encuentra el origen de la referencia.** inicializa todas las máquinas, empleando un trabajo ficticio y forzándolo a iniciar todas sus operaciones en el tiempo 0. De este modo, las demás ecuaciones, en especial las que relacionan la precedencia, pueden operar correctamente. Cabe aclarar que, para este trabajo ficticio, cada una de las operaciones se hace en una máquina diferente, los tiempos de procesamiento son cero, al igual que los tiempos de alistamiento. La Ecuación **¡Error! No se encuentra el origen de la referencia.**, establece que las operaciones de cada trabajo solo pueden darse por finalizadas, una vez se completa el tiempo de procesamiento. Mientras que, la Ecuación **¡Error! No se encuentra el origen de la referencia.** garantiza que se cumpla la precedencia de las operaciones de un mismo trabajo.

Por otro lado, las Ecuaciones **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** garantizan que únicamente un modo de ejecución sea seleccionado y que dicho modo se encuentre entre los permitidos para cada operación, respectivamente. Luego, las Ecuaciones **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** determinan que solo puede haber precedencia de dos operaciones en una máquina, si ambas se estén ejecutando en ella. También, es preciso garantizar que la operación de un trabajo no puede precederse a ella misma, así que se incorpora la Ecuación **¡Error! No se encuentra el origen de la referencia.** para garantizarlo.

Adicionalmente, la ecuación **¡Error! No se encuentra el origen de la referencia.** desempeña un papel fundamental en esta variación del FJSP, pues establece que si la operación R_{jh} precede a la operación R_{kl} , entonces la operación R_{kl} no podrá iniciar, hasta tanto la operación R_{jh} termine su procesamiento y cumpla con el alistamiento entre operaciones. Del mismo modo, la ecuación **¡Error! No se encuentra el origen de la referencia.** ha sido parte de los cambios que se implementaron frente al modelo base presentado en [29], con el objetivo de garantizar que el tiempo de finalización de cada una de las operaciones corresponda al tiempo de inicio más el tiempo de procesamiento.

Finalmente, las Ecuaciones **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** identifican a las variables X y Y como binarias. Mientras que, las Ecuaciones **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** corresponden a las típicas restricciones de no negatividad para las variables F, T y C .

2) Algoritmo genético

Un algoritmo genético tiene como objetivo buscar soluciones de alta calidad en el espacio factible, tomando como referencia la teoría de la supervivencia de los individuos con mejores aptitudes de una población. Para ello se atraviesan por 5 etapas: Inicialización de la población, selección de la población a cruzar, operador de cruce, operador de mutación y reemplazo de la población inicial. Si bien es cierto que todos los AG pasan por las mismas fases, cada una puede implementarse mediante diferentes técnicas y empleando diversos parámetros, según las preferencias del investigador. En esta sección, se describen en detalle las técnicas empleadas en cada una de las etapas del AG, así como la codificación que se implementó para describir cada uno de los individuos (soluciones factibles).

Codificación

La codificación hace referencia a la forma como se va a representar la información de cada uno de los individuos. En este caso, cada uno de ellos se compone dos cromosomas, representados por vectores, y estos a su vez se componen por genes, es decir, posiciones del vector. El primer cromosoma contiene la información relacionada con la secuencia de las operaciones, mientras que, el segundo representa la asignación de máquinas. Además, cabe resaltar que existe una correspondencia entre los cromosomas; cada posición del vector de máquinas está relacionada a la operación del trabajo que se ejecuta en el vector de secuencia, en la misma posición.

El vector de secuencia es un vector que contiene en cada posición un número entero l tal que $1 \leq l \leq n$, donde n , cómo se ha mencionado anteriormente, representa el número total de trabajos. Adicionalmente, cada vez que aparece un trabajo, se hace referencia a una operación de dicho trabajo, siempre respetando la precedencia en las operaciones. Es decir, si un número de trabajo aparece por segunda vez, entonces se está representando la segunda operación del trabajo.

Por otro lado, el vector asignación de máquinas representa la posición elegida en el vector de modos. Por lo tanto, se compone de números enteros que van desde 0 y el número máximo que puede tomar, depende de cuantos modos de ejecución tiene la operación correspondiente.

Por ejemplo, suponga que el vector P representa el vector de secuencia y que el vector A, el cromosoma de asignación de máquinas:

$$P = [2, 1, 1, 2]$$

$$A = [0, 1, 0, 2]$$

Entonces, la primera posición del vector A, donde aparece un 0, indica que se debe emplear el primer modo de ejecución para operación 1 del segundo trabajo 2 (Ver correspondencia con el vector P). Luego, la segunda posición del vector A, sugiere que se debe seleccionar el segundo modo de ejecución para la operación 1 del trabajo 1.

Inicialización de la población

La inicialización de la población es la primera fase que se ejecuta en el AG y solo se hace una vez durante todo el proceso. Esta etapa cumple un papel fundamental, pues como su nombre lo indica, es el punto de partida para la búsqueda de la solución y, por tanto, se busca que esta sea lo más diversa posible, es decir que se busca que los individuos sean diferentes entre sí.

En un estudio presentado por Guohui Zhang [30], se proponen diferentes métodos para realizar el proceso de inicialización de la población, de los cuales se seleccionó el “método de generación aleatoria” pues genera una solución inicial diversa e imparcial. En este caso, se deben generar dos vectores, uno para la secuencia y otro para la asignación del modo de ejecución como ya se mencionó arriba, para ello, se genera un gen del vector de secuencia e inmediatamente después, el gen correspondiente del vector de máquinas. Así pues, se genera un número entero aleatorio l tal que $0 \leq l < n$, donde n representa el número total de trabajos, dicho número corresponde a una operación del trabajo l . Note que, cada trabajo debe repetirse tantas veces como operaciones tiene, por lo tanto, este es un factor determinante en la generación del aleatorio.

Una vez se crea un nuevo gen en el vector de secuencias, se procede a crear el gen del vector de asignación de máquinas. Aquí se genera nuevamente un número entero aleatorio, pero en este caso primero se determina cuantos modos tiene la operación del trabajo asignado en el vector de secuencias, pues el número aleatorio debe ir desde cero hasta dicha cantidad menos uno, teniendo en cuenta que representará una posición del vector de modos.

Este proceso debe repetirse hasta completar los dos vectores del individuo y posteriormente, crear tantos individuos como se haya configurado para el tamaño de la población N .

Por lo tanto, algunos factores que deben tenerse en cuenta son:

- El vector de secuencia y el vector de asignación deben ser del mismo tamaño, pues hay una correspondencia entre ellos. Dicho tamaño debe ser igual a la suma del número de operaciones de cada trabajo.
- Cada trabajo debe repetirse tantas veces como operaciones tiene, pues cada vez que aparece en el vector de secuencia, representa una operación.

- Cada gen en el cromosoma de máquinas debe ser estrictamente menor a la cantidad de modos de ejecución que tiene la operación del trabajo que corresponde.

Selección de cromosomas a cruzar

La segunda fase en el AG corresponde a la selección de los cromosomas a cruzar, esto es seleccionar una muestra de la población que va a ser candidata a ser padres y posteriormente, la generación de parejas que se emplearán en el proceso de cruce. Para ello, se implementó el método *Tournament*, pues es el más empleado por diferentes autores en la literatura [49].

El *Tournament* consiste en elegir aleatoriamente una muestra de tamaño K individuos de la población inicial. Para ello, primero debe elegirse aleatoriamente el tamaño de la muestra K , y luego, generar un vector de aleatorios sin reemplazo de tamaño K , el cual determinará cuales son los individuos candidatos a ser padres. Posterior a ello, se ejecuta un torneo entre los elementos de la muestra para conformar las parejas, empleando el método de la ruleta, descrito por [50].

El método de la ruleta consiste en calcular la probabilidad que tiene un individuo en ser seleccionado de acuerdo con el valor *fitness* o también llamado función de aptitud. Esta, hace referencia a que tan buena o apta es un individuo, como solución al problema. En este caso, entre más pequeño sea el *Makespan*, mejor será la solución. Por lo tanto, el *fitness* de un individuo se define como el inverso multiplicativo de su *Makespan*; es decir, $f(Z_i) = \frac{1}{C_{max}}$. Así pues, entre mayor sea $f(Z_i)$, mejor es la solución y por tanto, debe tener una mayor probabilidad de ser seleccionado como padre. A continuación, se describe el método de la ruleta para la selección de pares:

1. Calcular la función de aptitud $f(Z_i)$ para cada individuo Z_i .
2. Calcular la función de aptitud para la muestra seleccionada K .

$$Ft = \sum_i^K f(Z_i) \quad (16)$$

donde C es la población seleccionada para realizar el cruce.

3. Calcular la probabilidad P_i para cada individuo Z_i .

$$P_i = \frac{f(Z_i)}{Ft} \quad i = 1, 2, 3, \dots, K \quad (17)$$

4. Calcular probabilidad acumulada Pa_i para cada individuo (Z_i).

$$Pa_i = \sum_j^i P_j \quad i = 1, 2, 3, \dots, K \quad (18)$$

5. Generar un numero aleatorio s , tal que $s \in [0,1]$.
6. Se selecciona el individuo Z_i , tal que, $Pa_{i-1} < s \leq Pa_i$.

Los pasos 5 y 6 deben repetirse hasta seleccionar $\frac{K}{2}$ parejas de padres. Note que un padre puede repetirse en dos parejas distintas, pero no en la misma pareja.

En la Fig. 12 se muestra un ejemplo de selección de individuos para conformar las parejas. Se tiene una población inicial $Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ de tamaño $N = 10$ individuos. Mediante *Tournament*, primero, se genera un numero aleatorio K , tal que $K \in [1, N=10]$ y $K \in \mathbb{N}$. Aquí, $K = 5$, es decir que, se seleccionará solamente 5 individuos de Q . Para proceder a seleccionar los candidatos a ser padres, se genera el vector aleatorio sin reemplazo, donde cada elemento n_i es menor o igual a N , es decir, $n_i \in [1, N=10]$ y $n_i \in \mathbb{N}$. Es este caso, los individuos seleccionados son: 2, 9, 7, 4, 1.

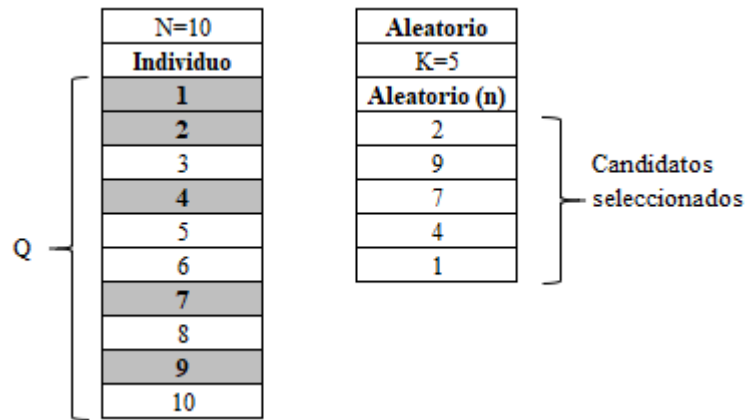


Fig. 12 Selección de candidatos como padres

Luego, para la selección de las parejas, se implementa el método de la ruleta. En la Fig. 13, se selecciona el primer padre de la primera pareja. Primero, como se indicó arriba, se calcula el *fitness* para los candidatos teniendo en cuenta el *Makespan* de cada uno de ellos. Después, se calcula la probabilidad de ser seleccionado P_i y la probabilidad acumulada P_{ai} obteniendo la porción de la ruleta para cada candidato. Consecutivamente, se genera el número aleatorio s , en este caso, igual a 0.7. Por lo tanto, el primer padre debe ser el individuo 4. Así, la generación de aleatorios s continua hasta que sean seleccionados los 4 padres de las 2 parejas.

N=10	Aleatorio					Aleatorio
Individuo	K=5					$s=0.7$
1	Aleatorio (n)	Cmax	f(Zi)	Pi	Pai	Intervalo
2	2	25	0.040	0.188	0.188	[0.000 , 0.188)
3	9	23	0.043	0.204	0.392	[0.188 , 0.392)
4	7	28	0.036	0.168	0.559	[0.392 , 0.559)
5	4	18	0.056	0.261	0.820	[0.559 , 0.820)
6	1	26	0.038	0.180	1.000	[0.820 , 1.000]
7	Ft		0.213			

Parejas
 $5/2=2.5 \approx 2$

Fig. 13 Ejemplo método de la ruleta selección de parejas.

Operador de cruce

El operador de cruce hace referencia al proceso de reproducción biológica que busca crear nuevos individuos mediante el cruce de información (genes) de los cromosomas. Debido a que el individuo se conforma de dos cromosomas, se implementa el cruce únicamente en el vector de secuencia, empleando el método de Cruce de Preservación de Precedencia (PPX). Mientras que, el vector de asignación de máquinas se genera nuevamente, de forma aleatoria, para el hijo.

El método de Preservación de Precedencia (PPX) respeta el orden absoluto de genes resultando en una perfecta preservación de las relaciones de precedencia entre genes [51]. Este método consiste en generar números enteros aleatorios $\{0,1\}$, de forma sucesiva, con el fin de definir el padre del que debe extraerse cada gen. Al extraer un gen de un padre, automáticamente, se elimina del padre contrario. En la Fig. 14 se muestra un ejemplo. En este caso, el vector de selección indica seleccionar los primeros dos genes del padre 1, consecutivamente, estos dos genes seleccionados son eliminados del padre 2. Después, el vector de selección indica elegir los tres primeros genes posibles a seleccionar del padre 2 siendo estos 4, 3, 2. Estos pasos se repiten hasta completar el último gen.

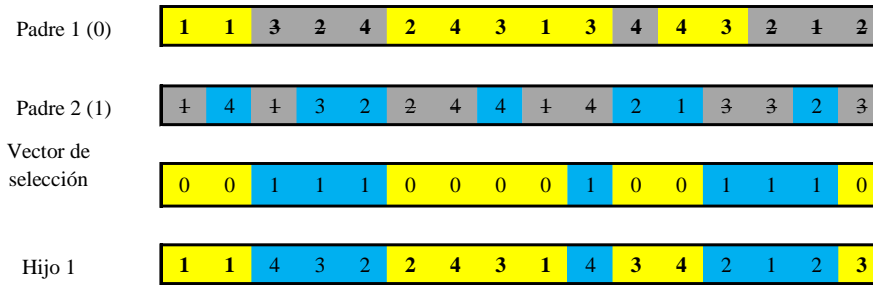


Fig. 14 Ejemplo operador de cruce con PPX

Operador de mutación

Con el fin de aleatorizar y diversificar la búsqueda de soluciones por medio de los individuos de la población, se procede a realizar el operador de mutación. Este permite “modificar” los cromosomas de ciertos individuos para fomentar la exploración, es decir, ampliar un poco el espacio de búsqueda y tener soluciones diferentes. Se conocen varios métodos para llevar a cabo este operador, sin embargo, se usará el método denominado “mutación de intercambio” o “*swap mutation*” expuesto por Muhammad et al., [52].

La mutación de intercambio consiste en asignar a cada individuo de la población una probabilidad de mutación $P_m \in [0,1]$. Generalmente, la probabilidad de mutación es baja con el fin de que solo una pequeña muestra de la población sea mutada; pues de ser muy alta, podría alterar significativamente la mejora de la búsqueda. Luego, se genera un número aleatorio para cada individuo entre 0 y 1, para realizar el operador de mutación siempre y cuando el valor obtenido sea igual o inferior a la probabilidad de mutación.

Si un individuo es seleccionado para mutar, entonces se debe aplicar el siguiente procedimiento.

1. Se seleccionan dos posiciones al azar.
2. Intercambiar estas dos posiciones del vector de secuencias.
3. Generar nuevamente todo el vector de modos de forma aleatoria.

En la Fig. 15 se muestra gráficamente el proceso de mutación para un individuo.

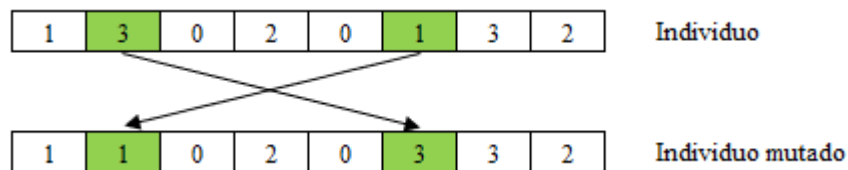


Fig. 15 Operador de mutación con *Swap Mutation*

Reemplazo de la población inicial

Para el reemplazo de la población inicial, se selecciona $20\%N$ individuos de los mejores *makespan* del total de la población, incluyendo padres e hijos. Después, se selecciona el 80% restante, aleatoriamente y sin reemplazo.

B. Validación del diseño propuesto

El GA propuesto se ejecutó con un tamaño de población inicial de 100, una probabilidad de mutación del 5% y un porcentaje de reemplazo del 20%. La compilación del algoritmo y del modelo matemático se realizó en un PC (*Personal Computer*) de sistema operativo Windows 10, procesador Intel Core i5-5200U 2.20 Ghz con 8 GB de memoria RAM. Cinco veces se ejecutaron para cada una de las instancias con un tiempo aproximado de 2,5 minutos. Los resultados se

muestran en la TABLA XVI, indicando el *makespan* alcanzado y su respectivo MRE, observando que el Algoritmo Genético obtiene un MRE promedio mucho menor que el Modelo matemático, siendo la diferencia del 24%. Por otra parte, se puede observar que tanto el Modelo y el Algoritmo obtienen un mejor MRE que el SPT o ERD, evidenciando la utilidad de los métodos utilizados.

TABLA XVI
VALIDACIÓN DEL DISEÑO

Instancia	Tamaño	LB	<i>Cmax</i> SPT	MRE SPT	<i>Cmax</i> ERD	MRE EDR	<i>Cmax</i> Mínimo GA	MRE GA	<i>Cmax</i> Modelo Matemático	MRE Modelo Matemático
la01	10 x 5	609	1391	128.4%	1225	101.1%	749.4	23.1%	798.2	31.1%
la02	10 x 5	655	1173	79.1%	1081	65.0%	813.4	24.2%	774.6	18.3%
la03	10 x 5	550	1063	93.3%	966	75.6%	716	30.2%	717	30.4%
la04	10 x 5	568	1212	113.4%	1061	86.8%	736.8	29.7%	736	29.6%
la05	10 x 5	503	1199	138.4%	905	79.9%	649	29.0%	660	31.2%
la06	15 x 5	833	1699	104.0%	1367	64.1%	1066.2	28.0%	1321.6	58.7%
la07	15 x 5	762	1550	103.4%	1327	74.1%	1046.4	37.3%	1239.4	62.7%
la08	15 x 5	845	1766	109.0%	1398	65.4%	1085.4	28.4%	1251.6	48.1%
la09	15 x 5	878	1760	100.5%	1555	77.1%	1097.8	25.0%	1312.4	49.5%
la10	15 x 5	866	1873	116.3%	1486	71.6%	1080.6	24.8%	1229	41.9%
la11	20 x 5	1087	2025	86.3%	1788	64.5%	1456.4	34.0%	2078.8	91.2%
la12	20 x 5	960	1863	94.1%	1958	104.0%	1285	33.9%	1974.2	105.6%
la13	20 x 5	1053	2088	98.3%	2009	90.8%	1380.6	31.1%	1857	76.4%
la14	20 x 5	1123	2090	86.1%	2034	81.1%	1424.2	26.8%	2280.6	103.1%
la15	20 x 5	1111	2113	90.2%	2043	83.9%	1541	38.7%	1855.8	67.0%
la16	10 x 10	892	1794	101.1%	1680	88.3%	1136.2	27.4%	1344.6	50.7%
la17	10 x 10	707	1519	114.9%	1358	92.1%	935.6	32.3%	956.6	35.3%
la18	10 x 10	842	1871	122.2%	1685	100.1%	1066.8	26.7%	1247.4	48.1%
la19	10 x 10	796	1556	95.5%	1440	80.9%	1058.4	33.0%	1099.4	38.1%
la20	10 x 10	857	1775	107.1%	1659	93.6%	1056	23.2%	1279.6	49.3%
Promedio			1669	104.1%	1501.25	82.0%	1069.06	29.3%	1300.69	53.3%

Otra validación realizada, es la variación del tiempo de ejecución a uno mucho más extenso, mostrado en la TABLA XVII con un tiempo de ejecución por instancia de 24 minutos. Los resultados muestran en el caso del algoritmo genético una mejora del 2.9% y en el Modelo matemático del 25.7%, comparado con el tiempo de ejecución anterior de 2.5 minutos. Además, la diferencia entre el modelo y el algoritmo es mucho menor siendo de 1.2%. Es decir, que el modelo matemático requiere mucho más tiempo para igualar los resultados que el algoritmo genético consigue en un menor tiempo.

TABLA XVII
VALIDACIÓN DISEÑO 24 MIN DE EJECUCIÓN

Instancia	Tamaño	LB	<i>Cmax</i> Mínimo GA	MRE GA	<i>Cmax</i> Modelo Matemático	MRE Modelo Matemático
la01	10 x 5	609	735	20.7%	751	23.3%
la02	10 x 5	655	783	19.5%	772	17.9%
la03	10 x 5	550	697	26.7%	678	23.3%

Instancia	Tamaño	LB	<i>C</i> _{max} Mínimo GA	MRE GA	<i>C</i> _{max} Modelo Matemático	MRE Modelo Matemático
la04	10 x 5	568	719	26.6%	690	21.5%
la05	10 x 5	503	630	25.2%	622	23.7%
la06	15 x 5	833	1040	24.8%	1018	22.2%
la07	15 x 5	762	1016	33.3%	991	30.1%
la08	15 x 5	845	1054	24.7%	1099	30.1%
la09	15 x 5	878	1082	23.2%	1071	22.0%
la10	15 x 5	866	1061	22.5%	1107	27.8%
la11	20 x 5	1087	1439	32.4%	1497	37.7%
la12	20 x 5	960	1263	31.6%	1307	36.1%
la13	20 x 5	1053	1372	30.3%	1562	48.3%
la14	20 x 5	1123	1404	25.0%	1409	25.5%
la15	20 x 5	1111	1510	35.9%	1565	40.9%
la16	10 x 10	892	1107	24.1%	1086	21.7%
la17	10 x 10	707	910	28.7%	925	30.8%
la18	10 x 10	842	1024	21.6%	1046	24.2%
la19	10 x 10	796	1021	28.3%	1012	27.1%
la20	10 x 10	857	1050	22.5%	1017	18.7%
Promedio			1045.85	26.4%	1061.25	27.6%

En la TABLA XVIII se observa la variación de la probabilidad de mutación del algoritmo genético, realizado con un número de 1000 generaciones para cada instancia. Los resultados muestran un incremento en el MRE después del 15%, indicando que los resultados empeoran si se ejecuta el algoritmo genético con una probabilidad de mutación alta. Gráficamente los resultados por generaciones se muestran en la Fig. 16, Fig. 17 y Fig. 18 para las probabilidades de mutación de 5%, 30% y 100% respectivamente, donde se observa como aumenta la inestabilidad de los resultados al ser una probabilidad de mutación alta, esto afecta directamente el resultado final, ya que inicialmente el algoritmo debería tener una tendencia descendente y posteriormente estabilizarse en un mínimo, como se observa en la Fig. 16 con una probabilidad de mutación del 5%

TABLA XVIII
VARIACIÓN PROBABILIDAD DE MUTACIÓN.

Probabilidad de Mutación	MRE promedio
5%	29.5%
10%	29.1%
15%	30.2%
20%	30.6%
30%	31.1%
50%	32.0%
70%	32.2%
100%	32.7%

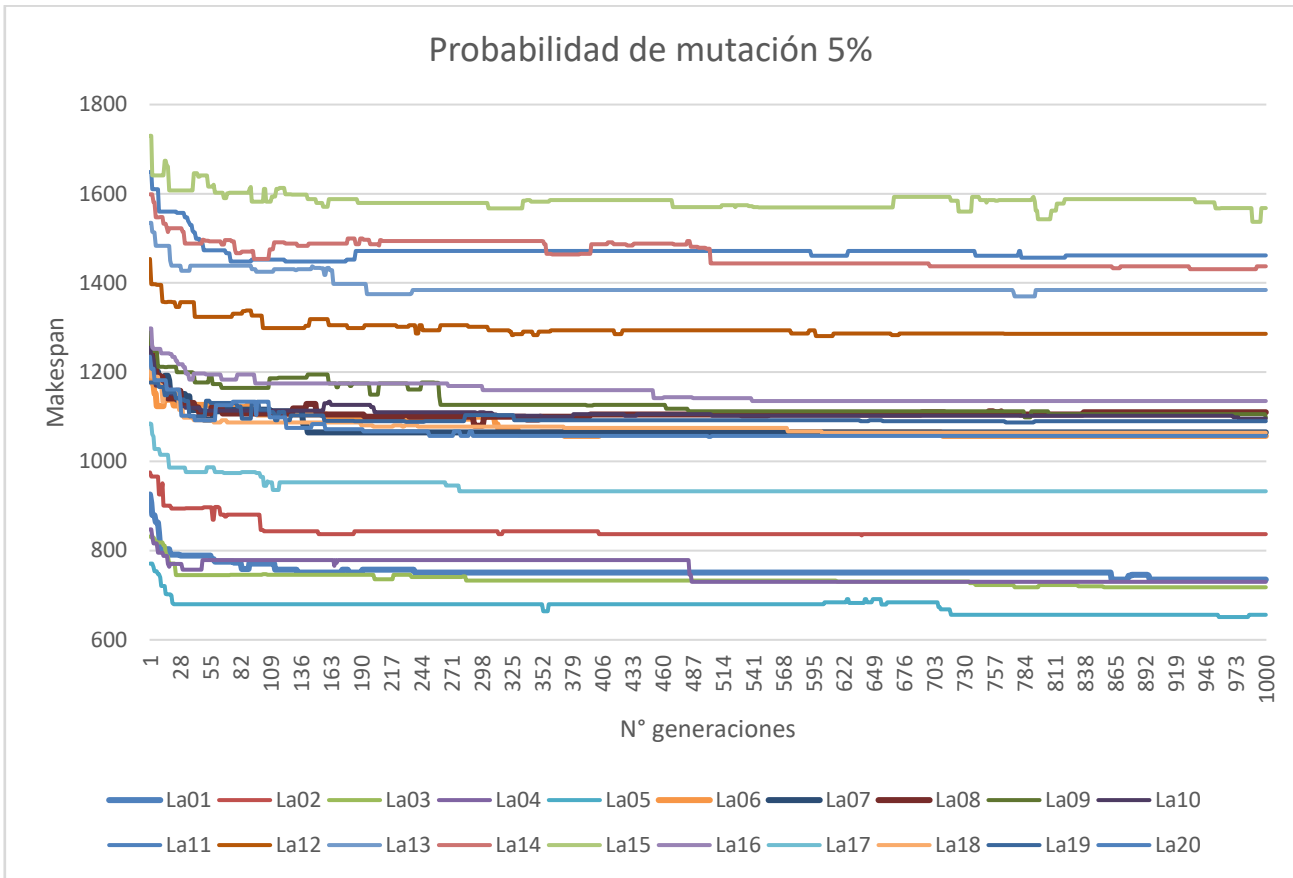


Fig. 16. Resultados por instancia con una probabilidad de mutación del 5%.

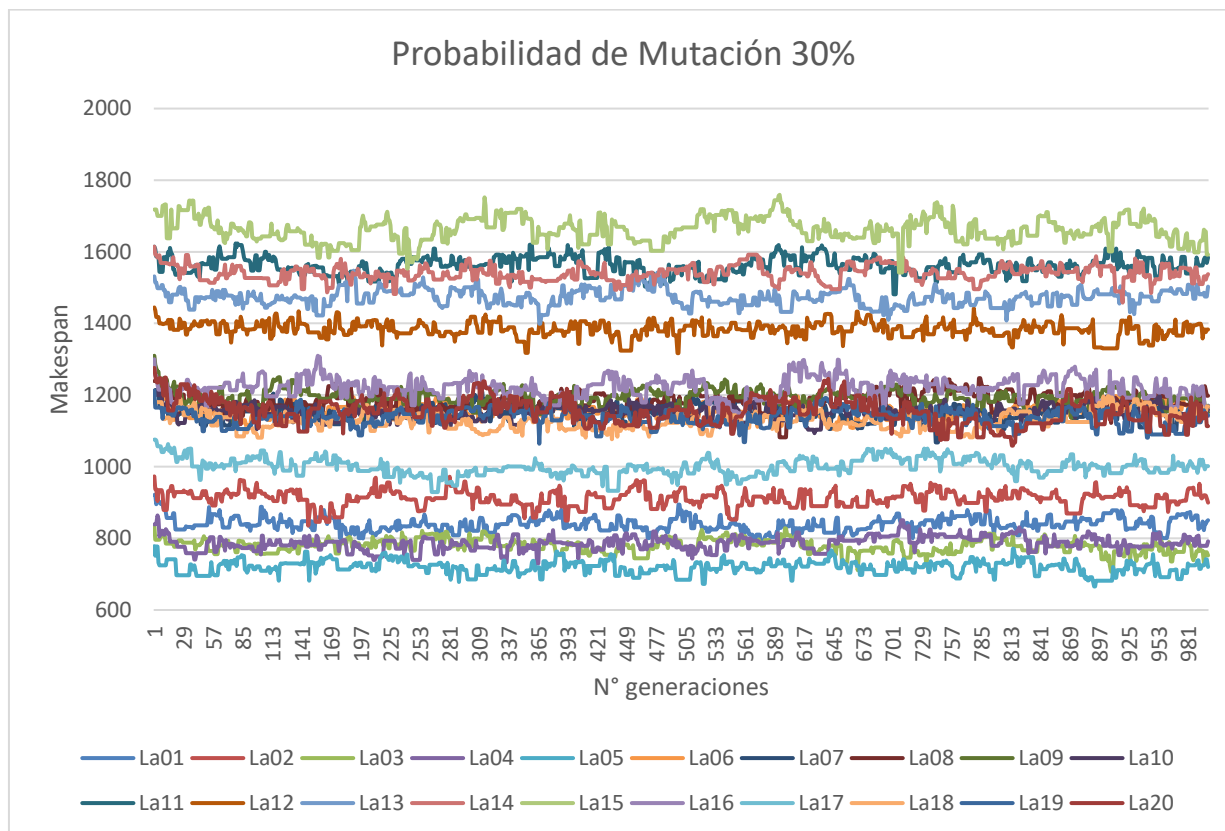


Fig. 17. Resultados por instancia con una probabilidad de mutación del 30%.

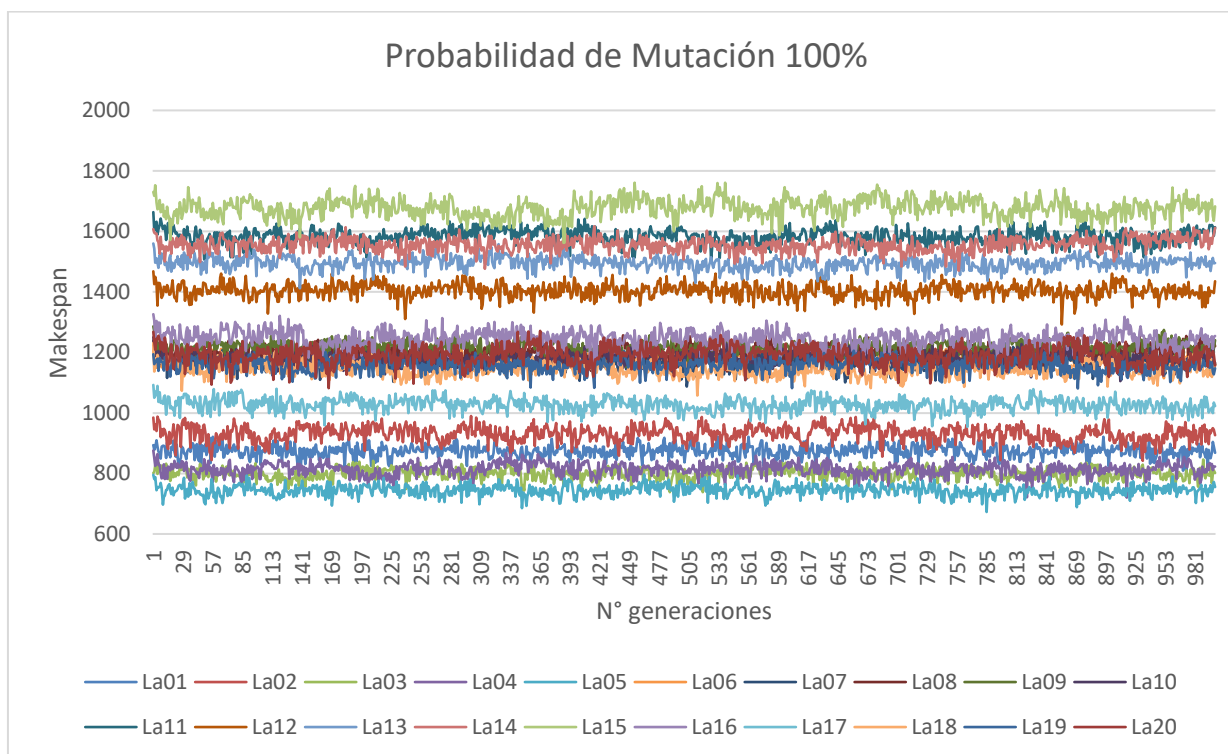


Fig. 18. Resultados por instancia con una probabilidad de mutación del 30%.

En la TABLA XIX se realizó una comparación de los *Makespan* obtenidos para las instancias SDST-HUData por [31], en donde se aplica un algoritmo híbrido autoadaptable para resolver problemas de trabajos flexibles con tiempo de configuración dependiente de la secuencia y en [11] se aplica la búsqueda de acoplamiento iterativo para el mismo problema. Se puede observar que el algoritmo y el modelo matemático propuesto supera al SAHA.

TABLA XIX
COMPARACIÓN DE RESULTADOS

Instancia	Tamaño	LB	<i>Cmax</i> SAHA [31]	<i>Cmax</i> IFS [11]	<i>Cmax</i> GA PROPUESTO	<i>Cmax</i> Modelo Matemático	MRE Modelo Matemático	MRE GA
la01	10 x 5	609	751	726	735	751	23.3%	20.70%
la02	10 x 5	655	839	749	783	772	17.9%	19.5%
la03	10 x 5	550	749	652	697	678	23.3%	26.7%
la04	10 x 5	568	747	673	719	690	21.5%	26.6%
la05	10 x 5	503	660	603	630	622	23.7%	25.2%
la06	15 x 5	833	1106	950	1040	1018	22.2%	24.8%
la07	15 x 5	762	1105	916	1016	991	30.1%	33.3%
la08	15 x 5	845	1096	954	1054	1099	30.1%	24.7%
la09	15 x 5	878	1142	1002	1082	1071	22.0%	23.2%
la10	15 x 5	866	1067	977	1061	1107	27.8%	22.5%
la11	20 x 5	1087	1488	1265	1439	1497	37.7%	32.4%
la12	20 x 5	960	1336	1088	1263	1307	36.1%	31.6%
la13	20 x 5	1053	1404	1255	1372	1562	48.3%	30.3%
la14	20 x 5	1123	1448	1292	1404	1409	25.5%	25.0%
la15	20 x 5	1111	1519	1298	1510	1565	40.9%	35.9%
la16	10 x 10	892	1249	1012	1107	1086	21.7%	24.1%
la17	10 x 10	707	1036	864	910	925	30.8%	28.7%
la18	10 x 10	842	1139	985	1024	1046	24.2%	21.6%
la19	10 x 10	796	1205	956	1021	1012	27.1%	28.3%
la20	10 x 10	857	1173	997	1050	1017	18.7%	22.5%
Promedio					1045.85	1061.25	27.6%	26.4%

VERIFICAR

A. Medición de los impactos

Impacto Financiero

Desde el análisis del impacto financiero, la creación del algoritmo genético por medio de la decodificación, tomo una cantidad de tiempo. Teniendo en cuenta que en Colombia el salario de un ingeniero industrial es de \$ 2'000.000 al mes en promedio, o \$12.308 por hora en promedio, según informa el portal de empleo “*talent*” [53].

Para la creación del GA, los integrantes del grupo dispusieron de 4 horas diarias durante 3 semanas en promedio (descansando los domingos), dando un total de 72 horas de trabajo.

Teniendo el valor de la hora laboral de un ingeniero industrial en Colombia, y teniendo la cantidad de horas que se ocuparon para realizar el GA obtenemos un valor de

$$12.308 \frac{\$}{\text{hora laboral}} * 72 \text{ horas} = 886.176 \$$$

Obteniendo un total de \$ 886.176 en cuanto a hora hombre de la realización del GA. Para la implementación de la solución, sería el único costo en el que se incurriría, pues el programa para realizar el algoritmo genético es Visual Studio, y este se puede descargar gratuitamente por medio de la web. Ahora bien, para la implementación de la solución del modelo matemático, igualmente existen programas gratuitos para correr los modelos luego de ser creados. Por medio de NEOS Solver con sus respectivos servidores, o simplemente descargando el programa de AMPL para el equipo o el computador. Por lo tanto, también se realizará el costo de su creación.

Aunque el modelo solo se está implementado para verificar inviabilidad de este, igualmente hace parte del proceso para la creación del algoritmo, por lo tanto, la implementación de la solución también incluye la creación del modelo matemático. Para este se tomaron 2 semanas (descansando los domingos) y 1 día, es decir, unos 13 días, en los que cada uno de los días fueron utilizados 4 horas de trabajo para la creación de este, dando como resultado:

$$13 \text{ días} * 4 \text{ horas} = 52 \text{ horas}$$

Tomando el mismo salario de un ingeniero industrial, para la creación del modelo matemático se tiene que:

$$12.308 \frac{\$}{\text{hora laboral}} * 52 \text{ horas} = 640.016 \$$$

Por lo tanto, teniendo en cuenta el tiempo invertido en el método de solución del problema del FJSP-STD, la implementación tendría un costo de:

$$886.176 \$ + 640.016 \$ = 1'526.192 \$$$

1'526.192 \$ costaría la implementación de la solución.

Ahora bien, en cuanto a los ahorros, se tomarán algunas máquinas industriales principales para obtener los consumos de cada una de ellas, con el fin de cuantificar monetariamente el consumo de energía y posteriormente calcular el ahorro energético que se presentaría en el caso de que se implemente la solución. Para eso, se tomará como referencia el artículo de la Universidad Politécnica de Catalunya [54].

Las máquinas que expone el artículo son:

- Horno de fundición con un consumo energético de 50 kW
- Cortadora- Fresadora con un consumo energético de 12 kW
- Taladradora con un consumo energético de 6 kW
- Torno CNC con un consumo energético de 11 kW
- Máquina de bobinar con un consumo energético de 10 kW
- Máquina de barnizar con un consumo energético de 35 kW
- Máquina de calar paquetes con un consumo energético de 35 kW

- Máquina embaladora con un consumo de 6 kW

Se entiende por kW una unidad de potencia determinada de una máquina o dispositivo para que pueda funcionar. Es la cantidad de Joules (trabajo) por segundo (tiempo). Ahora, se debe introducir el concepto de kWh, la cual es una unidad de energía que se utiliza por una máquina o dispositivo que requiere de 1000 vatios (kW) al ponerlo a funcionar durante 1 hora. Por lo tanto, si por ejemplo una máquina requiere de 2000 vatios para funcionar, en media hora habría logrado consumir 1 kWh. Dicho esto, las unidades de kW expuestas anteriormente por cada una de las máquinas se expresarán en el tiempo que tomará consumir 1 kWh:

- Horno de fundición: 50 kW = 0,02 horas (1,2 minutos) para consumir 1 kWh
- Cortadora – Fresadora: 12 kW = 0,083 horas (4,98 minutos) para consumir 1 kWh
- Taladradora: 6 kW = 0,16 horas (9,6 minutos) para consumir 1 kWh
- Torno CNC: 11 kW = 0,090 horas (5,4 minutos) para consumir 1 kWh
- Máquina de bobinar: 10 kW = 0,1 horas (6 minutos) para consumir 1 kWh
- Máquina de barnizar: 35 kW = 0,0286 horas (1,716 minutos) para consumir 1 kWh
- Máquina de calar paquetes: 35 kW = 0,0286 horas (1,716 minutos) para consumir 1 kWh
- Máquina embaladora: 6 kW = 0,16 horas (9,6 minutos) para consumir 1 kWh

Según VALORA ANALITIK, fundamentándose con datos de XM (Compañía de Expertos en Mercados), el precio del kWh para este 2020 está en 500\$ [55]. Además, la comisión de regulación de energía y gas (CREG) [56], proporcionó un tarifario para enero del 2020, en el que especifica que el valor del kWh para el sector industrial oscila entre los 500\$ y 650\$ (según su contribución). Se tomará entonces el valor de:

$$\frac{650\$ + 500\$}{2} = 575\$$$

Se tiene que para cada uno de los tiempos en los que las máquinas alcanzan a consumir 1 kWh, en costos serían 575 \$ COP.

Según los resultados anteriores TABLA XVI, en promedio el Makespan antes de implementar la mejora tenía un valor de 1501,25 unidades de tiempo. Para realizar la comparación del ahorro financiero que se obtiene al implementar la solución, se supondrá que las unidades de tiempo son horas. Al no saber los intervalos en los que las máquinas estarán en espera (pues no es un valor cuantificable), se supondrá la continuidad del máximo funcionamiento de las máquinas, con el fin de observar el ahorro energético de la implementación de la solución. Así, el consumo energético de las máquinas en pesos colombianos sería:

- Horno de fundición: (1501,25 hr / 0,02 hr) * 575 \$ = 43'160.937,5 \$
- Cortadora – Fresadora: (1501,25 hr / 0,083 hr) * 575 \$ = 13'692.123,5 \$
- Taladradora: (1501,25 hr / 0,16 hr) * 575 \$ = 5'395.117,2 \$
- Torno CNC: (1501,25 hr / 0,090 hr) * 575 \$ = 12'627.180,55 \$
- Máquina de bobinar: (1501,25 hr / 0,1 hr) * 575 \$ = 8'632.187,5 \$
- Máquina de barnizar: (1501,25 hr / 0,0286 hr) * 575 \$ = 30'182.473,8 \$
- Máquina de calar paquetes: (1501,25 hr / 0,0286 hr) * 575 \$ = 30'182.473,8 \$

- Máquina embaladora: $(1501,25 \text{ hr} / 0,16 \text{ hr}) * 575 \$ = 5'395.117,2 \$$

Dando un total de 149'267.611,05 \$ COP en una industria en la que todas estas máquinas funcionen continuamente durante 1501,25 horas a su máxima capacidad.

Haciendo la implementación de la solución, el Makespan en promedio es de 1.069,06 horas. Por lo tanto, el costo del gasto energético de las máquinas sería de:

- Horno de fundición: $(1.069,06 \text{ hr} / 0,02 \text{ hr}) * 575 \$ = 30'735.475 \$$
- Cortadora – Fresadora: $(1.069,06 \text{ hr} / 0,083 \text{ hr}) * 575 \$ = 7'406.138,5 \$$
- Taladradora: $(1.069,06 \text{ hr} / 0,16 \text{ hr}) * 575 \$ = 3'841.934,4 \$$
- Torno CNC: $(1.069,06 \text{ hr} / 0,090 \text{ hr}) * 575 \$ = 6'830.105,55 \$$
- Máquina de bobinar: $(1.069,06 \text{ hr} / 0,1 \text{ hr}) * 575 \$ = 6'147.095 \$$
- Máquina de barnizar: $(1.069,06 \text{ hr} / 0,0286 \text{ hr}) * 575 \$ = 21'493.339,16 \$$
- Máquina de calar paquetes: $(1.069,06 \text{ hr} / 0,0286 \text{ hr}) * 575 \$ = 21'493.339,16 \$$
- Máquina embaladora: $(1.069,06 \text{ hr} / 0,16 \text{ hr}) * 575 \$ = 3'841.934,4 \$$

Para un total de 101'789.361,17 \$ COP en la misma industrial al hacer la implementación de la solución.

Evidentemente se ve un ahorro de 47'478.249,88 \$, es decir, $1 - (101'789.361,17 \$ / 149'267.611,05 \$) = 0,3180 \%$

Luego de implementar la solución, los costos energéticos se han disminuido en un 31,80 %

Impacto Ambiental

Teniendo los valores del consumo energético de las anteriores máquinas en unidades de kWh, se procede a verificar la afectación de ambiental de la implementación de la solución, en valores de emisiones de CO2 por cada kWh consumido. Según el periódico económico, empresarial y financiero de Colombia La República [57], afirma que el factor de emisión por cada kWh en Colombia en emisiones de CO2 es de 164,38 gramos. Por lo tanto, para el caso de continuar con el funcionamiento de una industria con las máquinas experimentales, antes de implementar la solución, se obtiene que:

- Horno de fundición: $(1501,25 \text{ hr} / 0,02 \text{ hr}) * 164,38 \text{ g CO}_2 = 12'339.524 \text{ g CO}_2$
- Cortadora – Fresadora: $(1501,25 \text{ hr} / 0,083 \text{ hr}) * 164,38 \text{ g CO}_2 = 2'973.379 \text{ g CO}_2$
- Taladradora: $(1501,25 \text{ hr} / 0,16 \text{ hr}) * 164,38 \text{ g CO}_2 = 1'542.441 \text{ g CO}_2$
- Torno CNC: $(1501,25 \text{ hr} / 0,090 \text{ hr}) * 164,38 \text{ g CO}_2 = 2'742.117 \text{ g CO}_2$
- Máquina de bobinar: $(1501,25 \text{ hr} / 0,1 \text{ hr}) * 164,38 \text{ g CO}_2 = 2'467.905 \text{ g CO}_2$
- Máquina de barnizar: $(1501,25 \text{ hr} / 0,0286 \text{ hr}) * 164,38 \text{ g CO}_2 = 8'629.038 \text{ g CO}_2$
- Máquina de calar paquetes: $(1501,25 \text{ hr} / 0,0286 \text{ hr}) * 164,38 \text{ g CO}_2 = 8'629.038 \text{ g CO}_2$
- Máquina embaladora: $(1501,25 \text{ hr} / 0,16 \text{ hr}) * 164,38 \text{ g CO}_2 = 1'542.441 \text{ g CO}_2$

Para un total de 40'865.882 g CO2 en el transcurso de 1501,25 horas

Para el caso de que se implemente la solución, se obtiene que:

- Horno de fundición: $(1.069,06 \text{ hr} / 0,02 \text{ hr}) * 164,38 \text{ g CO}_2 = 8'787.139 \text{ g CO}_2$
- Cortadora – Fresadora: $(1.069,06 \text{ hr} / 0,083 \text{ hr}) * 164,38 \text{ g CO}_2 = 2'117.383 \text{ g CO}_2$
- Taladradora: $(1.069,06 \text{ hr} / 0,16 \text{ hr}) * 164,38 \text{ g CO}_2 = 1'098.392 \text{ g CO}_2$
- Torno CNC: $(1.069,06 \text{ hr} / 0,090 \text{ hr}) * 164,38 \text{ g CO}_2 = 1'952.697 \text{ g CO}_2$
- Máquina de bobinar: $(1.069,06 \text{ hr} / 0,1 \text{ hr}) * 164,38 \text{ g CO}_2 = 1'757.428 \text{ g CO}_2$
- Máquina de barnizar: $1.069,06 \text{ hr} / 0,0286 \text{ hr}) * 164,38 \text{ g CO}_2 = 6'144.852 \text{ g CO}_2$
- Máquina de calar paquetes: $(1.069,06 \text{ hr} / 0,0286 \text{ hr}) * 164,38 \text{ g CO}_2 = 6'144.852 \text{ g CO}_2$
- Máquina embaladora: $(1.069,06 \text{ hr} / 0,16 \text{ hr}) * 164,38 \text{ g CO}_2 = 1'098.392 \text{ g CO}_2$

Para un total de 29'101.136 g CO2 en el transcurso de 1.069,06 horas debido a la disminución del Makespan. Por lo tanto, se presenta un impacto ambiental positivo, permitiendo que se evite la emisión de 11'764.746 g CO2 al medio ambiente, es decir, una disminución del 28,8% de las emisiones que normalmente se estarían proporcionando al medio ambiente.

Difusión de la información:

Con el fin de hacer conocer la idea, se propone hacer una difusión de la información por medio de la Cámara y Comercio de Cali, en la cual se les permite a diferentes empresas proporcionar y publicitar diversos cursos virtuales en la sección de “Asiste a capacitaciones, eventos y convocatorias” de la página oficial de la Cámara y Comercio de Cali. Dentro de esta sección se encuentra la modalidad de “Campus Virtual”, el cual proporciona una cantidad de cursos sobre diferentes temas. Por medio de este campus se propone publicar por medio de un *webinar* la información sobre la funcionalidad de la propuesta al “diseño de la solución del problema de FJSP-STD”, demostrando su implementación, su manual de uso (la proporción de la información necesaria para su funcionamiento) y sus mejoras dentro de un proceso productivo que no use nuestra propuesta de solución al problema mencionado.

De esta manera se podría brindar la información a empresas que estén interesadas en obtener la idea, con el fin de mejorar sus procesos productivos.

B. Estandarización de la solución – POE'S (plan de control)

Para dar asistencia a los usuarios sobre cómo deben hacer uso del algoritmo genético (GA) al momento de ingresar la información pertinente para su correcta ejecución y la configuración del tiempo en que se desea una respuesta por parte del GA se presenta el manual de usuario en el ANEXO 9.

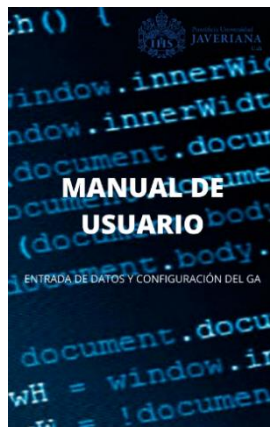


Fig. 19 Manual de Usuario.

C. Conclusiones

En este trabajo de investigación se presenta un método de solución para el problema de FJSP-STD, mediante el uso de heurísticas o metaheurísticas, acompañado con la realización de un modelo matemático con el fin de comparar la eficiencia de ambos métodos para el problema estudiado. Inicialmente, se realizó una revisión sistemática de la literatura y se encontró que los principales algoritmos implementados por parte de los investigadores son GA, TS y VNS. Sin embargo, al hacer un estudio comparativo, bajo los criterios de tiempo computacional (8%), facilidad de implementación (36%), precisión en el resultado (36%) y tamaño de las instancias (20%), se observó que el GA obtenía mejores resultados frente a los otros algoritmos.

En el desarrollo del algoritmo genético se implementaron las siguientes partes: Población inicial, operador de cruce (método *Tournament*), operador de mutación (método *Swap Mutation*) y el operador de selección. Los resultados evidencian una mejora que superan las expectativas en los indicadores de desempeño: para el GA, el Makespan disminuyó un 28,6% frente al ERD, evidenciando que, si supera la meta establecida en los objetivos, del 10%. Por su parte, en el indicador del MRE obtiene una disminución del 52,70%. Por otro lado, el modelo matemático, a pesar de proporcionar la solución óptima exacta, por la naturaleza del problema requiere de largos periodos de ejecución que no son aplicables en la industria. Además, al ejecutarlo por un tiempo limitado y compararlo con el GA, se encuentra que los resultados son mejores para este último. Por lo tanto, para aplicaciones reales se sugiere implementar métodos heurísticos como el propuesto en este proyecto que permitan alcanzar buenas soluciones en un corto tiempo.

La aplicación de este algoritmo genético brindará una mejor programación de la producción dando como resultado a aquellas organizaciones con las características del este proyecto investigativo mejor ocupación de los recursos que posee. Además, impactará positivamente en los costos de consumo de energía de la organización y por consecuente, el impacto ambiental será también positivo.

D. Recomendaciones

- Se sugiere desarrollar una interfaz gráfica que sea amigable con el usuario, permitiéndole ingresar la información de forma simple al sistema.
- Se propone incluir una comunicación directa entre el programa y la base de datos de la empresa, con el fin de automatizar el proceso de programación diaria en una determinada compañía.
- Se sugiere hacer la validación del problema en un contexto en particular. Es decir, hacer la implementación del AG y el modelo en una industria y posteriormente, hacer un análisis sobre los beneficios obtenidos.

- Con el fin de establecer los parámetros adecuados para el GA se propone la implementación de un DOE (*Design Of Experiments*) en el cual se estudie la relación de cada uno de los parámetros con el desempeño del algoritmo.

VI.GLOSARIO

Instancia: Datos de prueba.

Job shop: Es un problema en donde se tiene un conjunto finito de trabajos que requieren que una cierta secuencia de operaciones sea realizada en un grupo de máquinas o recursos específicos; asumiendo que existe una secuencia de procesamiento para cada uno de los trabajos.

Flexible Job shop: es un problema de Job shop que se caracteriza porque cada una de las máquinas pueden o no realizar varios tipos de tareas.

Lateness: tardanza.

Makespan: tiempo que tardan en estar listos todos los trabajos, es decir, cuando el trabajo más tardío finaliza. Vale la pena aclarar que las unidades de medida no se especifican, puesto que este proyecto puede ser aplicable a distintos sectores, por lo tanto, puede haber sectores que manejen tiempos minutos, horas, días, o cualquier otra unidad de tiempo.

VII.REFERENCIAS

- [1] M. L. Pinedo, *The role of scheduling*, New York, NY, USA: Springer, 2016.
- [2] D. W. & P. E. Blazewicz Jacek, «The job shop scheduling problem:Conventional and new solution techniques.,» *European Journal Of Operational Research.*, n° 93., pp. 1 - 33., 1996..
- [3] J. A. Herrera, «Planos en SolidWorks,» PMP Productos mecánicos de calidad (S.A.S), Cali, 2020.
- [4] O. D. C. y. W. A. S. Santiago Ruiz, «A Multiobjective Methodology to Optimize a Job Shop Environment,» *Scielo*, vol. 23, n° 1, pp. 35-46, 2012.
- [5] E. B.V., «Scopus,» [En línea]. Available: <https://bdbib.javerianacali.edu.co:2127/term/analyzer.uri?sid=d93f1f4b3f35dcc39a12ef1eb4e6f54f&origin=resultslis&src=s&s=TITLE-ABS-KEY%28Flexible+AND+%28%28Job+AND+shop%29+OR+jobshop%29%29&sort=plf-f&sdt=b&sot=b&sl=55&count=2527&analyzeResults=Analyze+re.> [Último acceso: 15 Marzo 2020].
- [6] M. d. Trabajo, «mintrabajo,» [En línea]. Available: <http://www.mintrabajo.gov.co/web/portal-de-ninos-ministerio-del-trabajo/que-es-el-ministerio-del-trabajo.> [Último acceso: 22 02 2020].
- [7] M. d. l. TICS, «mintics,» [En línea]. Available: [https://www.mintic.gov.co/portal/inicio/Ministerio/Acerca-del-MinTIC/.](https://www.mintic.gov.co/portal/inicio/Ministerio/Acerca-del-MinTIC/)
- [8] «Wipo.int.,» 2020. [En línea]. Available: <https://www.wipo.int/edocs/lexdocs/laws/es/can/can012es.pdf.> [Último acceso: 26 Febrero 2020].
- [9] Alcaldiabogota.gov.co, «Compilación de la Legislación Aplicable al Distrito Capital :: RÈgimen Legal de Bogotá.,» 2020. [En línea]. Available: <https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=3431&dt=S..> [Último acceso: 26 02 2020].

- [1] Iso25000.com, «I. Calidad et al., "NORMAS ISO 25000",» 2020. [En línea]. Available: 0] <https://iso25000.com/index.php/normas-iso-25000..> [Último acceso: 26 02 2020].
- [1] R. R. A. C. y. S. F. S. A. Oddi, «Applying Iterative Flattening Search to the Job Shop Scheduling Problem with 1] Alternative Resources and Sequence Dependent Setup Times.,» *COPLAS*, pp. 15-22, 2011.
- [1] J. Hurink, B. Jurisch y M. Thole, «Tabu search for the job-shop scheduling problem with multi-purpose machines.,» 2] *OR Spectrum*, vol. 15, n° 4, p. 205–215, 1994.
- [1] G. Calleja y R. Pastor, «A dispatching algorithm for flexible job-shop scheduling with transfer batches: an industrial 3] application,» *Production Planning & Control*, vol. 25, n° 2, pp. 93-109, 2014.
- [1] L. Lin y H. Jia-zhen, «Multi-Objective Flexible Job-Shop Scheduling Problem in Steel Tubes Production,» *Systems 4] Engineering — Theory & Practice*, vol. 29, n° 8, pp. 117-126, 2009.
- [1] K. C. B. Felix T.S. Chan, «A genertic algorithm-based scheduler for multiproduct parellel machine sheet metal job 5] shop,» *ELSEVIER*, vol. 38, n° 7, JULIO 2011.
- [1] S. R. Herrera, «Metodología multiobjetivo basada en un comportamiento evolutivo para programar sistemas de 6] producción flexible job shop. Aplicaciones en la industria metalmecánica.,» Manizales, Colombia, 2015.
- [1] V. Jain y T. Raj, «An adaptive neuro-fuzzy inference system for makespan estimation of flexible manufacturing 7] system assembly shop: a case study,» *International Journal of Systems Assurance Engineering and Management*, vol. 9, n° 6, pp. 1302-1314, 2018.
- [1] M. Ortiz1, D. Neira y G. Jiménez, «Solving Flexible Job-Shop Scheduling Problem with Transfer Batches, Setup 8] Times,» *Springer International Publishing Switzerland*, vol. 9713, n° 1, pp. 47-58, 2016.
- [1] S. M. M. Morshed, «A hybrid genetic tabu search algorithm for solving job shop scheduling problems: a case study,» 9] *Journal of intelligent manufacturing*, vol. 23, n° 4, pp. 1063-1078, 2012.
- [2] J. Z. y. J. Yang, «Applying Iterative Flattening Search to the Job Shop Scheduling Problem with Alternative Resources 0] and Sequence Dependent Setup Times.,» *International Journal of Production Research*, vol. 55, n° 16, pp. 4894-4918, 2016.
- [2] M. A. González, V. C. R y V. Ramiro, «An Efficient Memetic Algorithm for theFlexible Job Shop with Setup Times,» 1] de *Twenty-Third International Conference on Automated Planning and Scheduling*, Roma, Italia, 2013.
- [2] G. Al Aqel, X. Li y Gao, «A Modified Iterated Greedy Algorithm for Flexible Job Shop Scheduling Problem,» 2] *Chinese Journal of Mechanical Engineering*, vol. 32, n° 21, 2019.
- [2] H. Nouria, O. B. Drissa y K. Gh´ediraa, «Hybrid metaheuristics within a holonic multiagent model for the flexible job 3] shop problem,» *Procedia Computer Science*, vol. 60, p. 83 – 92, 2015.
- [2] A. Bagheri y M. Zandieh, «Bi-criteria flexible job-shop scheduling with sequence-dependent setup times—Variable 4] neighborhood search approach,» *Journal of Manufacturing Systems*, vol. 30, pp. 8-15, 2011.
- [2] L. M. J. D. Daniel Morillo, «Metodologías Analíticas y Heurísticas para la Solución del Problema de Programación de 5] Tareas con Recursos Restringidos (RCPSp): una revisión. Parte 1,» *Scielo*, vol. 10, pp. 247-271, 2014.

- [2 F. M. Carlos Andre Reis Pniheiro, «Heuristics in Analytics,» de *Heuristics in Analytics*, New Jersey, Wiley, 2014, p. 6] 26.
- [2 J. A. M. Mario Cesar Vélez, «METAHEURÍSTICOS: UNA ALTERNATIVA PARA LA SOLUCIÓN DE 7] PROBLEMAS COMBINATORIOS EN ADMINISTRACIÓN DE OPERACIONES,» *EIA*, 2007.
- [2 T.-M. Reza, K. M y N. Bashir, «A hybridization of simulated annealing and electromagnetic-like mechanism for job 8] shop problems with machine availability and sequence-dependent setup times to minimize total weighted tardiness,» *Soft Computing*, vol. 13, n° 10, pp. 995-1006, 2009.
- [2 S.-M. Mohammad y P. Fattahi, «Flexible job shop scheduling with tabu search algorithms,» *The International Journal 9] of Advanced Manufacturing Technology*, vol. 32, pp. 563-570, 2007.
- [3 Z. Guohui, H. Yifan, J. Sun y Z. Wenqiang, «An improved genetic algorithm for the flexible job shop scheduling 0] problem with multiple time constraints,» *Swarm and Evolutionary Computation*, vol. 54, p. 100664, 2020.
- [3 A. Ameni, E. Meriem y B. Lamjed, «A self-adaptive hybrid algorithm for solving flexible job-shop problem with 1] sequence dependent setup time,» *Procedia Computer Science*, vol. 112, pp. 457-466, 2017.
- [3 A. Ameni, E. Meriem y B. Lamjed, «A hybrid algorithm for flexible job-shop scheduling problem with setup times,» 2] *International Journal of Production Management and Engineering*, vol. 5, n° 1, pp. 23-30, 2017.
- [3 A. Ameni, C. Abir, E. Meriem y B. Lamjed, «Handling Sequence-dependent Setup Time Flexible Job Shop Problem 3] with Learning and Deterioration Considerations using Evolutionary Bi-level Optimization,» *Applied Artificial Intelligence*, vol. 34, n° 6, pp. 433-455, 2020.
- [3 S. Oleh y M. Lars, «Heuristic approaches for scheduling jobs in large-scale flexible job shops,» *Computers & 4] Operations Research*, vol. 68, pp. 97-109, 2016.
- [3 L. Jun-Qing, P. Quan-Ke y S. P. N, «A hybrid tabu search algorithm with an efficient neighborhood structure for the 5] flexible job shop scheduling problem,» *The International Journal of Advanced Manufacturing Technology volume*, vol. 52, pp. 683-697, 2011.
- [3 Z. Mohsen, «A heuristic algorithm for solving flexible job shop scheduling problem,» *The International Journal of 6] Advanced Manufacturing Technology*, vol. 71, pp. 519-528, 2013.
- [3 X. Li-Ning, C. Ying-Wu, W. Peng, Z. Qing-Song y J. Xiong, «A Knowledge-Based Ant Colony Optimization for 7] Flexible Job Shop Scheduling Problems,» *Applied Soft Computing*, vol. 10, n° 3, pp. 888-896, 2009.
- [3 W. Lei, C. Jingcao, L. Ming y L. Zhihu, «Flexible Job Shop Scheduling Problem Using an Improved Ant Colony 8] Optimization,» *Scientific Programming*, p. 9, 2017.
- [3 F. S. H. y G. J. Lieberman, *Introduction to operations research*, New York: McGraw-Hill, 2014. 9]
- [4 S. S. y S. Deepa, *Introduction to Genetic Algorithms*, India: Springer, 2008. 0]

- [4 L. Z. X. S. Y. W. y. C. Z. Guohui Zhang, «A variable neighborhood search based genetic algorithm for flexible,»
1] *Cluster Computing*, vol. 22, pp. 11561-11572, 2019.
- [4 P. & M. N. & M.-P. J. Hansen, «Variable Neighbourhood Search: Methods and applications,» *4RO*, pp. 319-360,
2] 2008.
- [4 J. Y. P. Michel Gendreau, *Handbook of Metaheuristics*, canada: Springer International Publishing, 2019.
3]
- [4 F. Glover, «Tabu search: Part I,» *ORSA Journal on Computing*, vol. 1, n° 3, pp. 190-206, 1989.
4]
- [4 E.-G. Talbi, *METAHEURISTICS FROM DESIGN TO IMPLEMENTATION*, John Wiley & Sons , 2009.
5]
- [4 C. Bluma, J. Puchingerb, G. R. Raidlc y A. Roli, «Hybrid metaheuristics in combinatorial optimization: A survey,»
6] *Applied Soft Computing*, vol. 11, p. 4135–4151, 2011.
- [4 T. L. Saaty, *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*, Springer
7] Science+Business Media, LLC , 2000.
- [4 A. Figueroa, *Análisis Jerárquico total AHP*, Cali, 2020.
8]
- [4 S. I. B. R. K. R. A. M. H. A. Z. F. N. A. a. U. A. Muhammad Kamal Amjad, «Recent Research Trends in Genetic
9] Algorithm Based Flexible Job,» *Mathematical Problems in Engineering* , vol. 2018, p. 32, 2017.
- [5 H. M. H, *Fundamentals of Artificial Neural Networks*, MIT, 1995.
0]
- [5 N. H. S. J. T. Kazi Shah Nawaz Ripon, «Improved precedence preservation crossover for multi-objective job shop
1] scheduling problem,» *Springerlink*, pp. 119-129, 2009.
- [5 M. Kamal Amjad, S. Ikramullah Butt, R. Kousar, R. Ahmad, M. H. Agha, Z. Faping, N. Anjum y U. Asgher,
2] «Tendencias de investigación recientes en problemas de programación de talleres de trabajo flexibles basados en algoritmos genéticos,» *Hindawi*, vol. 2018, 2018.
- [5 talent.com, «Salario medio para ingeniero industrial en Colombia 2020,» talent.com, 2020. [En línea]. Available:
3] <https://co.talent.com/salary?job=ingeniero+industrial>.
- [5 U. P. d. Catalunya, «Anexo I estudio de implantación de una industria de motores de baja tensión ubicada en
4] Catalunya».
- [5 Valora Analitik, «Precio de energía en bolsa creció más del 300% en enero de 2020 en Colombia,» 03 Feb 2020. [En
5] línea]. Available: [https://www.valoraanalitik.com/2020/02/03/precio-de-energia-en-bolsa-crecio-mas-del-300-en-enero-de-2020-en-colombia/#:~:text=Videos-,Precio%20de%20energ%C3%ADa%20en%20bolsa%20creci%C3%B3m%20del%20300,enero%20de%202020%20en%20Colombia&text=Para%](https://www.valoraanalitik.com/2020/02/03/precio-de-energia-en-bolsa-crecio-mas-del-300-en-enero-de-2020-en-colombia/#:~:text=Videos-,Precio%20de%20energ%C3%ADa%20en%20bolsa%20creci%C3%B3m%20del%20300,enero%20de%202020%20en%20Colombia&text=Para%20).

- [5 CREG, «Tarifario Enero 2020,» 2020.
6]
- [5 X. Gonzales, «En Colombia el factor de emisión de CO2 por generación eléctrica es de 164,38 gramos por kWh,» *La República*, 19 Feb 2020.
7]
- [5 Grupo de estudios empresariales y de competitividad., «Enforque competitivo,» Cámara de comercio de Cali, Cali,
8] Colombia, 2018.
- [5 Ingredion, «Enfoque Competitivo Industrial metalmecánicas,» Cámara de comercio de Cali, Cali, 2018.
9]
- [6 T. Pyzdek y P. Keller, *The Six sigma Handbook*, United State of America: McGraw-Hill, 2010.
0]
- [6 F. García Córdoba, *La tesis y el trabajo de tesis*, México, DF.: Limusa, 2008.
1]
- [6 P. Grech, *Introducción a la Ingeniería, un enfoque a través del diseño*, Colombia: Pearson, 2013.
2]
- [6 IISE - Institute of Industrial and Systems Engineers, «Industrial Engineers Body of Knowledge,» 2018. [En línea].
3] Available: <http://www.iise.org/details.aspx?id=43631>. [Último acceso: 15 Julio 2018].
- [6 M. Melanie, *An Introduction to Genetic Algorithms*, London,England: A Bradford Book The MIT Press, 1999.
4]
- [6 L. Wanhammar, « DSP Algorithms,» de *DSP Integrated Circuits*, Academic Press, 1999, pp. 225-275.
5]
- [6 X.-S. Yang, «Chapter 5 Genetic Algorithms,» de *Nature Inspired Optimizations Algorithms*, ElSelvier, 2014, pp. 77-
6] 87.

VIII.ANEXOS

TABLA XX.
TABLA DE ANEXOS

No. Anexo	Nombre	Desarrollo (propio o terceros)	Tipo de Archivo (PDF, HTLM, Excel, Word...)
1	Funcionamiento de las industrias del sector metalmeccánico	Propio	Word
2	Información de las entrevistas a las empresas metalmeccánicas en el Valle del Cauca.	Propio	Word
3	Diagrama de flujo del proceso de programación manual de trabajos según la regla de despacho ERD con varios modos de realización	Propio	Word
4	Encuesta de evaluación de métodos de solución para el Flexible Job Shop (FJSP) (Respuestas)	Propio	Excel
5	Matriz de selección multicriterio AHP	Propio	Excel
6	Plan de trabajo del proyecto	Propio	MS Project
7	Modelo matemático en AMPL	Propio	AMPL
8	Algoritmo genético	Propio	Visual Studio
9	Manual de usuario	Propio	PDF

ANEXO 1: Funcionamiento de las industrias del sector metalmeccánico en el Valle del Cauca.

Nombre de la empresa: PMC Productos mecánicos de calidad (S.A.S.)

Nombre del entrevistado: Jaime Andrés Herrera.

Fecha: 24/02/2020

Preguntas:

1. ¿Cómo realiza el proceso de programación (Planeación) de los trabajos en el taller?

En el taller el proceso de planeación inicia con una solicitud de cotización de parte del cliente. Cuando se trata de una pieza de la que el cliente ya tiene registro el procedimiento es el siguiente:

- El cliente envía una solicitud de cotización en la que adjunta el plano de la pieza que se va a fabricar.
- Después de recibir la RQ, se observa y analiza cuidadosamente el plano para determinar cuál es la cantidad óptima de material a comprar y el costo. Ejemplo: La empresa X, necesita cotizar la fabricación de 100 platinas en acero y adjunta el siguiente plano:

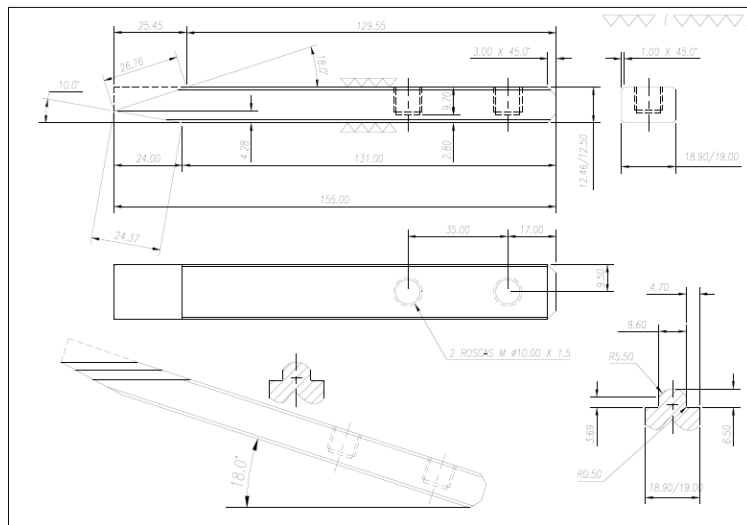


Fig. 12. Gráfico 2D de una platina de acero. Fuente: tomado de [3]

Al momento de observar el plano, la pieza lleva un plano inclinado con un ángulo de 18 grados. Por tal motivo, se debería comprar un material que le permita al operario realizar la operación del plano inclinado, a varias piezas al mismo tiempo, y así reducir el número de montajes realizados en la máquina, debido a que hacer el plano inclinado a cada platina por separado, tomaría más tiempo que hacerlo en varias a la vez. Después, se realiza el plano inclinado al paquete de las platinas, se procedería a separarlas mediante un corte. En la siguiente imagen se observa el plano que recibiría el operario con la línea guía por donde debería cortar las platinas luego de realizar el procedimiento de los planos inclinados.

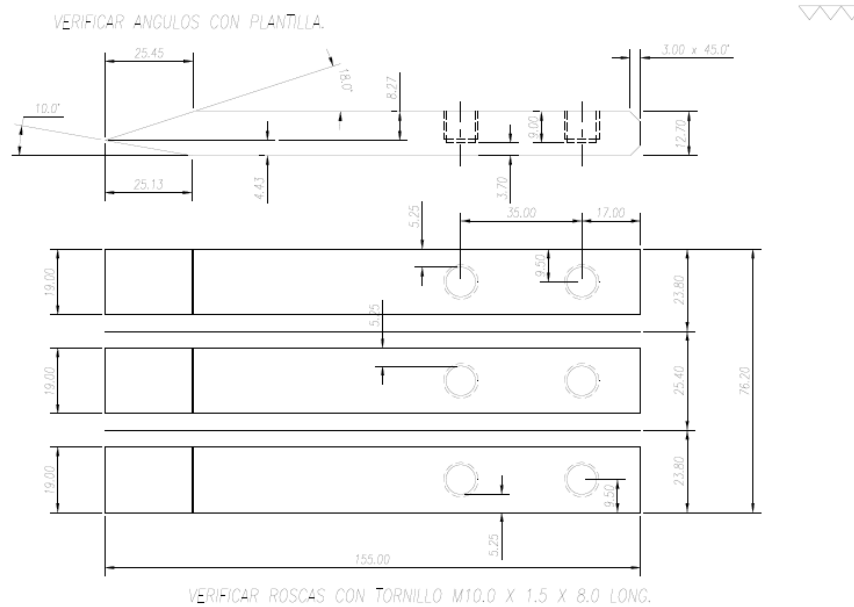


Fig. 13. Plano con líneas guías de la platina de acero. Fuente: tomado de [3]

- Después de evaluar el costo del material y el tiempo en horas máquina que se utilizará para la fabricación de las piezas, se procede a enviar la cotización al cliente
 - Una vez que el cliente recibe la cotización y envía una orden de compra, se procede a comprar el material para que el operario inicie con la fabricación.
2. ¿Qué tipos de problemas se presentan al momento de programar los trabajos?

- Mala interpretación del plano: En ocasiones el operario interpreta mal el plano y comete errores en el proceso de mecanizado.
 - Incorrecta planeación de los recursos: Un ejemplo común, es en el inicio de un trabajo con las herramientas desgastadas, hecho que no permite la realización del trabajo; en consecuencia, pérdida de tiempo esperando las herramientas nuevas.
 - Errores del dibujante: En ocasiones el cliente envía un plano al que le faltan medidas, lo que causa que el proceso de cotización tome más tiempo de lo normal, pues se debe esperar a que el cliente confirme las medidas faltantes.
 - No se cuenta con el sólido (Dibujo 3D). El no contar con el dibujo en 3D puede causar que el operario interprete mal el plano con el que está trabajando. Por otro lado, el sólido también es necesario cuando se va a utilizar una máquina de CNC que debe ser programada a través de un computador.
3. ¿Cuál es el aspecto (Costos, eficiencia, tiempos) más importante para desarrollar en el sector de la fabricación de piezas?

El aspecto más importante en la fabricación de piezas en la industria de la metalmecánica son la eficiencia y el tiempo. Se busca siempre tener una producción rápida y eficiente para disminuir las horas de máquina empleadas en cada trabajo y cumplir con los tiempos de entrega pactados con el cliente.

ANEXO 2: Información de las entrevistas a las empresas metalmecánicas en el Valle del Cauca.

TABLA XXI.
 REPORTE DE ENTREVISTA A EMPRESAS DEL SECTOR METALMECÁNICO.

Empresa	Entrevistado	Cargo	Respuesta a la entrevista
Metalmeccanica mecanizados S.A.S	Edinson	N/A	Bajo el criterio del cliente, si un cliente les genera mayores ingresos, le dan prioridad a ese pedido. Todas las máquinas tienen su propia función. Los productos ya manejan unos trabajos específicos, sin embargo, si una orden es pequeña y se puede sacar rápido, se asumen los tiempos de alistamiento para modificar las máquinas y sacar la orden pequeña. Quien confirme la cotización primero, a ese se le da prioridad.
Cubiertas metálicas	Marco Tulio	N/A	Depende del compromiso/requerimiento adquirido contractual. Cuentan con máquinas multifuncionales. Tratan de manejar un orden con los pedidos, con el fin de facturar. Priorizan la orden que llegue primero, bajo compromisos o requerimientos en la negociación.
Taller de mecanizados de aluminio	Cesar Sanchez	N/A	Depende de la contratación. Se basan en las fechas de entrega que se estipulan en la contratación.
Metalmeccánica de Occidente	Andres Valencia	Analista financiero	Atienden por orden de entrada, es decir que se priorizan las primeros que se confirmen. Cuando llega la orden. Cada máquina hace una sola función. Hay excepciones de ordenes urgentes.
Metalmeccánica JAN	Lina Marquez	Supervisora de producción	Depende de la dificultad del trabajo, se priorizan los trabajos por tiempos, de manera que se puedan sacar rápido los trabajos cortos y fáciles y darles más plazo y tiempos a los trabajos más largos y difíciles. Manejan máquinas multifuncionales. También, tienen en cuenta los pedidos emergencia, que priorizan dependiendo de si hay disponibilidad y se pueden sacar rápido.
Metalmeccánica MEISA	Jhon Harold Melendez	Coordinador de proyectos	Se priorizan por contratos, dependiendo de las fechas de entrega que se estipulen en estas. Máquinas con una sola función.
AFH Montajes y Mantenimientos Metalmeccánicos	Andres Felipe	Gerente	Priorizan por el orden en que llega la toma de servicio. Tienen máquinas multifuncionales.
Metalmeccánica LUCENA	Hernan Mejía	Jefe de producción y compras	Se prioriza por la fecha de entrega. Si manejan máquinas multifuncionales
Ingemental S.A.S	Juan David Carvajal	Director de proyectos	Se calculan los tiempos de producción y se priorizan por la fecha de entrega. Se recolectan todas las órdenes y se priorizan por medio de discusión de ingenieros, teniendo en cuenta los tiempos de producción y las fechas de entregas. Manejan máquinas multifuncionales.
Maquinados Industriales Huber Valencia	Pilar Valencia	Ingeniera de producción	Dependen del orden de llegada, excepto por pedidos de urgencia. Manejan máquinas multifacéticas.

ANEXO 3: Diagrama de flujo del proceso de programación manual de trabajos según la regla de despacho ERD con varios modos de realización

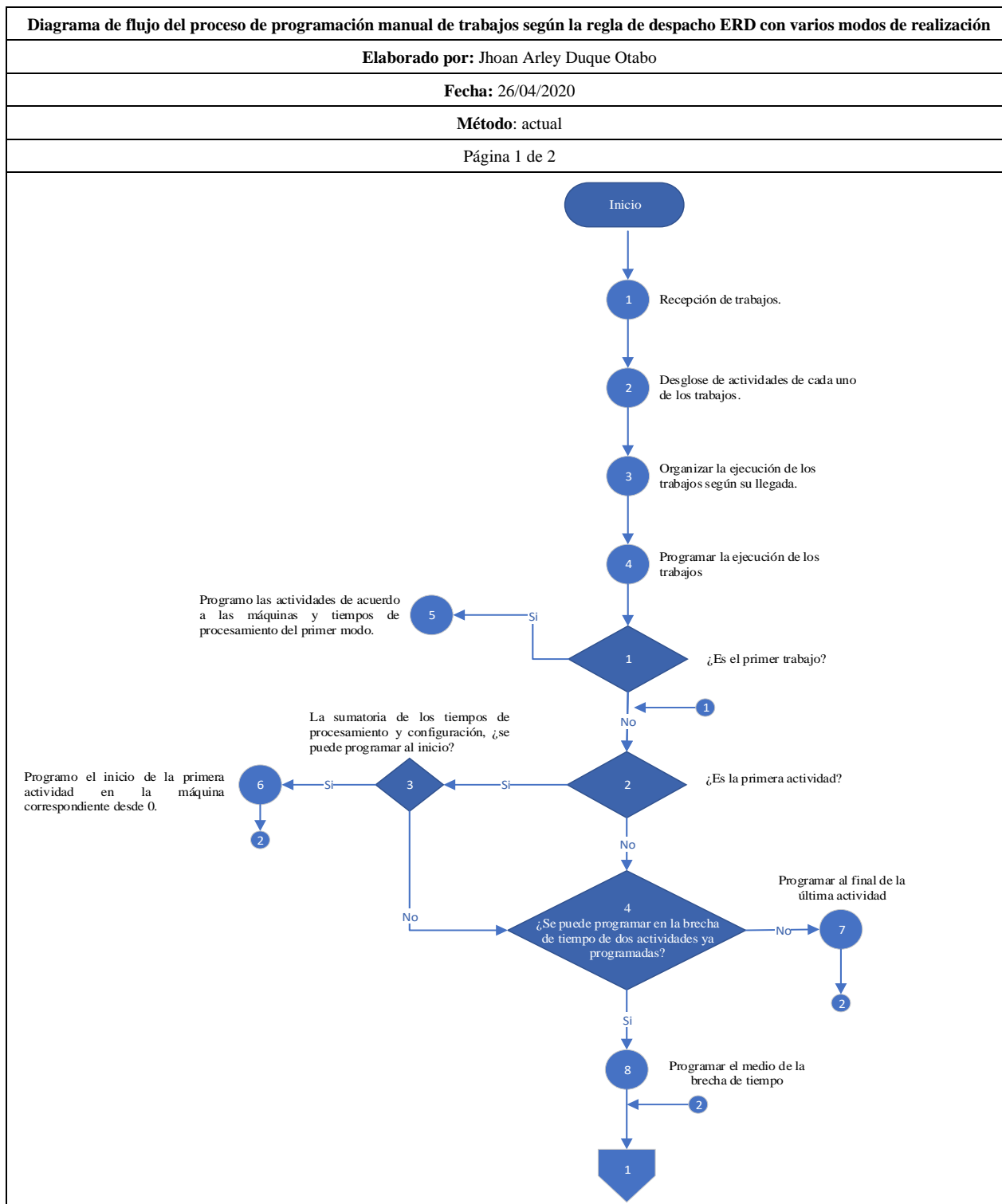


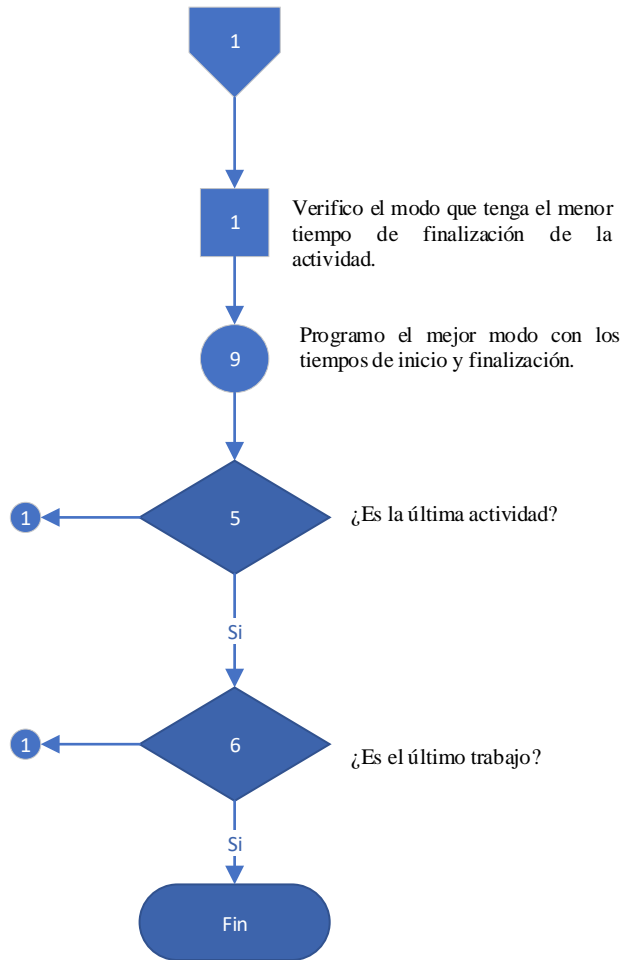
Diagrama de flujo del proceso de programación manual de trabajos según la regla de despacho ERD con varios modos de realización

Elaborado por: Jhoan Arley Duque Otabo

Fecha: 26/04/2020

Método: actual

Página 2 de 2



Resumen de eventos:

Operaciones: 9

Inspecciones: 1

Decisiones: 6

Total: 16

Comentarios: Este diagrama de flujo es un ejemplo generalizado tomando en cuenta las instancias (SDST-HUData).

Cuello de Botella: Para el caso de un Flexible Job Shop con tiempos de configuración dependientes, la identificación de un cuello de botella es muy variable y ambigua, ya que el tiempo de la actividad puede aumentar, dependiendo de su ubicación. Sin embargo, se puede realizar un análisis de ciertas actividades con respecto a la disminución del Makespan. Un ejemplo es la actividad 3 del trabajo 4, y la actividad 2 del trabajo 3, debido a que, si se puede disminuir el tiempo de procesamiento de estas actividades, se podría aprovechar mejor la brecha de tiempo generada en la máquina 4, entre el trabajo 5 y 4. De igual forma se puede identificar en la actividad 5 del trabajo 8, ya que si se disminuye su respectivo tiempo, se podría aprovechar mejor la brecha de tiempo generada en la máquina 2 entre el trabajo 8 y 10.

La decisión de invertir recursos en tratar de disminuir los tiempos de procesamiento de cada una de las actividades puede ser demasiado costosa, debido a que si no se realiza esta disminución a gran parte de las actividades, sino a unas pocas, su efecto no podría verse reflejado, puesto que la naturaleza del FJS, hace variar la secuencia de trabajos en cada interacción, y probablemente en esa nueva configuración, la disminución del tiempo de procesamiento realizado a ciertas actividades es irrelevante por la naturaleza de la nueva secuencia de trabajos. Por ello, es mucho más eficiente realizar un estudio enfocado a buscar la ejecución óptima o más cercana a este, de los trabajos en cada una de las máquinas, como es el caso de estudio del problema del FJS.

Oportunidad de mejora: La oportunidad de mejora puede identificarse en buscar disminuir el tiempo de procesamiento, para aquellas actividades que generen grandes brechas de tiempo, sin embargo, por lo mencionado anteriormente, puede ser ineficiente, debido a la naturaleza cambiante de los trabajos. Por ello, la oportunidad de mejora radica en la operación 3 del diagrama de flujo, ya que, si se busca un tipo de programación mucho más eficiente que el ERD, se puede reducir el Makespan significativamente.

Puntos Críticos: Los puntos críticos del diagrama de flujo, son: la operación 3 y la inspección 1, puesto que su correcta ejecución, permite reducir el Makespan.

Ventaja competitiva: La ventaja competitiva está enmarcada a todos los procesos enfocados a la reducción del Makespan, en este caso, puede ser la disminución de los tiempos de procesamiento o los tiempos de configuración dependientes. O un método mucho más eficiente de programación, a través de un método exacto, una regla de despacho, o un algoritmo heurístico o metaheurístico.