

MODELO DE PROGRAMACIÓN DE PRODUCCIÓN EN LA LÍNEA TEXTIL DE LA EMPRESA KROMIA S.A.S

Julieth Andrea Espitia Méndez

Germán Leonardo Mendoza Rojas

Escuela Colombiana de Ingeniería Julio Garavito

Decanatura de Ingeniería Industrial

Maestría en Ingeniería Industrial

Bogotá D.C., Colombia

2020

MODELO DE PROGRAMACIÓN DE PRODUCCIÓN EN LA LÍNEA TEXTIL DE LA EMPRESA KROMIA S.A.S

Julieth Andrea Espitia Méndez

Germán Leonardo Mendoza Rojas

Trabajo de grado para optar al título de
Magíster en Ingeniería Industrial

Director

Carlos Rodrigo Ruiz Cruz

Escuela Colombiana de Ingeniería Julio Garavito

Decanatura de Ingeniería Industrial

Maestría en Ingeniería Industrial

Bogotá D.C., Colombia

2020

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

RECONOCIMIENTOS O AGRADECIMIENTOS

Agradecemos a nuestro tutor de tesis, profesor Carlos Rodrigo Ruiz y a la Directora de la maestría Sonia Jaimes por su dedicación, seguimiento y contribución al desarrollo exitoso de este proyecto, así como a los profesores de la maestría de ingeniería industrial de la Escuela Colombiana de Ingeniería Julio Garavito y al profesor Edgar Leonardo Duarte de la Universidad Libre de Colombia por sus asesorías.

De igual forma, un agradecimiento a Luis Carlos Villamil por el apoyo técnico en el desarrollo del software requerido para la implementación y pruebas del algoritmo.

RESUMEN

En la empresa estudiada, perteneciente a la industria textil, se evidencia una configuración productiva tipo flow shop híbrido flexible, en la cual se presenta una serie de características que excluye la programación de producción de los modelos estándares. Debido a la naturaleza NP-Hard del problema presentado en la empresa Kromia S.A.S, ha sido necesario el uso de la metodología de programación basada en algoritmos genéticos, como herramienta de scheduling dirigida hacia la minimización del makespan y cantidad de trabajos tardíos; este modelo es desarrollado teniendo en cuenta las complejidades de la industria, entre las cuales se encuentra la capacidad de máquinas paralelas, tiempos de entrega, flujo de proceso lineal y stage skipping. La metodología desarrollada genera buenos resultados comparados con escenarios reales de la empresa.

ABSTRACT

In the studied company, which belongs to the textile industry, a flexible hybrid flow shop type production configuration is evident, in which is presented a series of characteristics that excludes the production programming of the standard models. Due to NP-Hard nature of the problem shown in the company Kromia S.A.S, it has been necessary to use the programming methodology based on genetic algorithms, as a scheduling tool, aimed at minimizing makespan and the amount of late jobs; this model is developed taking into account the complexities of the industry, among which are the capacity of parallel machines, delivery times, linear process flow and stage skipping. The methodology developed generates good results compared to real company scenarios.

TABLA DE CONTENIDO

1.	INTRODUCCIÓN	11
2.	PROBLEMÁTICA	12
3.	OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN.....	21
4.	MARCO TEÓRICO.....	23
4.1.	Flow Shop	23
4.1.1.	Flow shop híbrido	25
4.1.2.	Flow shop híbrido flexible	26
4.2.	Scheduling – Programación de producción	28
4.2.1.	Heurísticas	29
4.2.2.	Metaheurísticas	31
4.3	Algoritmos Genéticos	34
4.3.1	Algoritmo Genético Multiobjetivo.....	45
5.	METODOLOGÍA PROPUESTA BASADA EN AG	48
5.1.	Supuestos	48
5.2	Conjuntos, parámetros variables y función objetivo	49
5.3	Diagrama de flujo	50
5.4	Metodología propuesta.....	53

5.5	Pseudocódigo de la metodología propuesta	63
6.	EXPERIMENTACIÓN Y RESULTADOS	64
6.1	Análisis datos de entrada	65
6.2	Parámetros del modelo	67
6.3	Pruebas reales	69
7	CONCLUSIONES Y RECOMENDACIONES	79
7.1	CONCLUSIONES.....	79
7.2	RECOMENDACIONES	80
	REFERENCIAS	82

LISTADO DE TABLAS

Tabla 1. Archivo de entrada	65
Tabla 2. Cantidad de metros procesados vs. makespan real	66
Tabla 3. Paso de cada tipo de producto por máquinas	67
Tabla 4. Tiempos de procesamiento por etapa	68
Tabla 5. Parámetros del sistema	69
Tabla 6. Cantidad de metros procesados vs. makespan pruebas	73
Tabla 7. Resultados de las simulaciones	76
Tabla 8. Comparación mejores soluciones vs. datos reales	76

LISTADO DE FIGURAS

Figura 1. Diagrama de bloques Kromia S.A.S	15
Figura 2. Planta de producción Kromia S.A.S.....	16
Figura 3. Recorrido de producto en planta.....	17
Figura 4. Diagrama multiproducto de la fase de acabados	20
Figura 5. Representación gráfica de un sistema flow shop híbrido flexible.....	27
Figura 6. Representación de la población en un algoritmo genético.....	37
Figura 7. Representación cruce de un punto	42
Figura 8. Cruce de dos puntos.....	43
Figura 9. Cruce uniforme	44
Figura 10. Operador de cruzamiento PMX.....	44
Figura 11. Metodología basada en el algoritmo genético.....	53
Figura 12. Transformación de artículos a procesar.....	55
Figura 13. Tabla asignación de artículos a máquinas y turnos	55
Figura 14. Ejemplo de asignación de artículos a máquinas y turnos	57
Figura 15. Representación de una subpoblación.....	57
Figura 16. Representación de un cromosoma	58
Figura 17. Ejemplo de generación de subpoblaciones y cromosomas	58
Figura 18. Medición de fitness a cada individuo	59
Figura 19. Selección de mejores padres según makespan.....	59
Figura 20. Pseudocódigo de asignación de máquinas para proceso tinturado	63
Figura 21. Pseudocódigo del algoritmo genético	64
Figura 22. Gráfico análisis de correlación makespan real.....	66

Figura 23. Paso a paso funcionamiento del sistema.....	70
Figura 24. Interfaz gráfica de usuario	70
Figura 25. Interfaz gráfica configuración mejor cromosoma	71
Figura 26. Gráfico análisis de correlación	73
Figura 27. Comparación makespan pruebas vs. makespan real	76
Figura 28. Tiempo de ejecución vs configuraciones del sistema	77
Figura 29. Diagrama de Gantt.....	78

1. INTRODUCCIÓN

El sector textil en los últimos 10 años se ha caracterizado por pertenecer a uno de los rubros superiores frente a la economía nacional, por lo cual actualmente se está enfrentando a economías emergentes como la china y a mercados como el americano con costos más competentes. Actualmente en Colombia se encuentran activas 518 empresas caracterizadas dentro del sector textil, de las cuales, 117 de ellas corresponden al tipo de empresas de fabricación de productos textiles técnicos, adicionalmente de la totalidad de empresas textiles, 311 (17% de la totalidad) se encuentran caracterizadas como medianas con base en ingresos operacionales (Martín, 2016). Dada la complejidad de procesamiento de los productos desarrollados en la industria de textiles técnicos, la programación de producción no cuenta con métodos o teorías que tengan en cuenta la mayoría de variables que impactan la productividad de este sector; esto se debe a que, según la investigación realizada, la mayor parte de empresas tienen una configuración productiva tipo flow shop en sus distintas clasificaciones, de igual forma, los diferentes autores apuntan a los algoritmos genéticos como principal herramienta, ya que es una técnica robusta que encuentra soluciones aceptables en tiempos razonables y además se adapta a las particularidades de cada problema.

2. PROBLEMÁTICA

KROMIA S.A.S es una compañía especializada en textiles y productos técnicos en el área de ortopedia y deportiva, a través de las marcas propias “TECNOMED” se desarrolla la línea blanda ortopédica, y para este caso, la marca “KRIK” genera el producto “Cierre de contacto”.

La empresa KROMIA S.A.S actualmente maneja 3 líneas de producto, las cuales son:

- Textil: fabricación de KRIK
- Laminado: fabricación de telas impermeables
- Dispositivos médicos: productos ortopédicos

La línea de productos de “Laminado” es una línea recientemente implementada en la empresa, y no cuenta con una producción continua, sino por demanda; la línea de “Dispositivos médicos” actualmente es desarrollada con procesos de manufactura de personal, distribuidos en los diferentes puntos de producción o “talleres élites”, los cuales corresponden a plantas de producción tercerizadas de acuerdo a la demanda de los productos de este catálogo; la línea de productos “Textil” es la encargada de la producción del cierre de contacto, registrado como “KRIK”, esta línea de productos cuenta con una planta de producción independiente, la cual funciona con un personal de planta correspondiente a 14 personas, quienes operan en el manejo de las siguientes máquinas:

- Tejeduría (12)
- Urdido (1)
- Retorcido (1)
- Teñido (3)
- Mageba (1)
- Enrollado (1)
- Empaque (1)

- Cepillo Breiten (1)
- Acabado Termofijadora (2)
- Corte de Monofilamento (1)
- Corte de Ultrasonido (1)

La línea de producción compuesta por esta maquinaria genera 4 tipos de productos:

- Krick Gancho
- Krick Felpa
- Reatas*
- Elástico Plano*

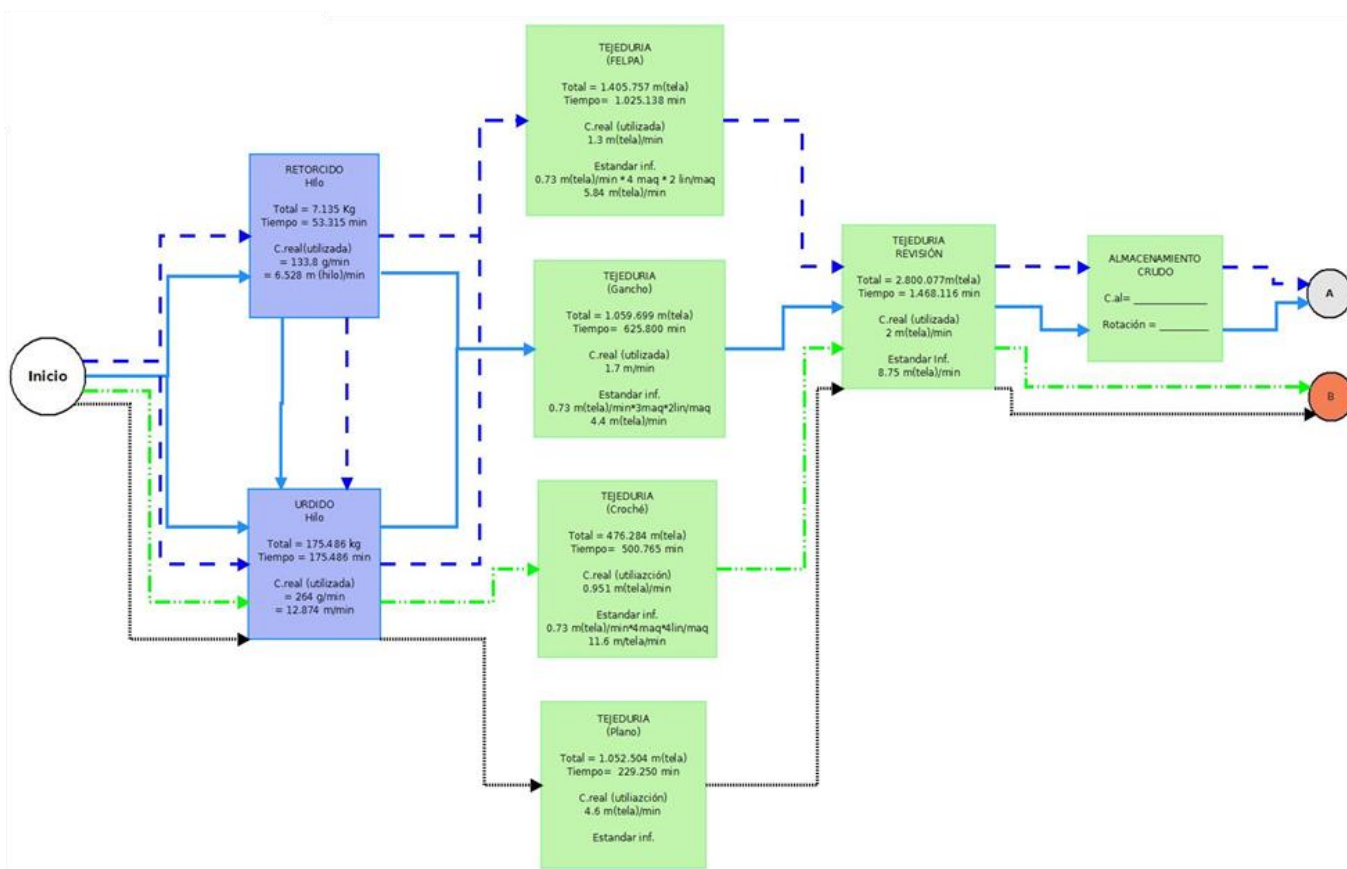
* Los productos “Reatas” y “Elástico Plano” no son directamente comercializados, estos, son utilizados como suministro de materia prima para la producción de la línea de ortopédicos.

A continuación, en la figura 1 se presentará el recorrido correspondiente a los productos y maquinaria de la línea textil:

LÍNEAS DE PRODUCTOS



ÁREAS DE PRODUCCIÓN



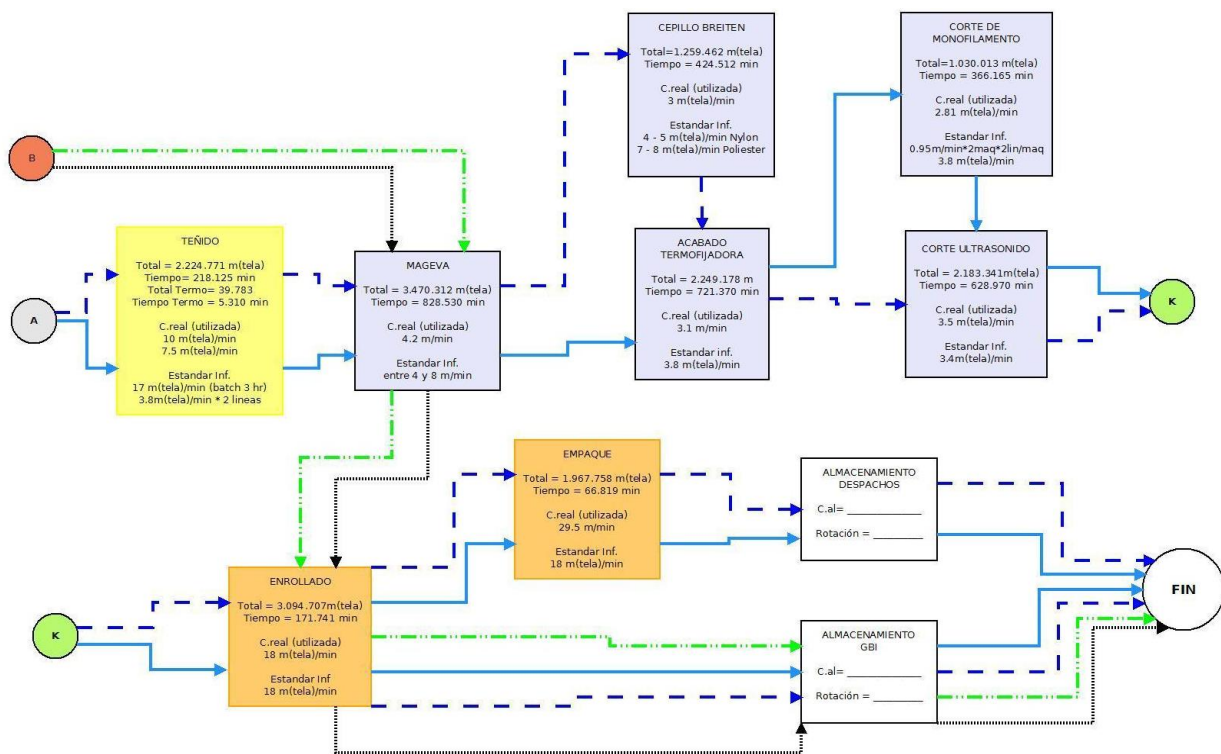


Figura 1. Diagrama de bloques Kromia S.A.S

Fuente: Kromia S.A.S

En la figura 2 se presenta el diseño de planta que indica el proceso de producción:



Figura 2. Planta de producción Kromia S.A.S

Fuente: Kromia S.A.S

1. Retorcido y Urdido
2. Tejeduría
3. Teñido
4. Acabados
5. Enrollado y Empaque

A continuación, en la figura 3 se presenta el recorrido del producto en la planta textil de Kromia S.A.S:

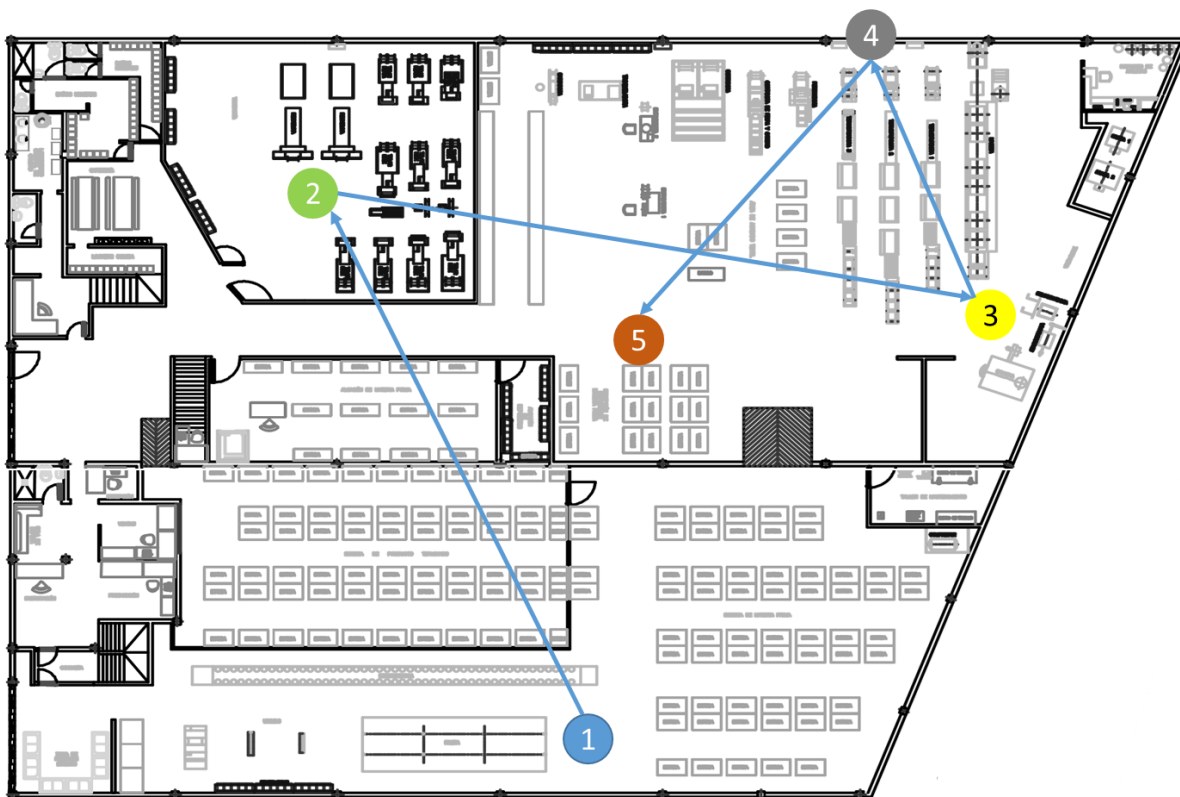


Figura 3. Recorrido de producto en planta

Fuente: Elaboración propia

El 95,4% de los productos fabricados en la línea textil, corresponden a productos de Krik (Felpa y Gancho), los cuales se clasifican en 2 tipos de producto:

- **Productos con terminación total:** Corresponde a los Krik de felpa y gancho de color blanco y negro, procesados en un solo lote de fabricación generado a partir de la definición de stock mínimo; para estos productos se maneja el modelo de producción en línea.
- **Productos con terminación parcial:** Corresponde a los Krik de felpa y gancho, con terminaciones de otros colores (diferentes a blanco y negro), los cuales son procesados en 2 etapas; la primera, construye la tela en un color “crudo” bajo los lineamientos de los productos con terminación total, una vez

producidos, son almacenados para llevar a cabo la segunda etapa, la cual corresponde a la adaptación de los colores requeridos por “producción bajo pedido”.

Para la construcción del modelo objetivo, se selecciona la fase de acabados de la línea textil dados los siguientes argumentos:

1. Los productos generados en la línea textil son considerados como el insumo principal en la línea de dispositivos médicos.
2. En esta planta en el área de acabados, se evidencia una configuración de producción Flow Shop Híbrido Flexible, debido a que cuenta con máquinas paralelas en alguna de sus fases de producción y existen trabajos que omiten al menos una de las etapas del proceso.

La etapa de acabados consta de los siguientes procesos:

a. Teñido: Aplicación de tintes o pigmentos en materiales textiles como fibras, hilos y telas con el objetivo de lograr un color con la solidez deseada a través de la acción física o química (Testex, 2019) Existen dos formas de tinturar una fibra: por afinidad entre colorante y fibra y por pigmentación de la fibra. Kromia utiliza la forma de afinidad también denominada por agotamiento, en este proceso son las fuerzas de afinidad entre colorante y fibra lo que hace que el colorante pase del baño a la fibra hasta saturarla y quedar fijada en él; para este tipo de teñido se utilizan tres tipos de máquinas que se diferencian por su acción mecánica que actúa sobre la materia textil, sobre el baño tintóreo o sobre ambos elementos a la vez.

Tipo I: Máquinas con la fibra a teñir estática y la solución de colorante en movimiento, entre estas máquinas se encuentran las autoclaves.

Tipo II: Máquinas con el sustrato en movimiento y la solución fija (teñido en cuerda o torniquete, teñido en Jigger).

Tipo III: Máquinas en las que el sustrato y la solución están en movimiento durante el proceso (teñido en jet, teñido de madejas, teñido en máquina mezzera) (Mejía Azcárate, 2015)

Kromia para el proceso de tintorería cuenta con tres autoclaves cada una con diferente capacidad; los costos de montaje de la maquinaria son relativos a la capacidad de cada una de ellas, por lo tanto, se tiene como política empresarial para la minimización de costos, la restricción de capacidades asignadas a cada máquina.

- b. Mageba:** Proceso en el que se agregan aditivos o suavizantes a las fibras.
- c. Cepillado breitenbach:** Proceso que se realiza con el fin de darle cuerpo a las telas o fibras, se realiza en una o varias máquinas con múltiples rodillos revestidos con púas metálicas que giran y contragiran continuamente tirando las fibras de la superficie del tejido, para dar una textura afelpada, mientras las telas ganan volumen y adherencia (Mejía Azcárate, 2015)
- d. Termofijado:** El objetivo de esta operación es proporcionar estabilidad dimensional a la tela; que depende de la temperatura del tratamiento, del tiempo de este y de las condiciones de tensión o distensión que se aplican. Este proceso también modifica la resistencia de los tejidos a las arrugas (Naik & Gacén, 1985)
- e. Corte monofilamento:** Proceso en el que se realiza un corte a los filamentos de la fibra del producto tipo gancho para que se obtenga la adherencia adecuada.
- f. Corte ultrasonido:** Operación en la que se sellan los bordes de corte evitando que los mismos se desenheben.
- g. Enrollado:** Proceso en el que las cintas se enrollan para ser entregadas al cliente según la unidad de medida de venta.
- h. Empaque:** Operación en el que se empacan las cintas.

A continuación, se presenta el diagrama multiproducto de la fase de acabados de la planta textil para los productos krik felpa y krik gancho.

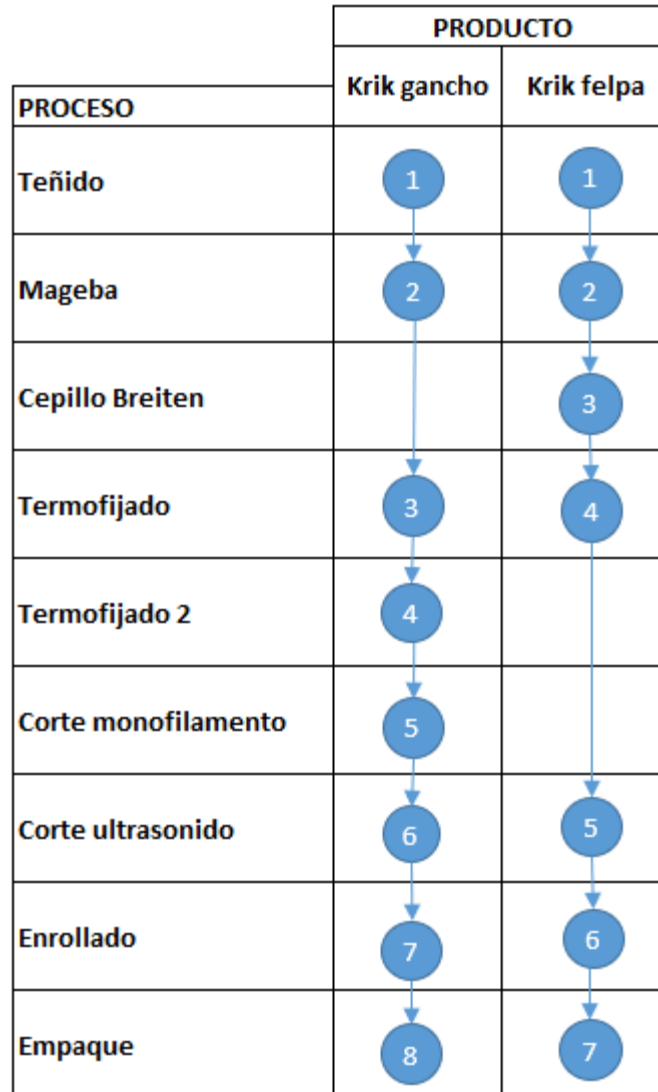


Figura 4. Diagrama multiproducto de la fase de acabados

Fuente: Elaboración propia

Actualmente la empresa Kromia no cuenta con un proceso de la gestión de la producción mediante el cual se establezca la estrategia de esta, por ende, no se determinan unos criterios de competitividad en los cuales se centren los esfuerzos del departamento de producción dirigidos a la necesidad de dar cumplimiento a los tiempos de entrega prometidos al cliente final.

Dada la ausencia de la estrategia de producción, la empresa actualmente no cuenta con las herramientas tácticas y operativas que permitan:

1. Tener una visión de la capacidad de la planta.
2. Estimar los recursos y asignaciones requeridas para el cumplimiento de la demanda en los tiempos pactados.
3. Una programación de producción que minimice los costos y tiempos del proceso.

Teniendo en cuenta el estado de la empresa, se evidencia que es necesario el desarrollo de herramientas dirigidas a la programación de la producción que permita dar cumplimiento a los presupuestos de producción, y niveles de servicio con los clientes internos y externos.

3. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

Objetivo General:

Desarrollar e implementar una metodología de programación de producción basada en el uso de algoritmos genéticos como método de solución para minimizar el makespan y los trabajos tardíos de la línea textil de Kromia SAS.

Objetivos específicos:

- Analizar y caracterizar el sistema productivo estableciendo los factores que afectan la programación de producción de la línea seleccionada.
- Construir un modelo basado en la lógica de los algoritmos genéticos, que se adapte al sistema de producción de la línea textil de Kromia S.A.S.
- Desarrollar el modelo diseñado para minimizar el makespan y número de trabajos tardíos del sistema productivo de Kromia S.A.S.
- Implementar y evaluar el desempeño del modelo propuesto frente a la situación actual de Kromia S.A.S

Pregunta de investigación

¿Cómo el desarrollo de una metodología basada en algoritmos genéticos puede optimizar la programación de producción de la empresa objeto de estudio, que cuenta con una configuración de producción flow shop híbrido flexible?

Alcance

El diseño de la metodología contempla la etapa de acabados compuesta por los procesos de tintorería, mageba, cepillo, acabado, termofijado, corte monofilamento, corte ultrasonido, enrollado y empaque; ya que esta etapa tiene un sistema de manufactura make to order.

Metodología

Con el objetivo de llevar a cabo el desarrollo de este proyecto, se ejecutarán las actividades desglosadas a continuación:

1. Para analizar el contexto dirigido a la metodología basada en algoritmos genéticos a aplicar, en el modelo multiobjetivo de flow shop híbrido flexible, se realizará una búsqueda de información en las diferentes bases de datos especializadas, con el fin de establecer un punto de referencia para la caracterización del sistema productivo de Kromia S.A.S.

2. Revisar de forma detallada el estado actual de la línea de producción textil de la empresa, identificando los factores claves para el diseño del modelo.
3. Estudiar la información recopilada con el objetivo de determinar las diferentes variables que intervienen en el sistema de producción, seleccionando las de mayor impacto en la productividad del sistema.
4. Desarrollar la metodología de programación basada en algoritmos genéticos e implementarla en la línea de producción de Kromia SAS.
5. Generar y ejecutar escenarios de pruebas con el objetivo de comparar los resultados de la metodología planteada con información de tiempos de programación reales de la línea de producción.
6. Conclusiones y recomendaciones.

4. MARCO TEÓRICO

A continuación, se exponen los conceptos teóricos con el tema objeto de estudio del presente trabajo, el cual está enfocado hacia el flow shop híbrido flexible, y desarrollo de algoritmos genéticos:

4.1. Flow Shop

El flow shop clásico es un tipo de configuración productiva en la que un conjunto de trabajos fluye a través de múltiples etapas en el mismo orden (Ribas, Leisten, & Framiñan, 2010); el objetivo de este tipo de configuración es encontrar la mejor secuencia de programación para el procesamiento de cada uno de los trabajos en las máquinas y de esta manera optimizar un criterio específico como makespan, tiempo de flujo total, tardanza total, o número de trabajos tardíos, etc. (Ibrahim, El-shaer, Al-rwasheda, & Nawarra, 2013)

Los productos a fabricar en un flow shop requieren una secuencia similar de operaciones (existen ocasiones que algunos de los productos pueden omitir una o varias operaciones), razón por la que las máquinas se disponen en línea y cuando termina de producirse un lote de cierto producto, se ajustan las máquinas para realizar un producto diferente (J.A. Domínguez, García, M.A. Domínguez, Ruíz & Álvarez, 1995 citado por (González, 2013)

El problema de flow shop es un problema de programación en el que n trabajos tienen que ser procesados por m máquinas, este problema se denomina NP-Hard cuando $m > 3$ (Gómez, Mendes, Riveros, & Resende, 2012) Cada trabajo j ($j = 1, 2, \dots, n$) pasa a través de las m máquinas en un orden definido, requiere un tiempo de procesamiento $p(i, j)$ en cada máquina i ($i = 1, 2, \dots, m$) y tiene una fecha de entrega $r(j)$ (Ren, Diao, & Luo, 2012). En el tipo de configuración flow shop tradicional, los trabajos deben ser procesados en cada una de las etapas, lo que significa que no se permite omitir fases en el proceso productivo (Gomez, Romano, & Cruz, 2012)

Debido a la cantidad de variables que existen en los ambientes de producción reales; (Allahverdi, Ng, Chen, & Kovalyov, 2008) clasifican los problemas flow shop de la siguiente manera:

- a. **Flow shop simple:** Existe sólo una máquina en cada etapa.
- b. **Flow shop sin espera:** Una operación inicia inmediatamente después de que termina la anterior.
- c. **Flow shop de montaje:** cada trabajo consta de $m-1$ operaciones, donde cada una es realizada en una máquina previamente determinada en la primera etapa, y posteriormente se realiza un montaje en la máquina de la segunda etapa.
- d. **Flow shop híbrido:** Existe más de una máquina en al menos una etapa del proceso productivo.

Por su parte (De Micheli, 2009) afirma que un sistema flow shop puro no es suficiente para simular la mayoría de las líneas de producción y por consiguiente introduce dos variantes del sistema:

- a. **Stage skipping:** Esta variante trata la posibilidad de que algunos trabajos puedan saltar algunas etapas de la línea de producción, lo que la convierte en una línea flexible. Un problema de flow shop con posibilidad de stage skipping es conocido como flow shop flexible.
- b. **Flow shop con tiempos de preparación dependientes de la secuencia:** Esta forma del flow shop tiene en cuenta los tiempos necesarios de preparación de máquinas para ejecutar el siguiente trabajo.

(Hekmatfar, Fatemi, & Karimi, 2011) agregan a las clasificaciones del flow shop, el flow shop reentrante, aquel en el que los trabajos pueden pasar varias veces por una misma etapa del proceso.

Otra clasificación del flow shop es la denominada flow shop de permutación, esta configuración aplica el criterio de no permitir que un trabajo pase a otro (job passing), es decir, que la secuencia inicial definida se mantenga a lo largo del proceso; una ventaja de este tipo de flow shop es que reduce el espacio de búsqueda (Quan-Ke & Ruiz, 2014).

4.1.1. Flow shop híbrido

El sistema flow shop híbrido es un sistema que ha recibido mucha atención de diferentes investigadores debido a su importancia desde los puntos de vista teóricos y prácticos. Este problema ha demostrado ser el modelo adecuado para el estudio de una gran cantidad de sistemas de producción como el de vidrio, papel, metalurgia, madera, textil y aeroespacial (Ranjan & Mahapatra, 2011) esta clasificación surge de las necesidades de las industrias de incluir más máquinas a una o varias de las etapas de producción con el fin de aumentar o balancear la capacidad del sistema, eliminar cuellos de botella, aumentar la productividad y la flexibilidad, o bien para generar nuevos productos que necesiten de máquinas dedicadas (Jungwattanakit, Reodecha, Chaovalitwongse, & Werner, 2008)

El entorno flow shop híbrido (HFS) es uno de los más comunes en la industria y puede verse como una generalización de dos tipos de problema de programación; el problema de programación de máquina paralela (PMS) y el problema de programación del flow shop (FSS). La decisión clave del problema PMS es la asignación de trabajos a las máquinas y la del FSS es la secuencia de trabajos a través del flujo productivo, por lo tanto, una vez que la configuración del HFS ha sido diseñada, la principal decisión en el funcionamiento de esta es asignar y programar los trabajos a las máquinas en cada una de las etapas, es decir, determinar el orden en el que se realizarán los trabajos (Ribas et al., 2010).

Las características de este tipo de configuración según (Ruiz & Vázquez, 2010) son:

- a. El número de etapas de procesamiento es al menos dos.
- b. En al menos una etapa pueden existir dos o más máquinas en paralelo, que se pueden clasificar como: máquinas paralelas idénticas (los tiempos de procesamiento son los mismos para cada máquina), máquinas paralelas uniformes (las máquinas tienen una relación paramétrica en términos de tiempo de procesamiento) y máquinas paralelas no relacionadas (si las diferencias entre los tiempos de procesamiento de las máquinas no pueden ser expresadas en una relación paramétrica). (Cevikcan, Durmusoglu & Baskak, 2011 citado por (López & Arango , 2015).
- c. Todos los trabajos son procesados siguiendo el mismo flujo de producción.
- d. Cada trabajo requiere un tiempo de procesamiento en cada etapa.
- e. Todos los trabajos y máquinas están disponibles en el instante cero.
- f. Cualquier máquina puede procesar solo un trabajo a la vez y cualquier trabajo puede ser procesado por una sola máquina a la vez.
- g. Los tiempos de preparación no tienen una magnitud relevante.
- h. No se permite la preferencia entre trabajos.
- i. Los datos del problema son determinísticos y conocidos de antemano.

4.1.2. Flow shop híbrido flexible

El flow shop híbrido flexible es una extensión del flow shop híbrido, donde los trabajos siguen presentando una secuencia lineal a través de las etapas, pero uno o más trabajos pueden omitir una o más etapas durante su procesamiento (Zandieh & Karimi, 2011)

La figura 5 ilustra el ambiente flow shop híbrido flexible con 5 etapas, cada una de las cuales tiene un número diferente de máquinas y se puede observar cómo algunos de los trabajos omiten una o más etapas.

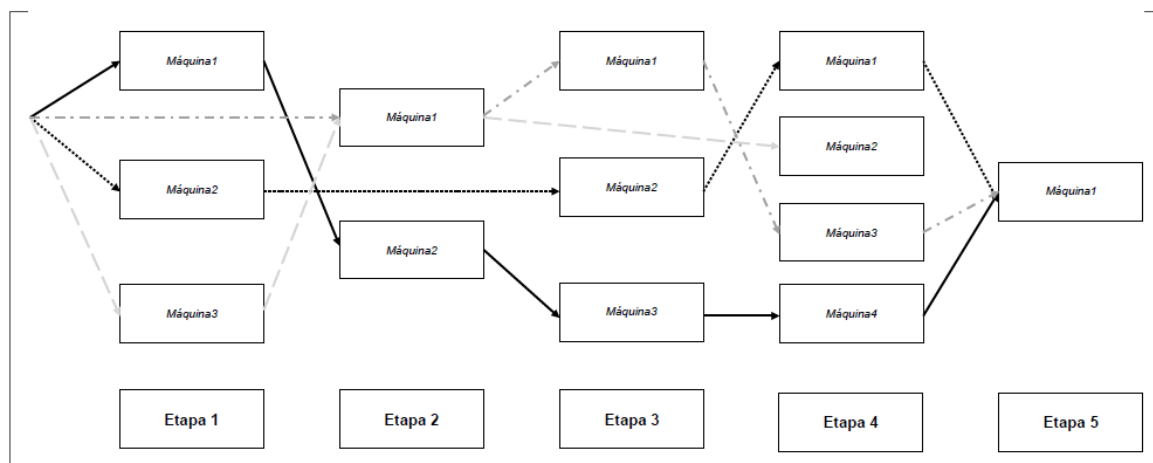


Figura 5. Representación gráfica de un sistema flow shop híbrido flexible

Fuente: López & Arango (2015).

El problema de programación en un flow shop híbrido flexible donde existen más de dos etapas, además de la cantidad de máquinas que intervienen en el proceso y el número y variedad de trabajos a producir; genera un crecimiento exponencial de la cantidad de alternativas de solución, es decir, las posibles combinaciones de asignaciones de los trabajos a las máquinas en las diferentes etapas son demasiadas y llevan a que el problema obtenga un nivel de complejidad NP-Hard, para encontrar la solución a este tipo de problemas han surgido distintas herramientas de optimización combinatoria que tienen la capacidad de encontrar soluciones de muy buena calidad en tiempos de cómputo razonables, estas son conocidas en la literatura como metaheurísticas (López et al., 2015).

4.2. Scheduling – Programación de producción

Los problemas de programación se encuentran en todos los tipos de sistemas, ya que es necesario organizar y / o distribuir el trabajo entre diferentes procesos, ciclos o participaciones que se presentan en las diferentes industrias. "La programación se refiere a la asignación de recursos limitados a tareas a lo largo del tiempo. Es un proceso de toma de decisiones que tiene como objetivo la optimización de uno o más objetivos" (Carlier & Latapie, 1991).

El scheduling es utilizado como la herramienta de nivel más detallado bajo los análisis y definiciones de la planeación de la producción, este, consiste en una toma de decisiones para determinar la organización y secuencia de la producción, considerando la disponibilidad de recursos y restricciones de la planta de producción de los estudios, ya sean físicas o por políticas de producción (Osorio, Lasso, & Ruiz, 2012).

Para este proyecto, el scheduling se dirige hacia los trabajos para programar, los cuales, de forma individual se dividen en una serie de operaciones; las operaciones de un trabajo pueden estar conectadas por restricciones de precedencia, capacidades, productos y entre otras, que dentro de la programación de la producción definen las condiciones del enrutamiento del trabajo (T'kindt & Charles, 2006)

Tradicionalmente los problemas de scheduling dirigidos a los procesos de producción y logística, se dirigen hacia la minimización de los tiempos de proceso y recursos con el objetivo de optimizar sus gastos y beneficios. Sin embargo, cada vez más se necesita el desarrollo de modelos enfocados hacia los casos reales de la industria, los cuales requieren el uso de técnicas multiobjetivo que combinen los criterios habituales con las necesidades de mejoras únicas que se generan en las diferentes industrias (Sánchez & Salido Gregorio, 2017)

Para el scheduling – programación de producción, actualmente se manejan diferentes herramientas mediante las cuales se realiza de forma eficiente, la planeación de secuencias para el uso de los recursos y la ejecución de las diferentes tareas que el

producto final requiera para su desarrollo; para este diseño y planeación de secuencias, usualmente se requiere un método de programación automática que se dirija hacia la mejora de la calidad de los servicios (QoS) y hacia el cumplimiento de los acuerdos de nivel de servicio (ANS) que garantizan la puntualidad y calidad nombradas en diferentes teorías de calidad; estos métodos de programación automática son diseñados en diferentes industrias a partir de dos métodos de solución computacional dirigidos hacia los desarrollos de sistemas de producción, los cuales son la Heurística y la Meta Heurística (Mahapatra, Dash, & Pradhan , 2017)

4.2.1. Heurísticas

“Se califica de heurístico a un procedimiento para el que se tiene un alto grado de confianza en que encuentra soluciones de alta calidad con un coste computacional razonable, aunque no se garantice su optimalidad o su factibilidad, e incluso, en algunos casos, no se llegue a establecer lo cerca que se está de dicha situación. Se usa el calificativo heurístico en contraposición a exacto, que se aplica a los procedimientos a los que se les exige que la solución aportada sea “óptima o factible” (Belén, Moreno Pérez, & Moreno Vega, 2003).

Actualmente se presentan problemas cuya cantidad de variables y posibles soluciones los clasifican como complejos para su correspondiente desarrollo y solución óptima, este tipo de casos son denominados como problemas NP, los cuales corresponden a problemas en los cuales no se garantiza encontrar una mejor solución en un tiempo polinómico razonable; es en estos tipos de problemas, en donde los métodos heurísticos son construidos y se determinan como modelos eficientes para el desarrollo de “buenas” soluciones a los diferentes problemas, sin garantizar una solución óptima para los mismos. (Suárez, 2011)

Dadas las diferentes investigaciones y formulaciones de solución para los problemas de NP, se determina que las metodologías exactas no pueden dar solución a los casos complejos dada su falta de flexibilidad y robustez, las cuales son atribuidas a los algoritmos heurísticos determinados; en los casos particulares como son los de

scheduling, se han utilizado diferentes modelos heurísticos que arrojan unos resultados eficientes que compiten con diferentes métodos y modelos anteriormente formulados, a continuación, se describen algunos de estos modelos:

- **Reglas de despacho:** Dentro de las heurísticas se encuentran las reglas de despacho, las cuales representan una serie de normas que determinan la secuenciación de los trabajos según una característica dentro del flujo de producción; las más usadas y conocidas en scheduling son (Jiménez Vargas, 2008):
 - FIFO: los primeros trabajos en llegar son los primeros en programarse.
 - EDD: earliest due date, se programan primero las órdenes en las que su fecha de entrega es más cercana.
 - SPT: shortest processing time, los trabajos con un tiempo de proceso más corto respecto a los otros son los que se programan primero.
 - LPT: longest procesing time, aquellas órdenes que tienen un tiempo de proceso más largo son las primeras en ser programadas.
 - CR: critical ratio, se calcula la relación entre el tiempo restante para la fecha de entrega y el tiempo de procesamiento del trabajo, se programan primero los trabajos de menor CR.
 - SST: shortest setup time, se programa la orden con menor tiempo de alistamiento.

- **Búsqueda Local (Local Search):** También conocida como Hill Climbing (escalar montaña), esta búsqueda se enfoca en la exploración de una parte del espacio factible de las soluciones, localizando los mejores resultados ubicados en una vecindad; una vez seleccionado el mejor resultado de la vecindad, se aplica un método de transformación para generar una nueva solución, para su posterior evaluación y posicionamiento según su calificación como nueva mejor solución de la función objetivo. (Morillo, Moreno, & Díaz, 2014)

- **Algoritmos Codiciosos o Voraces (Greedy Algorithms):** Este algoritmo consiste en la asignación de valores a la totalidad de variables de decisión de forma individual, seleccionando en cada paso la mejor variable disponible y su valor óptimo, sin embargo, este modelo no garantiza el hallazgo del valor óptimo, y tampoco una eficiencia para la solución de los problemas, por tal razón, este algoritmo sirve como herramienta de adición a unas lógicas más globales para la mejora en los tiempos de procesamiento. (Morillo et al., 2014)
- **Métodos Truncados de Branch and Bound:** Este método inicia con la generación de soluciones parciales, las cuales son acotadas y evaluadas para determinar si se consideran unas soluciones parciales, las cuales, son evaluadas con funciones de acotamiento para establecer las cotas superiores e inferiores según el problema raíz planteado, en caso que sea encontrada una mejor solución, está será determinada como solución parcialmente óptima, la cual podría continuar con una siguiente iteración de ramificaciones. (Kleinberg & Tardos, 2006)

4.2.2. Metaheurísticas

En los años 70 surgió una clase de algoritmos aproximados cuyo objetivo era combinar diferentes métodos heurísticos a un nivel más alto para lograr una exploración del espacio de búsqueda de forma eficiente y efectiva, estas técnicas son denominadas metaheurísticas y fueron introducidas por primera vez por Glover. Las metaheurísticas tienen las propiedades que se enumeran a continuación (Chicano García, 2007):

- a. Son estrategias generales que guían el proceso de búsqueda.
- b. El objetivo es una exploración eficiente del espacio de búsqueda para encontrar soluciones “casi” óptimas.

- c. Las metaheurísticas son algoritmos no exactos y generalmente no deterministas.
- d. Pueden incorporar mecanismos para evitar regiones no prometedoras del espacio de búsqueda.
- e. El esquema básico de cualquier metaheurística tiene una estructura predefinida.
- f. Hace uso del conocimiento del problema que se trata de resolver en forma de heurísticos específicos que son controlados por una estrategia de más alto nivel.
- g. No saben si llegan a la solución óptima, por lo tanto, se les debe indicar cuándo detenerse.
- h. Son algoritmos aproximativos, por lo que no garantizan la obtención de la solución óptima.
- i. Aceptan malos movimientos, es decir, se trata de procesos de búsqueda en los que cada nueva solución no es necesariamente mejor que la anterior. Algunas veces aceptan soluciones no factibles como paso intermedio para acceder a nuevas regiones no exploradas.

En este tipo de técnicas es importante el equilibrio que haya entre diversificación e intensificación. La diversificación hace referencia a la evaluación de soluciones en regiones distantes del espacio de búsqueda, también se conoce como exploración. El término intensificación indica la evaluación de soluciones en regiones acotadas y pequeñas con respecto al espacio de búsqueda centradas en el vecindario de soluciones concretas (explotación del espacio de búsqueda); el equilibrio entre estos dos aspectos tiene un gran significado ya que por un lado deben identificarse las regiones prometedoras del espacio de búsqueda y por otro lado no se debe invertir tiempo en las regiones que ya han sido exploradas o que no contienen soluciones de alta calidad (Chicano García, 2007).

Algunos tipos de metaheurísticas son (Melián, Moreno P., & Moreno V., 2003)

- a. **Metaheurísticas de relajación:** Utilizan relajaciones del modelo original, es decir, modifican el modelo para hacer el problema más fácil de resolver y esta solución favorece la del problema original.
- b. **Metaheurísticas constructivas:** Orientadas a la obtención de una solución a partir del análisis y selección paulatina de los componentes que la forman.
- c. **Metaheurísticas de búsqueda:** Aquellas que guían los procedimientos que usan transformaciones para recorrer el espacio de soluciones alternativas y explotar las estructuras de entornos asociados.
- d. **Metaheurísticas evolutivas:** Enfocadas a los procedimientos basados en conjuntos de resultados que evolucionan sobre el espacio de soluciones.

Comparadas con las heurísticas, las metaheurísticas encuentran soluciones superiores con mayores esfuerzos computacionales pero que son aceptables desde el punto de vista práctico y son útiles cuando se requiere resolver problemas de optimización para los cuales las técnicas exactas no son eficientes o no se pueden aplicar. Estas técnicas de optimización han sido utilizadas en gran variedad de aplicaciones en campos como la ingeniería, economía, teoría de juegos, ciencias computacionales, mercadeo, biología, medicina, entre otras, logrando ajustarse para cada caso y obteniendo buenos resultados (Acuña, Domínguez, & Toro, 2012)

Las técnicas metaheurísticas más conocidas son: algoritmos genéticos, búsqueda tabú, recocido simulado, colonia de hormigas, algoritmo GRASP, entre otras. A continuación, se realiza una breve explicación de las metaheurísticas mencionadas profundizando en algoritmos genéticos ya que fue la técnica empleada en el desarrollo del presente trabajo.

4.3 Algoritmos Genéticos

El desarrollo del algoritmo genético (AG) se atribuye a John Holland quien a finales de la década de los 60 desarrolló una técnica que imitaba en su funcionamiento a la evolución natural de las especies y en la genética de la población (Gestal, 2013)

La analogía directa con el comportamiento natural es porque estos algoritmos trabajan con una población de individuos, cada uno de los cuales representa una solución factible al problema planteado, a cada individuo se le asigna un valor que está relacionado con la bondad de dicha solución. En la naturaleza esto corresponde al nivel de efectividad de un organismo para competir por recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor es la probabilidad de que el mismo sea escogido para reproducirse, cruzando su material genético con otro individuo seleccionado de igual manera. Este cruce produce nuevos individuos -descendientes de los anteriores- que contienen material genético de sus padres. De esta manera se genera una nueva población de posibles soluciones, la cual reemplaza la anterior y verifica la proporción de buenas características en comparación con la población anterior. Así, a lo largo de las generaciones las buenas propiedades se propagan a través de la población. Fomentando el cruce de los individuos con mayores aptitudes, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el algoritmo está bien diseñado, la población convergerá hacia una solución óptima del problema (Moujahid, Inza, & Larrañaga)

La estrategia de solución mediante esta metaheurística es aplicable a problemas de distinta índole, siendo un método robusto que, aunque no siempre encuentra una solución óptima, si encuentra soluciones aceptables y en tiempos razonables.

En los problemas de programación de producción los algoritmos genéticos han sido usados con éxito, encontrando soluciones muy cercanas a la óptima adaptándose a las particularidades de cada problema.

Existen elementos fundamentales en el algoritmo genético: la representación de los individuos (cromosoma), la función objetivo o función de aptitud que mide si el individuo es apto como solución, la población inicial, el reemplazo de la población y el criterio de

parada (Zames, Ajlouni, Hollan, Hills, & Golberg, 1981). También existen tres operaciones esenciales que se aplican a los cromosomas, estas son, selección, cruce y mutación.

Cromosoma: Esta codificación debe contener información acerca de la solución que representa y se puede construir de varias formas, la elección de la forma de codificar depende del problema a resolver.

El conjunto de parámetros representado por un cromosoma recibe el nombre de genotipo, cada uno de los bits pertenecientes al cromosoma suelen recibir el nombre de alelo (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010).

Las clases de representación de cromosomas son (Arranz & Parra):

- a. **Codificación binaria:** Es la más usada debido a que los primeros algoritmos utilizaron este tipo de codificación, en este caso el cromosoma es una cadena de bits (0 o 1). Este tipo de cromosoma se utiliza por ejemplo en el problema de la mochila, en este problema se tiene una mochila y un conjunto de elementos que tienen determinado peso y valor, y que deben introducirse dentro de la misma; el objetivo es conseguir que la suma de los valores sea máxima y que la suma de los pesos de los elementos no supere la capacidad de la mochila.
- b. **Codificación numérica:** En esta codificación se usan cadenas de números que representan un número en secuencia, se utiliza en problemas donde hay que ordenar algo. Un ejemplo de utilización de esta forma de cromosoma es en el problema del agente viajero.
- c. **Codificación por valor directo:** Es utilizada en caso de resolución de problemas en los que se necesite el uso de valores de cifrado, pudiendo ser, números decimales, cadenas de caracteres o incluso una combinación de

varios de ellos. Una aplicación de este tipo de codificación se da en la resolución de problemas para la búsqueda de pesos para las redes neuronales.

- d. **Codificación de árbol:** Se utiliza principalmente en el desarrollo de programas o expresiones para programación genética, en este caso cada cromosoma será un árbol con ciertos objetos.

Para la programación de producción se han propuesto métodos como: codificación basada en los trabajos, codificación basada en las máquinas y codificación basada en la operación (Chen, Pan, & Lin, 2008).

Población inicial: Corresponde al conjunto de posibles soluciones del problema, es sometida a evaluación y se seleccionan los individuos (cromosomas) más aptos según su aptitud para dar origen a nuevos cromosomas y a nuevas poblaciones.

Puede ser generada de forma aleatoria, sin incluir ningún conocimiento previo, o mediante heurísticas que introducen conocimiento previo de un conjunto probable de buenas soluciones, o tomada de una base de conocimiento de alguna solución anterior de otra instancia del problema (Salazar & Sarzuri, 2015). En la literatura se analiza que el iniciar la población con heurísticas puede tener como inconveniente que el algoritmo converja de forma prematura, es decir, se concentre en óptimos locales (Zames et al., 1981).

Es importante que la población inicial tenga un tamaño lo suficientemente grande para garantizar la diversidad de soluciones. En problemas de scheduling el tamaño puede depender del número de productos a programar y la cantidad de máquinas que hacen parte del proceso productivo, las poblaciones pequeñas corren el riesgo de no cubrir adecuadamente el espacio de búsqueda, mientras que trabajar con poblaciones de gran tamaño pueden llevar a un costo computacional elevado, sobre todo a la hora de calcular el valor de aptitud de los individuos; aunque no se recomienda que el tamaño de sea

mayor a 200 cromosomas tampoco es recomendable un tamaño menor a 100 para evitar encontrar una solución lejana a la óptima y una convergencia prematura, en general se recomiendan poblaciones que contengan entre 100 y 200 individuos (Aguilar, 2016)

En la figura 6 se puede ver la representación gráfica de un cromosoma y una población.

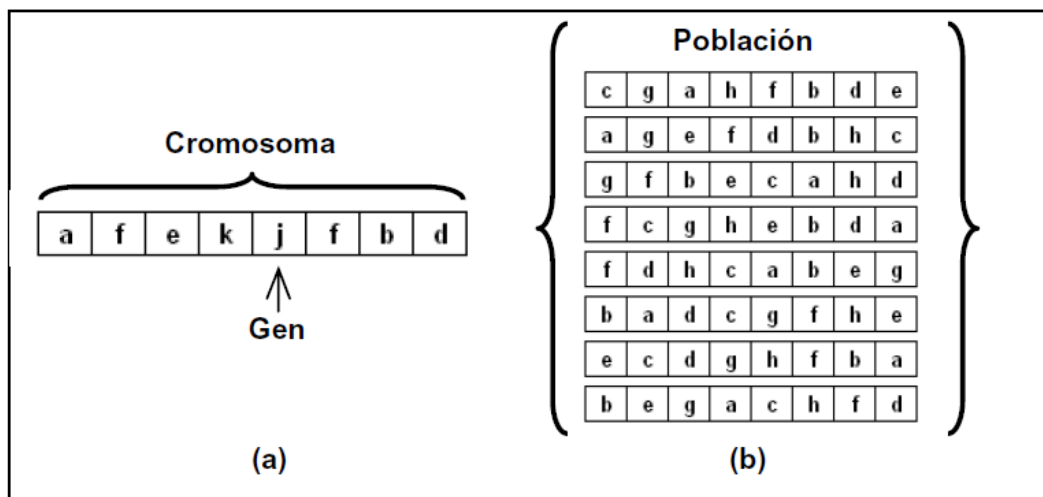


Figura 6. Representación de la población en un algoritmo genético

Fuente: Vélez & Montoya (2007).

Función fitness: Para el correcto funcionamiento de un algoritmo genético es necesario contar con un método que indique si los individuos de la población representan o no buenas soluciones al problema planteado, por lo tanto, para cada tipo de problema puede existir un método distinto. La función de evaluación o fitness establece una medida numérica de la bondad de una solución, esta medida recibe el nombre de ajuste. En el funcionamiento de los algoritmos genéticos esta medición servirá para controlar la aplicación de los operadores genéticos (Gestal et al., 2010).

Según Koza (1992) citado por (Gestal, et al., 2010) existen cuatro tipos de ajuste o fitness:

- a. **Fitness puro:** Medida de ajuste establecida en la terminología natural del propio problema. En los problemas de maximización los individuos con un

valor de fitness puro elevado serán los más interesantes, mientras que en un problema de minimización interesarán los individuos con un fitness puro reducido.

$$r(i, t) = \sum_{j=1}^{Nc} |s(i, j) - c(i, j)|$$

Donde:

$r(i, t)$: bondad del individuo i en la generación t .

$s(i, j)$: valor deseado para el individuo i en el caso j .

$c(i, j)$: valor obtenido por el individuo i en el caso j .

Nc : número de casos.

b. Fitness estandarizado: Para solucionar la dualidad ante problemas de maximización y minimización se modifica el ajuste puro

$$s(i, t) = \begin{cases} r(i, t) & \text{minimización} \\ r_{max} - r(i, t) & \text{maximización} \end{cases}$$

En el caso de problemas de minimización se emplea directamente la medida de fitness puro. Si el problema es de maximización se resta de una cota superior r_{max} del error del fitness puro. Empleando esta medida, la bondad de un individuo será menor cuanto más cercano esté a cero el valor del ajuste. Por lo tanto, dentro de la generación t , un individuo i siempre será mejor que uno j si se verifica que $s(i, t) < s(j, t)$.

c. Fitness ajustado: Se obtiene aplicando la transformación de la siguiente ecuación al fitness estandarizado

$$a(i, t) = \frac{1}{1 + s(i, t)}$$

De esta forma, la función objetivo tomará siempre valores del intervalo $[0,1]$. Cuanto más se aproxime la función objetivo de un individuo a 1, mayor será su aptitud.

- d. Fitness normalizado:** Este fitness indica la bondad de una solución con respecto al resto de soluciones representadas en la población. Considerando una población de tamaño M se tiene la siguiente ecuación

$$n(i, t) = \frac{a(i, t)}{\sum_{k=1}^N a(k, t)}$$

Siempre tomará valores en el intervalo $[0,1]$, con mejores individuos cuanto más próximos estén a la unidad, un valor cercano a 1 no sólo indica que ese individuo representa una buena solución al problema, sino, además que es una solución destacada respecto a las proporcionadas por el resto de la población. La suma de los valores de fitness normalizado de una población siempre da 1.

Reemplazo de la población: Establece el criterio de los supervivientes en cada iteración generacional. Dependiendo de los métodos de selección y cruce que se utilicen es posible que al final de cada iteración el tamaño de la población inicial aumente, para evitar esto, se puede lograr una reducción del tamaño de la población escogiendo varios individuos y los descendientes que estos han generado. Existen varias formas de generar la población de reemplazo (Corral, 2017):

- a. Generacional:** Toda la población es reemplazada en cada iteración. Cada individuo sobrevive solo una generación y todos los hijos reemplazan a todos los padres.

- b. Estado-estacionario:** Solo unos pocos individuos son reemplazados en cada generación, típicamente se genera uno o dos hijos y solo uno o dos individuos son reemplazados.
- c. Gap generacional:** Criterio combinado del reemplazo generacional y del reemplazo estado-estacionario. Se define un porcentaje de las soluciones que serán reemplazados cada generación, por evaluación del fitness.
- d. Reemplazo catastrofista:** Cada cierto tiempo se elimina una cantidad de individuos.

Criterio de parada: El proceso iterativo del algoritmo finaliza si se cumple determinada condición; que se haya procesado una cierta cantidad de generaciones, que la aptitud de una cantidad supere un valor definido, o que el valor de la función objetivo no mejore en cierto número de iteraciones (Solari & Ocampo, 2006 citado por (López J. , 2013).

Selección: Esta fase es la responsable de escoger a los individuos que se reproducirán y generarán una nueva población teniendo en cuenta que los buenos rasgos de los mejores se transmitirán a esta (Gesta et al., 2010). Esta selección se puede realizar de varias formas:

- a. Selección por ruleta:** Propuesto por DeJong, es posiblemente el método más usado desde los orígenes de los algoritmos genéticos. A cada uno de los cromosomas de la población se le asigna una parte proporcional a su valor de aptitud de una ruleta, de tal forma que la suma de todos los porcentajes sea la unidad. El cromosoma con mejor ajuste saldrá con mayor probabilidad (Arranz & Parra). Generalmente, la población está ordenada en base a la aptitud, por lo que las porciones más grandes se encuentran al inicio de la ruleta. Para elegir un individuo basta con generar un número aleatorio comprendido en el intervalo $[0,1]$ y devolver el individuo situado en esa posición de la ruleta. Esta posición se suele obtener recorriendo los cromosomas y acumulando sus proporciones de ruleta hasta que la suma exceda el valor obtenido (Gestal et al., 2010).

El algoritmo es simple, pero ineficiente a medida que aumenta el tamaño de la población, además, presenta el problema de que el individuo menos apto puede ser seleccionado más de una vez o que si un cromosoma tiene un 90% de

posibilidades de ser seleccionado, el resto apenas saldrá, lo que reduce la diversidad genética (Coello, 2004).

- b. Selección por torneo:** Este método consiste en una comparación directa entre individuos. Existen dos versiones de este tipo de selección, la determinística y la probabilística; la versión determinística consiste en seleccionar al azar un número p de individuos (generalmente se escoge un $p=2$) y de entre los elegidos se escoge al más apto para pasar a la siguiente generación. En la versión probabilística en vez de escoger siempre el mejor se genera un número aleatorio en el intervalo $[0,1]$, si es mayor que un parámetro p fijado se escoge el individuo más apto y en caso contrario el menos apto, generalmente p toma valores en el rango $0.5 < p \leq 1$ (Gestal, 2013).

- c. Selección por rango:** A cada cromosoma se le asigna un rango numérico basado en su aptitud y la selección se realiza con base a este ranking. La ventaja de este tipo de selección es que va a producir más variedad genética comparada con el método de ruleta, sin embargo, la convergencia puede ser más lenta debido a que no existe mayor diferencia entre el mejor individuo y el resto (Arranz & Parra).

- d. Selección elitista:** Esta forma de selección copia el mejor cromosoma o alguno de los mejores a la nueva población. El elitismo puede mejorar el funcionamiento de los algoritmos al evitar que se pierda la mejor solución. Una variación de esta selección es que el mejor cromosoma solo se copie a la siguiente generación en caso de que tras un cruce/mutación no se haya generado un cromosoma mejor (Arranz & Parra)

- e. Selección jerárquica:** El objetivo de esta técnica es disminuir la presión de la selección. Los individuos se clasifican por su aptitud, pero se seleccionan con base a su jerarquía, el uso de jerarquía hace que no se requiera escalar la aptitud, puesto que las diferencias entre las aptitudes absolutas se diluyen (Coello, 2004). Las evaluaciones de los primeros niveles son más rápidas y menos discriminatorias, mientras que los que sobreviven hasta niveles más altos son

evaluados de forma más rigurosa. La ventaja de este método es que reduce el tiempo total de cálculo al utilizar una evaluación más rápida y menos selectiva para eliminar a la mayoría de los individuos que se muestran poco o nada prometedores, y sometiendo a una evaluación más rigurosa y computacionalmente más costosa sólo a los que sobreviven a la prueba inicial (Arranz & Parra).

Cruce: Consiste en mezclar el material genético de dos cromosomas que serán progenitores y que han sido seleccionados (utilizando cualquier técnica). Por lo general los cromosomas son aleatoriamente divididos y mezclados, por lo que ciertos genes de los descendientes provienen de un progenitor, mientras otros provienen del otro, es decir, los descendientes tendrán en su genotipo información de ambos progenitores. El cruce garantiza la conservación de las características que hacen a un individuo fuerte para sobrevivir y la eliminación de las que lo hacen débil (Aguilar, 2016)

Existen diferentes técnicas de cruce:

- a. **Cruce de un punto:** Es la más sencilla de las técnicas, una vez seleccionados los dos padres se cortan los cromosomas por un punto seleccionado aleatoriamente para generar dos segmentos diferenciados en cada uno de ellos: la cabeza y la cola, se intercambian las colas entre los individuos para generar los nuevos descendientes; de esta manera los cromosomas hijos heredan información genética de los cromosomas padres (Gestal et al., 2010).

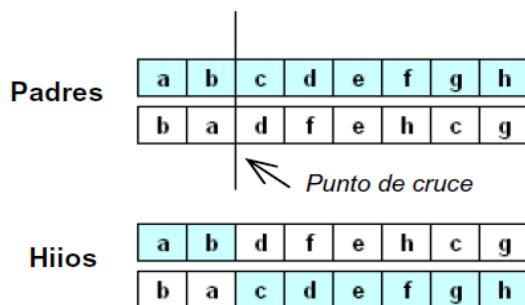


Figura 7. Representación cruce de un punto

Fuente: Vélez & Montoya (2007)

- b. Cruce de dos puntos:** DeJong fue el primero en implementar un cruce de n puntos, como una generalización del cruce de un punto, el valor $n=2$ es el que minimiza los efectos disruptivos del cruce y de ahí que sea usado con frecuencia. Se copia al descendiente los genes de un cromosoma padre desde el inicio hasta el primer punto de cruce, los genes del otro progenitor desde el primer punto de cruce hasta el segundo y del segundo punto de cruce hasta el final se copiará del otro padre (Coello, 2004).

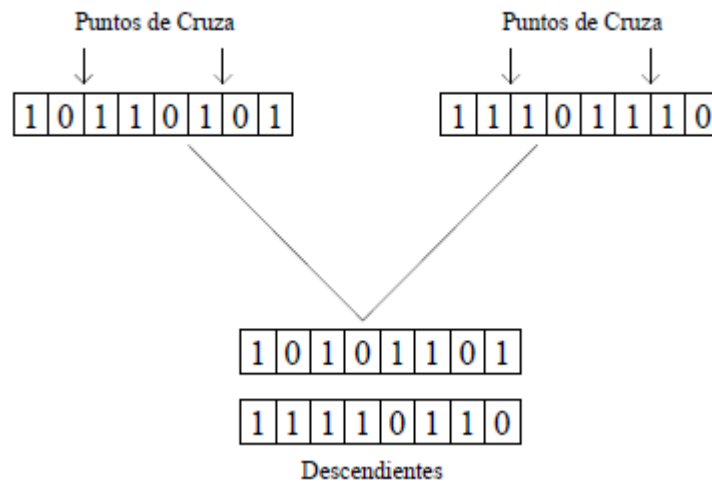


Figura 8. Cruce de dos puntos

Fuente: Coello (2004).

- c. Cruce uniforme:** Cada gen del descendiente se obtiene de cualquiera de los padres de forma aleatoria (Arranz & Parra).

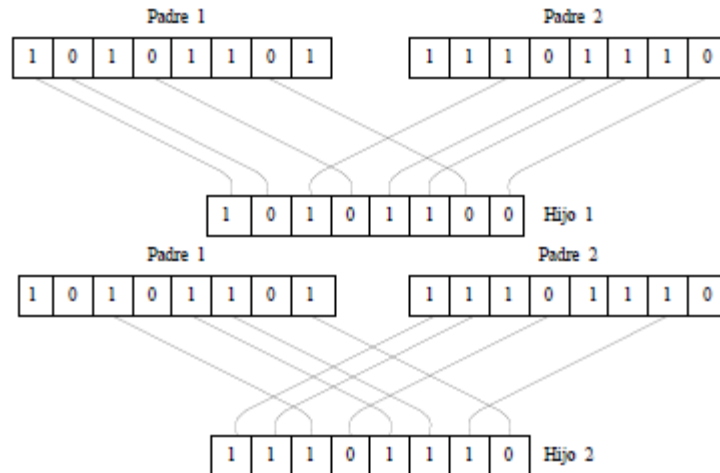


Figura 9. Cruce uniforme

Fuente: Coello (2004).

- d. **Operador Partially Mapped Crossover (PMX):** Selecciona dos puntos de cruce de forma aleatoria entre 1 y $(n-1)$ donde n es la longitud del cromosoma. Las subcadenas de código genético determinadas por los puntos de corte son denominadas sección de mapeo, estas son intercambiadas para generar los nuevos hijos. Se determina la relación de mapeo correspondientes entre las dos secciones, y mediante el uso de esta relación se certifican los descendientes (Salazar & Sarzuri, 2015).



Figura 10. Operador de cruzamiento PMX

Fuente: Salazar & Sarzuri (2015).

Mutación: Tras el cruce tiene lugar la mutación; este operador transforma levemente un individuo, evita la convergencia prematura y cambia la dirección de búsqueda. Se eligen dos posiciones en forma aleatoria entre 1 y n , los genes de cada posición son intercambiados. La probabilidad de mutación es baja, generalmente menor al 1%, esto es porque los individuos suelen tener un ajuste menor después de mutados, sin embargo, se realizan mutaciones para garantizar que ningún punto del espacio de búsqueda tenga probabilidad nula de ser evaluado (Salazar & Sarzuri, 2015).

4.3.1 Algoritmo Genético Multiobjetivo

Dentro del contexto de los problemas múltiples evidenciados en las diferentes investigaciones, se plantea la optimización multiobjetivo, la cual involucra múltiples objetivos frecuentemente en conflicto, que deben ser solucionados de forma simultánea (Esquivel, y otros, 2002)

Estableciendo el problema que cuenta con $k \geq 2$ FOB (de ahora en adelante, funciones objetivo) a ser maximizadas o minimizadas de forma simultánea, la solución de este problema busca establecer la solución x perteneciente al grupo de soluciones factibles X que den solución al problema:

$$\text{Min } x \in X \ f(x) = \{f_1(x), f_2(x), \dots, f_k(x)\}$$

Teniendo en cuenta que los objetivos en su mayoría se presentan en conflicto, no se encuentra un valor x que optimice todos los objetivos de forma simultánea, lo cual requiere métodos de solución diferentes a los lineales normalmente establecidos (Haral, Chen, Ferrell Jr, & Kurz, 2006) dada la consideración de un multiobjetivo, estos problemas son planeados únicamente cuando las soluciones de cada objetivo no presenten soluciones óptimas de la totalidad de los objetivos planteados (Lagunas Jiménez, Moo Yam, & Ortiz Moctezuma, 2013).

Para la solución de los problemas multiobjetivo, en 1985 J. David Schaffer propone el primer algoritmo genético multiobjetivo (llamado VEGA – Algoritmos genéticos evaluados por vectores), cuya idea principal se ve plasmada en la división de la población en subpoblaciones (de igual tamaño), para cada subpoblación, se asigna un único objetivo para su respectiva búsqueda de solución; dando así, un procedimiento de selección de

mejor hijo independiente por cada uno de los objetivos, así como la ejecución de los cruces a través de límites de subpoblación. (Schaffer, 1985)

Posterior al desarrollo del VEGA, se presentan desarrollos adicionales de diferentes algoritmos multiobjetivo como el algoritmo genético de Pareto Niched, algoritmo genético ponderado aleatorio (RWGA), algoritmo genético de clasificación no dominada (NSGA), algoritmo evolutivo de pareto por fuerza (SPEA), algoritmo genético de clasificación no dominada V2 (NSGA-II), y algoritmo evolutivo de objetivos múltiples (MEA) entre otros, diseñados con base en la lógica de multiobjetivo. (Abdullah Konak, 2006).

Los algoritmos genéticos creados para la solución de los casos multiobjetivo difieren en función de diferentes procedimientos modelados como lo son la aptitud, el elitismo y el enfoque de diversificación, los cuales representan ventajas y desventajas de acuerdo a los diferentes componentes del modelo a optimizar, o los objetivos planteados para el problema, sin embargo, muchos investigadores que aplicaron GA multiobjetivo a sus problemas han desarrollado diferentes diseños de algoritmos personalizados adaptando estrategias de varios algoritmos genéticos multiobjetivo. (Abdullah Konak, 2006).

Dentro de las diferentes aplicaciones del principio del modelo MOGA, se encuentra la aplicación de dos herramientas de selección en paralelo, con modelos independientes, uno, enfocado hacia el modelo de Pareto, y el segundo, dirigido hacia la selección por aptitud; para este problema, se realiza un desarrollo de población inicial y un cruce – mutación basados en los dos métodos, dando como resultado, una mejor solución basada en el modelo de Pareto (Ponnambalam, 2003).

Por otro lado, en el desarrollo de un algoritmo genético multiobjetivo con local search dirigido a un Flow Shop se evidencia la optimización de dos objetivos (Tiempo total de ciclo, máxima demora y tiempo total de flujo) con un algoritmo genético hibridizado con una técnica de búsqueda local paralela (Arroyo & Claudio, 2005).

Adicionalmente, a través del proyecto “*A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs*”, se evidencia el desarrollo de un algoritmo multiobjetivo ejecutado, con una posterior aplicación de dos heurísticas (Job-Index-Based Insertion Scheme (JIBIS) – (Esquema de

inserción basado en el índice de trabajos) y Overall-Seed Sequence-Based Insertion Scheme (OSSBIS) – (Esquema de inserción basada en la secuencia general de semillas)), encontrando secuencias de inicio que cumplan con los objetivos y restricciones planteados en el problema, dando como resultado unas secuencias no dominadas, dando cobertura a un margen de decisiones que cumplen uniformemente los objetivos en cada una de las generaciones. (Varadharajan & Rajendran, 2004)

En la revisión de literatura se encontró que la función objetivo más estudiada ha sido el tiempo máximo requerido para completar todos los trabajos, función denominada makespan; muy pocos han estudiado el criterio de trabajos tardíos y la mayoría utiliza como método de solución el algoritmo genético, como se muestra a continuación, autores como (Zobolas, Tarantilis, & Ioannou, 2009) en su investigación desarrollan una metodología que implica la combinación de las metaheurísticas algoritmo genético y vecino más próximo para minimizar el makespan en un ambiente flow shop de permutación. (Karmakar & Manhaty) en su trabajo aplican la teoría de restricciones (TOC) y un algoritmo genético con el objetivo de minimizar el makespan en una compañía de pintura que presenta una configuración flow shop flexible en la que se deben programar nueve productos en ocho máquinas y cinco etapas. (Gupta & Chauhan, 2015) desarrollan un algoritmo basado e el peso de las máquinas que se reduce en cada etapa para generar diferentes combinaciones de secuencias para minimizar el makespan en un ambiente flow shop, el algoritmo convierte el problema original de m -máquinas en $m-1$ problemas artificiales de dos máquinas, al primer problema artificial de 2 máquinas se aplica la regla de Johnson para determinar la secuencia de trabajos y el proceso se repite reduciendo el parámetro de peso hasta que se encuentren las secuencias $m-1$ (Oguz & Ercan, 2005) estudian un ambiente flow shop híbrido con tareas multiprocesadas en el que usan un algoritmo genético y lo comparan con la metaheurística búsqueda tabú para encontrar el mejor makespan. (Toro Ocampo, Restrepo Grisales, & Granada Echeverri, 2006) resuelven el problema de minimización del tiempo total de producción utilizando el algoritmo genético modificado de Chu-Beasley. (Perea González, 2015) en su trabajo diseña un algoritmo genético que permite dar solución al problema de Aceptación de Órdenes y Scheduling (OAS) o Scheduling admitiendo rechazos (MSR) en configuraciones flow shop flexibles con máquinas no relacionadas con el objetivo de

maximizar la utilidad. (Salazar Hornig, E. & Sarzuri Guarachi, R., 2015) en su artículo consideran un entorno flow shop flexible con tiempos de preparación dependientes de la secuencia en el que usan un algoritmo genético mejorado para minimizar la tardanza total del mismo. Otro punto que se pudo observar en la revisión del estado del arte es que son pocos los autores que enfocan la investigación a problemas multiobjetivo, aun cuando; los problemas reales de ingeniería tienen múltiples objetivos. (Allahverdi & Al-Anzi, 2008) proponen tres métodos; recocido simulado, colonia de hormigas y evolución diferencial autoadaptativa para minimizar tiempo de finalización medio y makespan. (Tasgetiren, Liang, Sevkli, & Gencyilmaz, 2007) investigan el uso de un algoritmo de optimización de enjambre de partículas para resolver el problema de secuenciación en un flow shop de permutación con el objetivo de minimizar el tiempo de fabricación y el tiempo total de trabajo. (Jarboui, Eddaly, & Siarry, 2011) buscan minimizar el makespan y el tiempo de flujo total en un ambiente flow shop sin espera para más de dos máquinas, y para esto proponen un algoritmo genético híbrido en el que usan la metodología de vecino más próximo como mecanismo de mejora en el último paso del AG. En la literatura existente son escasos los trabajos que abordan sistemas reales de manufactura, la mayoría de las investigaciones son desarrolladas sobre conjuntos de problemas de pruebas generados aleatoriamente; es por esto que este trabajo plantea la solución de dos objetivos (makespan y trabajos tardíos) y además es desarrollado en un ambiente real de manufactura (textiles técnicos).

5. METODOLOGÍA PROPUESTA BASADA EN AG

5.1. Supuestos

Para la construcción de la metodología se tienen en cuenta los siguientes supuestos:

- a. Cada máquina tiene la capacidad de procesar como máximo 1 montaje (conjunto de artículos por tipo y color) a la vez.
- b. Cada montaje es procesado solamente en una máquina a la vez.

- c. Toda máquina se caracterizará como disponible en el momento $T=0$, y no se tendrán en cuenta los tiempos de mantenimiento.
- d. No se permiten interrupciones durante el procesamiento de los montajes en cada máquina.
- e. Los tiempos de procesamiento son determinísticos y previamente conocidos.
- f. Los tiempos de montaje y alistamiento de máquina, en cada una de las etapas están inmersos en los tiempos de procesamiento.
- g. La flexibilidad del flow shop híbrido en el modelo se verá reflejada a partir de la asignación de un tiempo de procesamiento igual a 0 (cero) en la (s) etapas (s) donde el producto no requiera de su procesamiento.
- h. No se tendrán en cuenta los tiempos de transporte entre etapas, dado que no son relevantes en el tiempo del proceso.
- i. Todos los trabajos tienen la misma prioridad para su procesamiento.

5.2 Conjuntos, parámetros variables y función objetivo

a. Conjuntos:

γ = conjunto de artículos

m = conjunto de máquinas

t = conjunto de turnos

j = conjunto de posiciones

P_γ = agrupación de artículos por tipo de producto y color

b. Parámetros:

α = iteraciones

β = subpoblaciones

μ = cromosomas

ω = % de mutación

η = color

λ = tipo de artículo

R_γ = tiempo de procesamiento del artículo γ

F_γ = due date del artículo γ

N_γ = cantidad de metros del artículo γ

c. Variables:

$$T_\gamma = \begin{cases} 1, & \text{si el artículo } \gamma \text{ es asignado al turno } t \\ 0, & \text{en caso contrario} \end{cases}$$

$$X_{\gamma tm} = \begin{cases} 1, & \text{si el artículo } \gamma \text{ es asignado al turno } t \text{ en la máquina } m \\ 0, & \text{en caso contrario} \end{cases}$$

$$G_{\gamma mj} = \begin{cases} 1, & \text{si el artículo } \gamma \text{ es asignado a la máquina } m \text{ en la posición } j \\ 0, & \text{en caso contrario} \end{cases}$$

Q_γ = tiempo de terminación del artículo γ

$$U(\gamma) = \begin{cases} 1, & \text{si el artículo es tardío} \\ 0, & \text{en caso contrario} \end{cases}$$

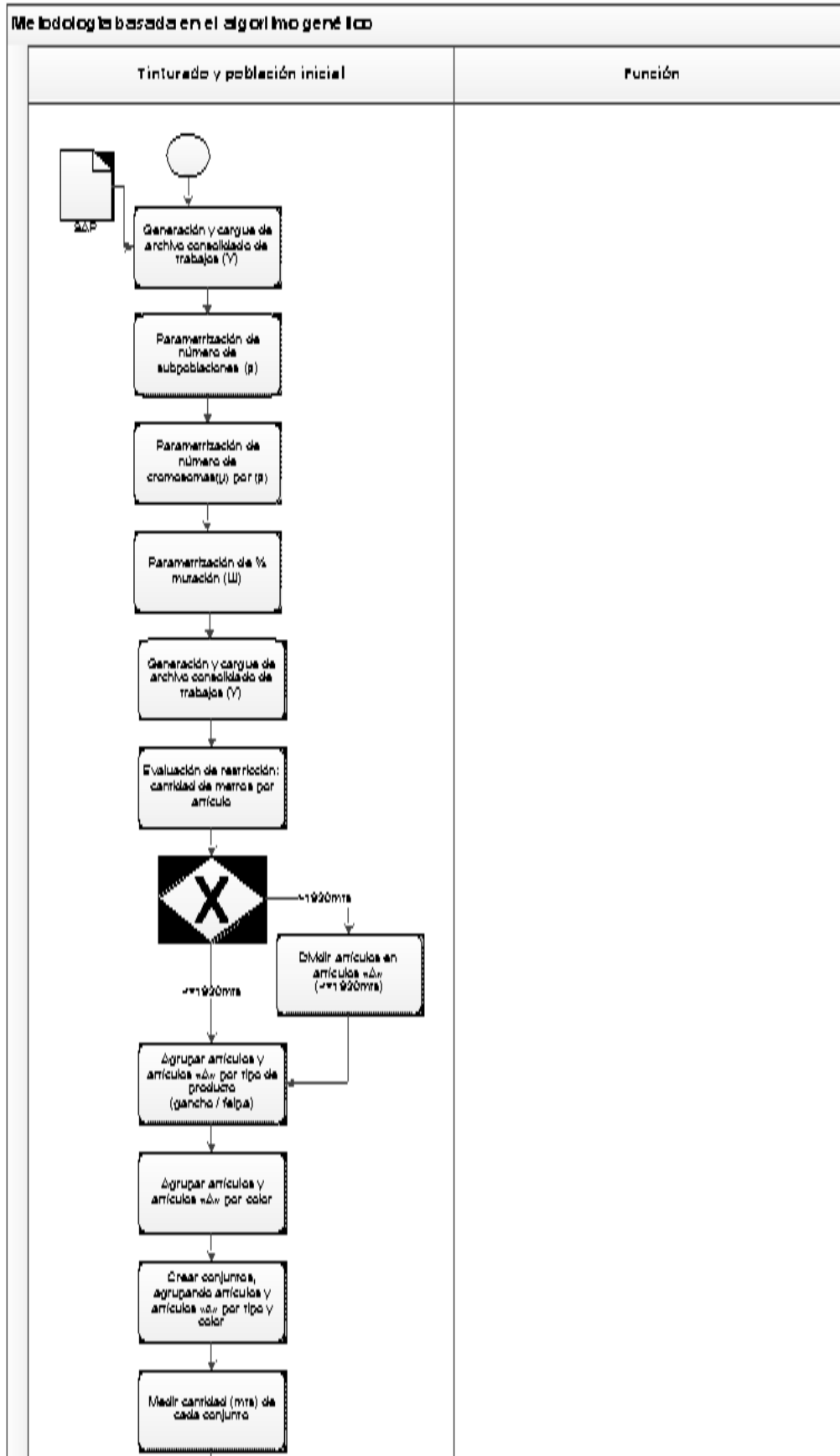
d. Funciones objetivo:

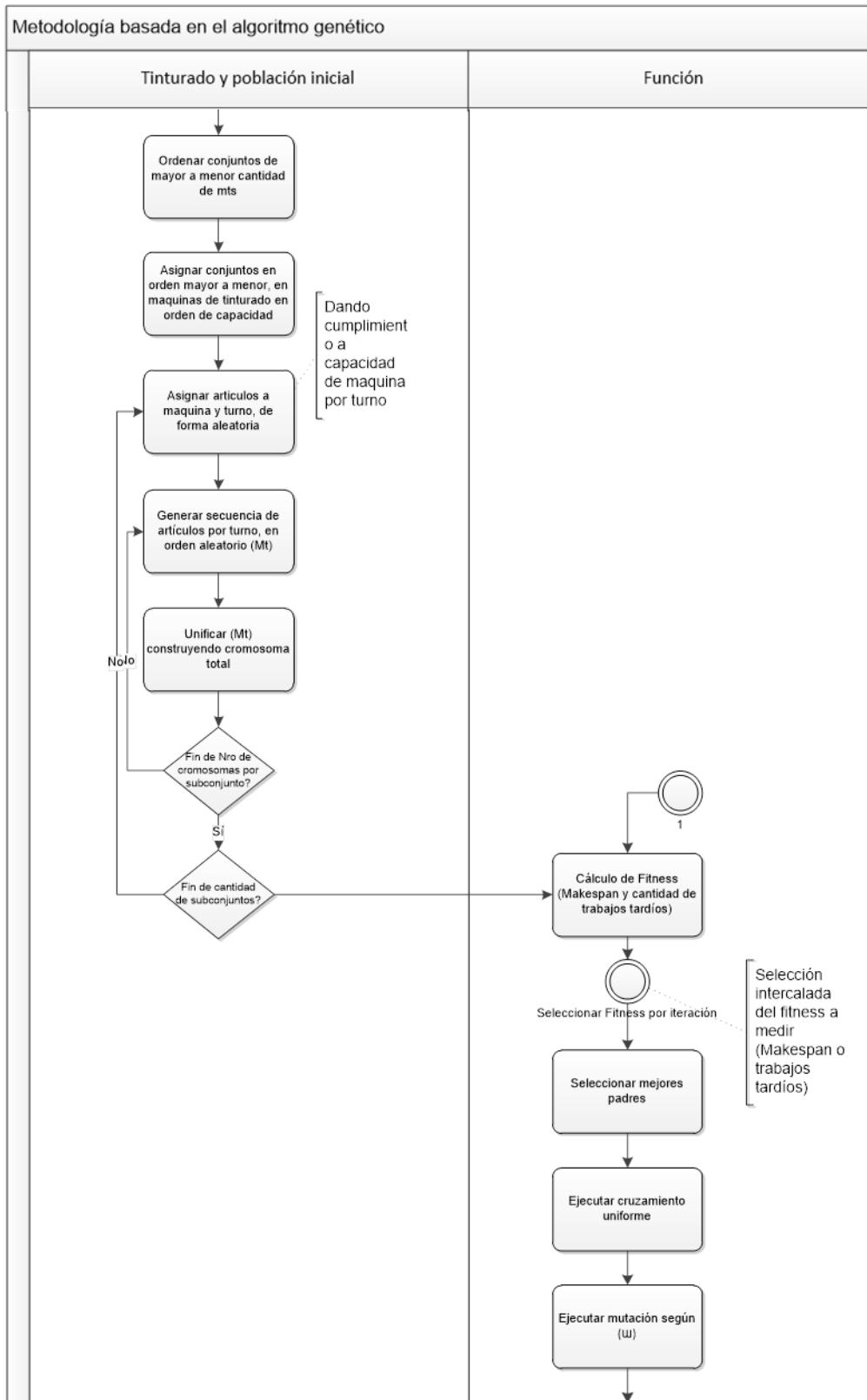
$$\text{Min } \sum U(\gamma)$$

$$\text{Min } \sum Q(\gamma)$$

5.3 Diagrama de flujo

La metodología propuesta se basa en el modelo de algoritmos genéticos descrita a través de la figura 11 – Metodología basada en el algoritmo genético.





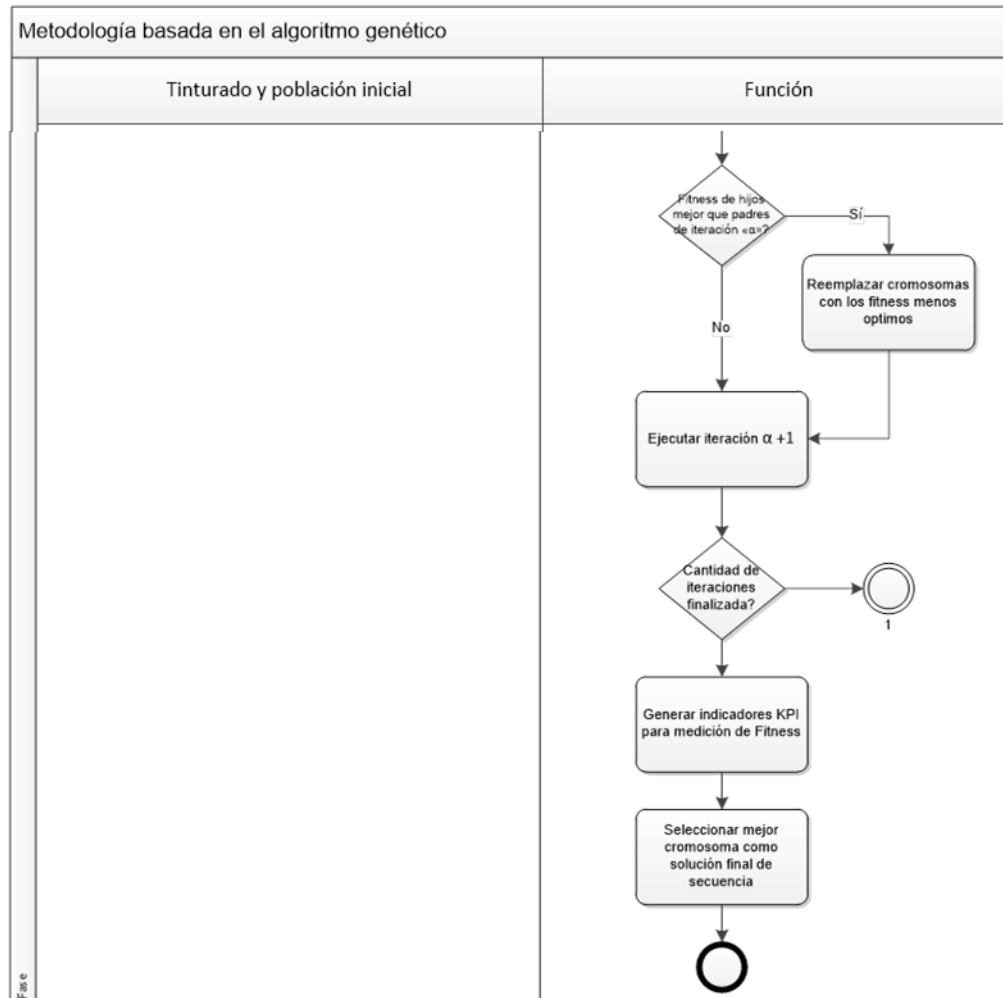


Figura 11. Metodología basada en el algoritmo genético

Fuente: Elaboración propia

5.4 Metodología propuesta

A continuación, la descripción del modelo basado en la lógica de los algoritmos genéticos:

5.4.1 Definición de conceptos

- Orden de Venta: Documento formal manejado para registrar los pedidos de venta en el ERP, el cual, contiene el listado de artículos solicitados por parte del cliente.
- Artículo (γ): Ítem de venta

- Tipo de artículo (λ): Producto felpa o gancho.
- Color (η): Color del producto a fabricar.
- Conjuntos (P_γ): Agrupación de artículos por tipo de producto y color
- Subpoblaciones (β): Asignación de artículos a cada turno y máquina de tintorería.

5.4.2 Descripción del modelo

El modelo propuesto se divide en dos fases, la primera corresponde al método de asignación de artículos (γ) al proceso de tintorería donde se utiliza el principio de la regla de despacho LPT dando como resultado una matriz que representa la asignación de los artículos (γ) en las tres máquinas (m) de tintorería y en los turnos (t) con duración igual a 3 horas y capacidad de tres turnos al día. La segunda fase abarca la programación de artículos (γ) desde el proceso de magueba (proceso posterior a la tintorería) hasta el empaque (proceso final) mediante el uso de un algoritmo genético.

5.4.2.1 Fase 1. Asignación de artículos a proceso de tintorería

1. Realizar consolidación de la totalidad de artículos (γ) pertenecientes a las órdenes de venta cargadas en SAP para su programación de producción.
2. Consultar la cantidad de metros de cada artículo (γ), y verificar que no supere la capacidad máxima de metros (capacidad máxima: 1920m, correspondiente a la capacidad instalada de la máquina 1 de tintorería).

2.1 Si la cantidad de metros de un artículo (γ) es mayor a 1920m, se divide el número de metros del artículo (γ) en 1920 generando un artículo (γ_k). En la figura 12 es visible la transformación de la información a procesar cuando los artículos superan la cantidad de 1920m, en este ejemplo los artículos con Id 2 y 6 se dividen y se generan los artículos 2_0 y 6_0 con cantidad de metros igual a 1920 y los artículos 2_1 y 6_1 con la cantidad de metros restantes al valor original.

2.2 Si la cantidad de metros del artículo (γ) es menor o igual a 1920m se conserva el Id del artículo como único.

Id Artículo	Tamaño	Id Artículo	Tamaño
1	1200	1	1200
2	1950	2_0	1920
3	100	3	100
4	1200	4	1200
5	1000	5	1000
6	1990	6_0	1920
7	1200	7	1200
8	200	8	200
9	700	9	700
10	940	10	940
11	800	11	800
12	160	6_1	70
13	120	2_1	30
		12	160
		13	120

Figura 12. Transformación de artículos a procesar cuando el tamaño es mayor a 1920m
Fuente: Elaboración propia

3. Agrupar los artículos (γ) y los artículos (γ_k) por tipo de producto (λ) (felpa y gancho), creando los conjuntos de artículos de tipo de producto (P_λ).
4. Desglosar los conjuntos de tipo de producto (P_λ), en agrupación por color (η) generando un conjunto ($P_{\lambda\eta}$).
5. Realizar la sumatoria de metros correspondientes a los artículos (γ) y artículos (γ_k), que contiene cada conjunto ($P_{\lambda\eta}$).
6. Ordenar los conjuntos ($P_{\lambda\eta}$) de mayor a menor cantidad en metros.
7. Asignar conjuntos de artículos ($P_{\lambda\eta}$) en orden de mayor a menor cantidad de metros, a la maquinaria en orden de mayor a menor capacidad.

Capacidades		Turno 1	Turno 2	Turno 3	Turno N
1920	Máquina 1					
960	Máquina 2					
160	Máquina 3					

- La máquina 1, tiene una capacidad de 1920 metros por turno
- Cada máquina, tendrá un N de turnos que serán asignados según la cantidad de artículos / conjuntos a producir

Figura 13. Tabla asignación de artículos a máquinas y turnos
Fuente: Elaboración propia

8. Generar una subpoblación (β) (cantidad de subpoblaciones parametrizable) seleccionando de forma aleatoria los artículos (γ) que contiene el conjunto ($P_{\lambda\eta}$) y asignándolos a cada máquina (m) y a cada turno (t) teniendo en cuenta que cada uno de los artículos (γ) a asignar, deberá evaluar la capacidad de cada turno de la máquina, si el tamaño del artículo (γ) es menor o igual a la capacidad del turno (t) en la maquina (m), este deberá ser asignado a la máquina (m) que cumpla con estas restricciones, esto, aprovechando la capacidad máxima de cada turno (t) y máquina (m). A continuación, la descripción de esta lógica:

8.1 Máquina 1: ¿La máquina m en el turno t (t de menor a mayor), tiene capacidad para el artículo (γ)?

Sí: Asignar el artículo al turno “ t ”

No: validar capacidad del turno “ $t+1$ ” para el artículo (γ)

8.2 Máquina 2: ¿El artículo (γ) es mayor a la capacidad de la maquina 2?

Sí: Asignar nuevo turno “ t ” a la máquina 1

No: Asignar a turno “ t ” de máquina 2 teniendo en cuenta la lógica de capacidad por turno “ t ”.

8.3 Máquina 3: ¿El artículo (γ) es mayor a la capacidad de la maquina 3?

Sí: Evaluar restricción 1 (¿el tamaño del trabajo (γ) es menor o igual a la capacidad de la máquina 2?)

Sí: Asignar a turno de máquina 2 teniendo en cuenta la lógica de capacidad por turno “ t ”.

No: Asignar nuevo turno M_t a la máquina 1 teniendo en cuenta la lógica de capacidad por turno “ t ”.

No: Asignar a máquina 3 teniendo en cuenta la lógica de capacidad por turno “t”.

Capacidades Max	Máquina	T_1	T_2	T_3	T_4	T_5	T_6
960 - 1920	Autoclave 1	1	2_0	4;3	6_0	5	7
160 - 960	Autoclave 2	8;9	10	11			
0 - 160	Autoclave 3	6_1;2_1	12	13			

Figura 14. Ejemplo de asignación de artículos a máquinas y turnos
Fuente: Elaboración propia

9. Una vez la totalidad de artículos (γ) sean distribuidos en la matriz de máquinas / turnos se da por cerrada la generación de la subpoblación (β) y se continuará con la subpoblación ($\beta+1$) hasta la cantidad de subpoblaciones (β) parametrizadas.

5.4.2.2 Fase 2. Algoritmo genético

1. Una vez generada la subpoblación (β) reflejada en la asignación de artículos (γ) a máquinas (m) y turnos (t), se construye un cromosoma compuesto por genes que representan los turnos de las máquinas de tintorería, los cuales se componen de los artículos ordenados en secuencia de procesamiento para las etapas de magueba en adelante.

La figura 15 representa una subpoblación y la figura 16 representa un cromosoma generado a partir de la subpoblación

ID subpoblación	Capacidades Max	Máquina	T_1	T_2	T_3	T_4	T_5	T_6
1	960 - 1920	Autoclave 1	1	2_0	4;3	6_0	5	7
	160 - 960	Autoclave 2	8;9	10	11			
	0 - 160	Autoclave 3	6_1;2_1	12	13			

Figura 15. Representación de una subpoblación
Fuente: Elaboración propia

Turno (T)	T_1	T_2	T_3	T_4	T_5	T_6
Secuencia (y)	1; 6_1; 8; 9; 2_1	12; 2_0; 10	4; 13; 11; 3	6_0	5	7

Figura 16. Representación de un cromosoma
Fuente: Elaboración propia

Cada cromosoma se compone de un grupo de genes conformados por los artículos (y) ordenados de forma aleatoria y asignados a cada turno.

2. Generar una cantidad (parametrizable) de cromosomas (μ) por cada subpoblación (β), a través de la figura 17 se puede ver un ejemplo de generación de tres subpoblaciones con tres cromosomas parametrizados.

ID subpoblación	Capacidades Max	Máquina	T_1	T_2	T_3	T_5	T_5	T_6
1	960 - 1920	Autoclave 1	1	2_0	4; 3	6_0	5	7
	160 - 960	Autoclave 2	8; 9	10	11			
	0 - 160	Autoclave 3	6_1; 2_1	12	13			
2	960 - 1920	Autoclave 1	4	2_0	1	5	6_0	7
	160 - 960	Autoclave 2	9	11	8; 3	10		
	0 - 160	Autoclave 3	13	6_1; 2_1	12			
3	960 - 1920	Autoclave 1	1;6_1	4	6_0	2_0	5	7
	160 - 960	Autoclave 2	10	11; 3	9	8		
	0 - 160	Autoclave 3	2_1	12	13			

ID subpoblación	ID Cromosoma	T_1	T_2	T_3	T_4	T_5	T_6
1	1	1; 6_1; 8; 9; 2_1	12; 2_0; 10	4; 13; 11; 3	6_0	5	7
1	2	1; 2_1; 8; 6_1; 9	12; 2_0; 10	4; 13; 11; 3	6_0	5	7
1	3	2_1; 6_1; 8; 9; 1	12; 2_0; 10	4; 13; 11; 3	6_0	5	7
2	1	13; 4; 9	2_0; 6_1; 11; 2_1	1; 12; 8; 3	10; 5	7	6_0
2	2	9; 13; 4	2_1; 11; 2_0; 6_1	3; 1; 12; 8	10; 5	7	6_0
2	3	13; 9; 4	6_1; 2_0; 11; 2_1	12; 8; 3; 1	5; 10	7	6_0
3	1	10; 2_1; 1; 6_1	12; 3; 4; 11	13; 6_0; 9	8; 2_0	5	7
3	2	6_1; 2_1; 1; 10;	11; 3; 4; 12	9; 6_0; 13	2_0; 8	5	7
3	3	2_1; 6_1; 10; 1	3; 12; 4; 11	6_0; 13; 9	2_0; 8	5	7

Figura 17. Ejemplo de generación de subpoblaciones y cromosomas
Fuente: Elaboración propia

3. Medir el makespan y trabajos tardíos a cada uno de los cromosomas (μ) generados, las funciones objetivo son tomadas como las funciones fitness planteadas para la evaluación de los individuos.

Identificadores		Cromosomas						FOB	
ID subpoblac	ID Cromosoma	T_1	T_2	T_3	T_4	T_5	T_6	Makespan	Q Trabajos T.
1	1	1; 6_1; 8; 9; 2_1	12; 2_0; 10	4; 13; 11; 3	6_0	5	7	2329	4
1	2	1; 2_1; 8; 6_1; 9	12; 2_0; 10	4; 13; 11; 3	6_0	5	7	2325	2
1	3	2_1; 6_1; 8; 9; 1	12; 2_0; 10	4; 13; 11; 3	6_0	5	7	3475	2
2	1	13; 4; 9	2_0; 6_1; 11; 2_1	1; 12; 8; 3	10; 5	7	6_0	1490	4
2	2	9; 13; 4	2_1; 11; 2_0; 6_1	3; 1; 12; 8	10; 5	7	6_0	2066	0
2	3	13; 9; 4	6_1; 2_0; 11; 2_1	12; 8; 3; 1	5; 10	7	6_0	2167	0
3	1	10; 2_1; 1; 6_1	12; 3; 4; 11	13; 6_0; 9	8; 2_0	5	7	2561	3
3	2	6_1; 2_1; 1; 10;	11; 3; 4; 12	9; 6_0; 13	2_0; 8	5	7	3295	0
3	3	2_1; 6_1; 10; 1	3; 12; 4; 11	6_0; 13; 9	2_0; 8	5	7	3383	3

Figura 18. Medición de fitness a cada individuo

Fuente: Elaboración propia

4. Una vez ejecutada la iteración (α) 0, que genera las subpoblaciones (β) y sus correspondientes cromosomas (μ), la selección del mejor cromosoma (μ), deberá basarse en los resultados del makespan o trabajos tardíos por subpoblación (β), según la medición seleccionada en cada iteración (α), la cual, deberá ser de forma intercalada.

- Iteración 0: Makespan
- Iteración 1: Trabajos tardíos
- Iteración 2: Makespan

En la figura 19 como ejemplo se seleccionan los mejores padres de la subpoblación dos según medición de makespan.

Identificadores		Cromosomas						FOB	
ID subpoblac	ID Cromosoma	T_1	T_2	T_3	T_4	T_5	T_6	Makespan	Q Trabajos T.
1	1	1; 6_1; 8; 9; 2_1	12; 2_0; 10	4; 13; 11; 3	6_0	5	7	2329	4
1	2	1; 2_1; 8; 6_1; 9	12; 2_0; 10	4; 13; 11; 3	6_0	5	7	2325	2
1	3	2_1; 6_1; 8; 9; 1	12; 2_0; 10	4; 13; 11; 3	6_0	5	7	3475	2
2	1	13; 4; 9	2_0; 6_1; 11; 2_1	1; 12; 8; 3	10; 5	7	6_0	1490	4
2	2	9; 13; 4	2_1; 11; 2_0; 6_1	3; 1; 12; 8	10; 5	7	6_0	2066	0
2	3	13; 9; 4	6_1; 2_0; 11; 2_1	12; 8; 3; 1	5; 10	7	6_0	2167	0
3	1	10; 2_1; 1; 6_1	12; 3; 4; 11	13; 6_0; 9	8; 2_0	5	7	2561	3
3	2	6_1; 2_1; 1; 10;	11; 3; 4; 12	9; 6_0; 13	2_0; 8	5	7	3295	0
3	3	2_1; 6_1; 10; 1	3; 12; 4; 11	6_0; 13; 9	2_0; 8	5	7	3383	3

Figura 19. Selección de mejores padres según makespan

Fuente: Elaboración propia

5. Una vez dada la selección de los mejores cromosomas (μ) según makespan de cada subpoblación (β), se debe llevar a cabo el proceso de cruce uniforme, el cual deberá realizarse intercambiando la posición de los genes.

Padre 1

T_1	T_2	T_3	T_4	T_5	T_6
13; 4; 9	2_0; 6_1; 11; 2_1	1; 12; 8; 3	10; 5	7	6_0

Padre 2

T_1	T_2	T_3	T_4	T_5	T_6
9; 13; 4	2_1; 11; 2_0; 6_1	3; 1; 12; 8	10; 5	7	6_0

Hijo 1

T_1	T_2	T_3	T_4	T_5	T_6
13; 4; 9	2_1; 11; 2_0; 6_1	1; 12; 8; 3	10; 5	7	6_0

Hijo 2

T_1	T_2	T_3	T_4	T_5	T_6
9; 13; 4	2_0; 6_1; 11; 2_1	3; 1; 12; 8	10; 5	7	6_0

6. Una vez cruzados los cromosomas, se realiza la medición del fitness

Hijo 1

T_1	T_2	T_3	T_4	T_5	T_6
13; 4; 9	2_1; 11; 2_0; 6_1	1; 12; 8; 3	10; 5	7	6_0

Makespan	Q Trabajos T.
2025	1

Hijo 2

T_1	T_2	T_3	T_4	T_5	T_6
9; 13; 4	2_0; 6_1; 11; 2_1	3; 1; 12; 8	10; 5	7	6_0

Makespan	Q Trabajos T.
2320	0

7. Al generar los hijos, determinar si se realiza el proceso de mutación de acuerdo a los criterios descritos a continuación:

- El % de probabilidad de mutación (w) deberá ser parametrizable
- El número aleatorio para determinar si se realiza la mutación, será realizado para cada uno de los hijos

Hijo 1

- **% Mutación parametrizado:** 10% (resultados de 1 a 10 en número aleatorio de 1 a 100)
- **Número aleatorio:** 86

- **Resultado:** No mutación

Hijo 2

- **% Mutación parametrizado:** 10% (resultados de 1 a 10 en número aleatorio de 1 a 100)
- **Número aleatorio:** 5
- **Resultado:** Cromosoma a mutar

8. En caso de que se cumpla el porcentaje para el proceso de mutación, se deberán seleccionar dos genes de forma aleatoria, e intercambiar sus posiciones.

Antes:

T_1	T_2	T_3	T_4	T_5	T_6
9; 13; 4	2_0; 6_1; 11; 2_1	3; 1; 12; 8	10; 5	7	6_0

Makespan	Q Trabajos T.
2320	0

Mutado:

T_1	T_2	T_3	T_4	T_5	T_6
9; 13; 4	2_0; 6_1; 11; 2_1	7	10; 5	3; 1; 12; 8	6_0

Makespan	Q Trabajos T.
2410	1

9. Los cromosomas (μ) cruzados y mutados (si aplica), deberán reemplazar a los cromosomas (μ) con mayor makespan generados en la iteración 0 para generar la nueva población; este método es denominado reemplazo por estado estacionario, teniendo en cuenta los siguientes puntos:

- Si los 2 cromosomas (μ) hijos no mutaron, pasan a reemplazar los 2 peores de la iteración 0
- Si de los 2 cromosomas (μ) hijos mutó 1, pasan a reemplazar los 3 peores de la iteración 0
- Si los 2 cromosomas (μ) mutan, se generarán 4 cromosomas (μ) que reemplazarán a los 4 peores
- Este caso se repetirá en cada iteración, y se escogerá dependiendo de la medición intercalada de cada una.

10. El proceso de cruce y mutación deberá realizarse para cada conjunto de cromosomas (μ) pertenecientes a cada una de las subpoblaciones (β) generadas.
11. Una vez integrados los cromosomas (μ) (hijos) en cada subpoblación (β), se deberá llevar a cabo la iteración $\alpha+1$, en donde la medición del fitness será intercalada (diferente a la iteración α).
12. La cantidad de iteraciones (α) deberá ser parametrizable.
13. Para cada cromosoma (μ), realizar el cálculo de los KPI's correspondientes a makespan y trabajos tardíos normalizándolos con la misma escala de medida (%) y promediarlos

Indicador makespan: (Valor makespan de cromosoma μ / Valor mínimo makespan de la totalidad de cromosomas μ) * 100

Indicador trabajos tardíos: Igualar el menor valor del indicador de makespan con la menor cantidad de trabajos tardíos y el mayor valor del makespan con la mayor cantidad de trabajos tardíos y calcular un porcentaje equivalente a estos valores.

Identificadores		Cromosomas						FOB		KPI'S		
ID subpoblac	ID Cromosoma	T_1	T_2	T_3	T_4	T_5	T_6	Makespan	Q Trabajos T.	Makespan	Q Trabajos T.	KPI Norm.
1	1	1; 6 1; 8; 9; 2 1	12; 2 0; 10	4; 13; 11; 3	6 0	7	5	1829	2	104%	151%	128%
1	2	1; 2 1; 8; 6 1; 9	12; 2 0; 10	11; 3; 4; 13	6 0	5	7	2138	0	121%	100%	111%
1	3	2 1; 6 1; 8; 9; 1	12; 2 0; 10	13; 4; 3; 11	5	6 0	7	3053	1	173%	126%	150%
2	1	13; 4; 9	2 0; 6 1; 11; 2 1	1; 12; 8; 3	10; 5	7	6 0	1761	3	100%	177%	139%
2	2	9; 13; 4	3; 1; 12; 8	1; 11; 2 0; 6	10; 5	6 0	7	2579	1	146%	126%	136%
2	3	13; 9; 4	6 1; 2 0; 11; 2 1	12; 8; 3; 1	5; 10	7	6 0	1830	0	104%	100%	102%
2	4	13; 9; 4	11; 6 1; 2 0; 2 1	3; 1; 12; 8	6 0	7	5; 10	1920	2	109%	151%	130%
3	1	10; 2 1; 1; 6 1	12; 3; 4; 11	13; 6 0; 9	8; 2 0	5	7	1868	0	106%	100%	103%
3	2	11; 3; 4; 12	6 1; 2 1; 1; 10;	9; 6 0; 13	2 0; 8	7	5	3120	0	177%	100%	139%
3	3	2 1; 6 1; 10; 1	6 0; 13; 9	3; 12; 4; 11	8; 2 0	7	5	3042	1	173%	126%	149%

14. Seleccionar el menor valor del KPI total como mejor resultado

Identificadores		Cromosomas						FOB		KPI'S		
ID subpoblac	ID Cromosoma	T_1	T_2	T_3	T_4	T_5	T_6	Makespan	Q Trabajos T.	Makespan	Q Trabajos T.	KPI Norm.
1	1	1; 6 1; 8; 9; 2 1	12; 2 0; 10	4; 13; 11; 3	6 0	7	5	1829	2	104%	151%	128%
1	2	1; 2 1; 8; 6 1; 9	12; 2 0; 10	11; 3; 4; 13	6 0	5	7	2138	0	121%	100%	111%
1	3	2 1; 6 1; 8; 9; 1	12; 2 0; 10	13; 4; 3; 11	5	6 0	7	3053	1	173%	126%	150%
2	1	13; 4; 9	2 0; 6 1; 11; 2 1	1; 12; 8; 3	10; 5	7	6 0	1761	3	100%	177%	139%
2	2	9; 13; 4	3; 1; 12; 8	1; 11; 2 0; 6	10; 5	6 0	7	2579	1	146%	126%	136%
2	3	13; 9; 4	6 1; 2 0; 11; 2 1	12; 8; 3; 1	5; 10	7	6 0	1830	0	104%	100%	102%
2	4	13; 9; 4	11; 6 1; 2 0; 2 1	3; 1; 12; 8	6 0	7	5; 10	1920	2	109%	151%	130%
3	1	10; 2 1; 1; 6 1	12; 3; 4; 11	13; 6 0; 9	8; 2 0	5	7	1868	0	106%	100%	103%
3	2	11; 3; 4; 12	6 1; 2 1; 1; 10;	9; 6 0; 13	2 0; 8	7	5	3120	0	177%	100%	139%
3	3	2 1; 6 1; 10; 1	6 0; 13; 9	3; 12; 4; 11	8; 2 0	7	5	3042	1	173%	126%	149%

5.5 Pseudocódigo de la metodología propuesta

Estructura cromosoma: codificación numérica que indica el número de artículo (γ) a programar
 Parametrizar α (iteraciones), β (subpoblaciones), μ (cromosomas) y W (% de probabilidad de mutación)

Conjuntos: γ (artículos), t (turnos), m (máquinas), P_γ (agrupación de artículos por tipo de producto y color)

Variables: T_γ (conjunto de turnos de artículos)

Crear conjuntos de artículo " ρ_γ " según color (η) y tipo (λ) de artículo (γ)

```

While  $\sum \beta > 0$ 
  While  $\sum \rho_\gamma > 0$ 
    Ordenar aleatoriamente los ( $\gamma$ ) en cada ( $\rho_\gamma$ )
    While  $\gamma \in \rho_\gamma > 0$ 
      Evaluar capacidad y prioridad de máquina ( $m$ )
      Evaluar capacidad de turno ( $t$ )
      IF ( $\gamma$ ) < Capacidad de máquina ( $m$ ) Then
        IF ( $\gamma$ ) < Capacidad del turno ( $t$ ) Then
          Asignar artículo ( $\gamma$ ) a turno ( $t$ )
        Else Evaluar turno ( $t + 1$ )
      Else Evaluar máquina ( $m+1$ )
     $\gamma \in \rho_\gamma - 1$ 
   $\rho_\gamma - 1$ 
  Agrupar ( $\gamma$ ) asignado a cada ( $t$ ) y conformar el conjunto ( $T_\gamma$ )

  While  $\sum \mu > 0$ 
    Generar ( $\mu$ ) mediante orden aleatorio de ( $\gamma$ ) correspondientes a cada conjunto ( $T_\gamma$ )
     $\mu - 1$ 
   $\beta - 1$ 
  
```

Figura 20. Pseudocódigo de asignación de máquinas para proceso tinturado
 Fuente: Elaboración propia

Calcular fitness de cada cromosoma:
 Calcular C_{max} (makespan): Tiempo máximo de terminación de todos los (γ) de cada (μ)
 Calcular trabajos tardíos $U(\gamma)$: Cantidad de artículos (γ) con due date (F_γ) < tiempo de terminación de los artículos Q_γ de cada (μ)
For $\alpha=1$ to α
 For $\beta=1$ to β
 Seleccionar de los padres con mejor fitness
 If $\alpha=2n+1$
 Then: Seleccionar (μ) padres con menor makespan
 Else: Seleccionar (μ) padres con menor cantidad de trabajos tardíos
 End If
 Operación de cruce uniforme
 Generar (μ) hijos
 Calcular random (0-100%)
 If random $\leq W$
 Then: Mutar 2 posiciones aleatorias de (μ) hijo
 Else: Omitir proceso de mutación
 End If
 Calcular fitness de (μ) hijos y (μ) hijos mutados
 Comparar el fitness de los (μ) hijos y (μ) hijos mutados, con los (μ) de las subpoblaciones (β)
 If Fitness (μ) hijos y (μ) hijos mutados < fitness (μ) de la subpoblación (β)
 Then: Reemplazar (μ) de menor fitness
 Else: Omitir proceso de reemplazo
 End If
 End for
End for
 Calcular KPI's de fitness para cada (μ) en cada (β)
 Seleccionar (μ) con mejor KPI

Figura 21. Pseudocódigo del algoritmo genético
 Fuente: Elaboración propia

6. EXPERIMENTACIÓN Y RESULTADOS

Se va a ejecutar la prueba del modelo propuesto utilizando datos históricos de la empresa objeto de estudio con el fin de comparar los resultados reales de makespan y cantidad de trabajos tardíos vs. resultados arrojados por el modelo. En estas pruebas se toman pedidos de venta que contienen productos de diferente tipo (felpa o gancho), tamaño y color.

El algoritmo diseñado ha sido desarrollado e implementado para el uso de la empresa a través del lenguaje de programación Python versión 3.8.5 y las pruebas fueron

ejecutadas en un equipo portátil con procesador Intel Core i5 g8 y capacidad de memoria RAM de 8 GB.

6.1 Análisis datos de entrada

Los datos de entrada del sistema se generan en un archivo desde SAP Business One (Tabla 1) que contiene información de las órdenes de venta (número de orden, código de producto, cantidad, tipo de producto, color, fecha de entrega prometida al cliente, fecha contabilización de la orden y un identificador único para cada producto), este archivo será la información que el algoritmo procesa para la generación de la mejor secuencia de programación.

Numero OV	Articulo	Cantidad (mts)	Tipo producto	Color	fecha entreg	fecha contabilizaci	Identificador
OV16296	223200345025	1350	GANCHO	345	17/02/2020	13/01/2020	2
OV16296	222200345025	575	FELPA	345	17/02/2020	13/01/2020	3
OV16441	222200345025	225	FELPA	345	17/02/2020	16/01/2020	43
OV16580	223200792050	100	GANCHO	792	17/02/2020	23/01/2020	70
OV16580	223200792100	100	GANCHO	792	17/02/2020	23/01/2020	71
OV16621	222200082100	800	FELPA	82	17/02/2020	24/01/2020	97
OV16621	223200082025	200	GANCHO	82	17/02/2020	24/01/2020	98
OV16804	222200082016	100	FELPA	82	17/02/2020	31/01/2020	114
OV16804	223200190016	150	GANCHO	190	17/02/2020	31/01/2020	117
OV16968	222200082016	50	FELPA	82	17/02/2020	7/02/2020	137
OV16968	223200190016	50	GANCHO	190	17/02/2020	7/02/2020	140
OV17112	223200190016	250	GANCHO	190	17/02/2020	13/02/2020	159
OV17156	223200082025	62	GANCHO	82	17/02/2020	14/02/2020	179
OV17344	223200615100	5500	GANCHO	615	17/02/2020	24/02/2020	207
OV17344	223200615100	5500	GANCHO	615	17/02/2020	24/02/2020	207
OV17368	222200082020	160	FELPA	82	17/02/2020	24/02/2020	210
OV18994	223200615025	125	GANCHO	615	17/02/2020	5/06/2020	242
OV19091	222200615025	125	FELPA	615	17/02/2020	9/06/2020	263
OV19091	223200615025	125	GANCHO	615	17/02/2020	9/06/2020	264
OV19341	222200615025	6	FELPA	615	17/02/2020	17/06/2020	278
OV19341	223200615025	6	GANCHO	615	17/02/2020	17/06/2020	279
OV19417	222200615025	6	FELPA	615	17/02/2020	23/06/2020	285
OV19417	223200615025	6	GANCHO	615	17/02/2020	23/06/2020	286
OV19467	222200615025	69	FELPA	615	17/02/2020	23/06/2020	293
OV19467	223200615025	69	GANCHO	615	17/02/2020	23/06/2020	294
OV19494	222200615025	125	FELPA	615	17/02/2020	24/06/2020	298
OV19494	222200615025	125	FELPA	615	17/02/2020	24/06/2020	298
OV19494	223200615025	125	GANCHO	615	17/02/2020	24/06/2020	299

Tabla 1. Archivo de entrada

Fuente: Elaboración propia a partir de datos de la empresa

Se realiza un análisis de correlación a los datos de tiempos (makespan) entregados por la empresa respecto a la cantidad de metros de los artículos a programar en cada una de las fechas, la correlación de estas dos variables es $R^2 = 0,72$; valor que puede verse afectado por la precisión en la que se registra la información en el ERP.

Fecha programación	Cantidad de trabajos	Cantidad de metros	Makespan real (min)
enero 12 2020	5	431	523
enero 15 2020	10	9712	5940
enero 22 2020	23	4624	1789
enero 27 2020	46	13871	6380
febrero 03 2020	28	16084	5986
febrero 07 2020	10	2175	1534
febrero 14 2020	11	5200	6238
febrero 17 2020	8	864	779
febrero 24 2020	12	5350	2359
junio 04 2020	12	960	766

Tabla 2. Cantidad de metros procesados vs. makespan real
Fuente: Elaboración propia partir de datos de la empresa

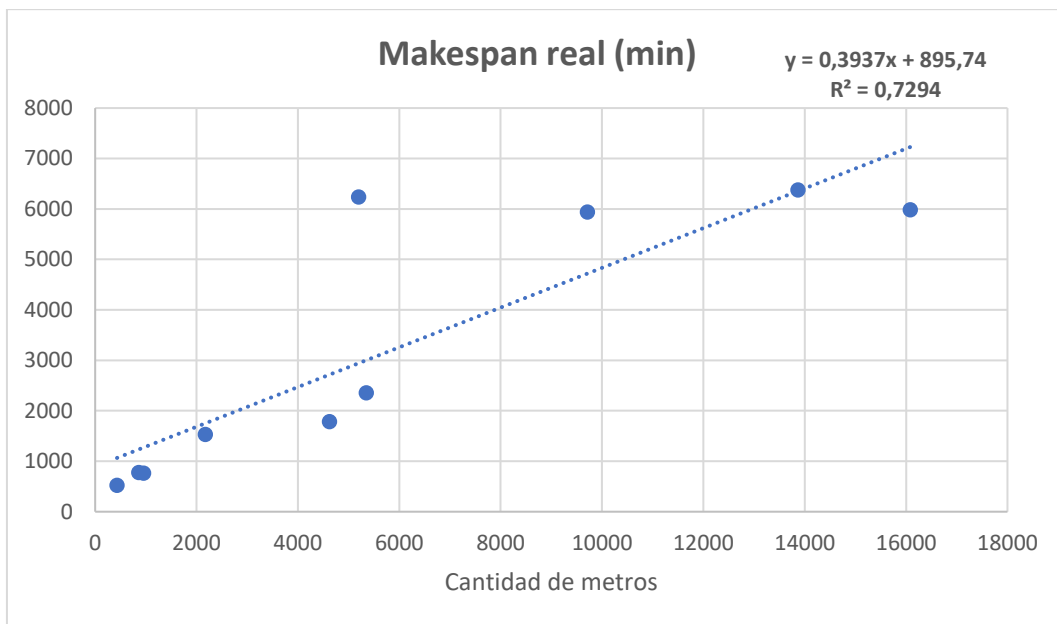


Figura 22. Gráfico análisis de correlación makespan real
Fuente: Elaboración propia

6.2 Parámetros del modelo

Para el funcionamiento del modelo es necesario el ingreso de ciertos parámetros con información base. (tabla binaria de proceso, matriz de tiempos de proceso) Iteraciones del modelo o del AG, cantidad de subpoblaciones, cantidad de cromosomas, porcentaje de mutación, tiempo de parada.

6.2.1 Tabla que relaciona los tipos de producto que se fabrican en la línea textil en la fase de acabados y el paso de cada uno por las máquinas existentes (Tabla 3), si existe un 1 el producto puede fabricarse en esa máquina, si se visualiza un 0 el trabajo no debe procesarse en esa máquina.

Máquinas	Tipo de producto	
	Krik Felpa	Krik Gancho
Autoclave 1	1	1
Autoclave 2	1	1
Autoclave 3	1	1
Mageba	1	1
Cepillo breiten	1	0
Termofijadora 1	1	1
Termofijadora 2	0	1
Corte monofilamento	0	1
Corte ultrasonido	1	1
Enrollado	1	1
Empaque	1	1

Tabla 3. Paso de cada tipo de producto por máquinas
Fuente: Elaboración propia a partir de datos de la empresa

6.2.2 Matriz de tiempos que indica el tiempo de procesamiento del producto en cada uno de los procesos, cuyo tiempo de alistamiento y factor de eficiencia de las máquinas se encuentran inmersos en cada uno de los tiempos estimados en minutos por metro; los productos que no son procesados en determinadas etapas se reflejan con un tiempo igual a 0 en la tabla 4. Tiempos de procesamiento por etapa.

Proceso	Etapas	Máquinas	Gancho	Felpa	Tiempo de procesamiento
1	Teñido Tuntún	Máquina Teñido 3	60	60	Seg/mt
	Teñido Autoclave 1	Máquina Teñido 2	10	10	Seg/mt
	Teñido Autoclave 2	Máquina Teñido 1	5	5	Seg/mt
2	Mageba	Mageba	12	12	Seg/mt
3	Cepillo Breiten	Cepillo	0	13	Seg/mt
4	Acabado	Termofijadora	7,9	7,9	Seg/mt
5	Segunda Pasada				
	Acabado	Termofijadora 2	6	0	Seg/mt
6	Corte Monofilamento	Corte Monofilamento	19	0	Seg/mt
S7	Corte Ultrasonido	Corte Ultrasonido	12	12	Seg/mt
8	Enrollado	Enrollado	3,3	3,3	Seg/mt
9	Empaque	Empaque	3,3	3,3	Seg/mt

Tabla 4. Tiempos de procesamiento por etapa

Fuente: Elaboración propia a partir de datos de la empresa

6.2.3 El listado de parámetros para la ejecución de las pruebas deberá contener las variables indicadas en la siguiente tabla:

Parámetro	Descripción
Cantidad de subpoblaciones	Número de grupos de cromosomas contruidos a partir de la asignación a máquinas de tintorería
Cantidad de cromosomas	Orden aleatorio de trabajos, basados en la asignación de turnos por máquina en cada subpoblación
% Mutación	Probabilidad de mutación de los cromosomas, lo cual aplica solamente para cromosomas de la misma subpoblación
Iteraciones AG	Cantidad de iteraciones del algoritmo genético, desde el proceso de cruce y mutación hasta el reemplazo de cromosomas y cálculo de fitness
Fecha inicio de producción	Fecha en la cual se inicia el proceso operativo de producción; base para el cálculo de tiempos de proceso

Tabla 5. Parámetros del sistema

Fuente: Elaboración propia

6.3 Pruebas reales

El procedimiento y software desarrollado se aplicó a la producción real de Kromia S.A.S con 165 trabajos correspondientes a 10 programas de producción pertenecientes a los meses de enero, febrero y junio con un horizonte de fechas de entrega de 14 días (ANS actual de la empresa).

La figura 23 indica el funcionamiento del programa desarrollado, las figuras 24 y 25 representan respectivamente los parámetros que se deben indicar para que el sistema inicie el procesamiento de datos y la configuración del mejor cromosoma donde se puede observar información relacionada con el turno y máquina de tintorería asignada en la fase 1.

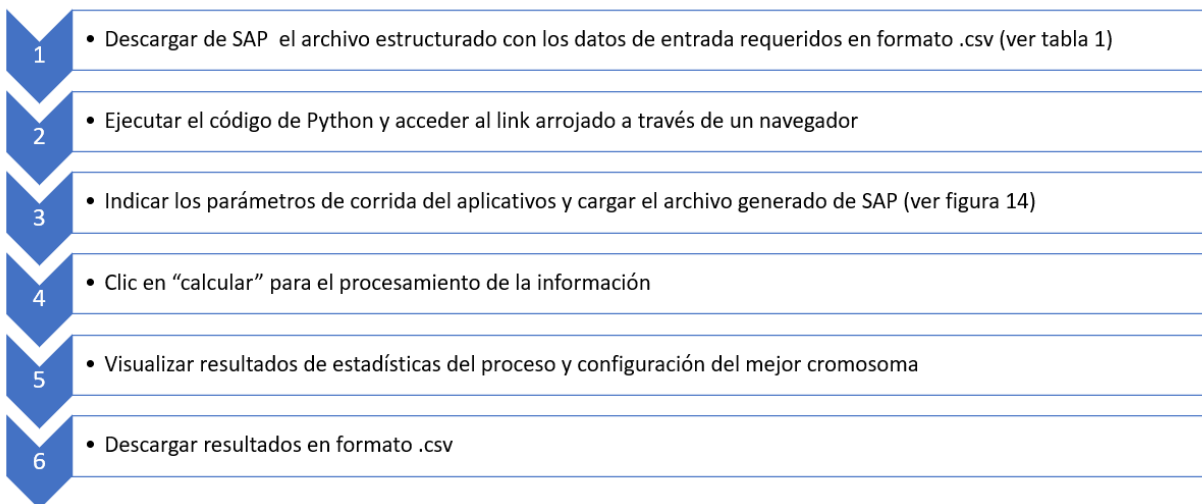


Figura 23. Paso a paso funcionamiento del sistema

Fuente: Elaboración propia

Estimar

Iteraciones:

Subpoblaciones:

Cromosomas:

Mutación:

Fecha inicio:

Ningún archivo seleccionado

Figura 24. Interfaz gráfica de usuario

Fuente: Elaboración propia

Configuración mejor cromosoma Descargar

Mostrar 100 registros Buscar:

#	index	id job	tipo	cantidad	entrega	turno	maquina elegida	Tintoreria	Mageba	Capillo	Termofijadora	Termofijadora2	Corte Monofilamento	Corte Ultrasonido	Enrollado	Empaque	tiempo total	dias total	fecha estimada entrega	trabajo tardio
0	219	219.0	FLPA	25	2020-02-05	1.0	m1	180.0	185.0	192.0	196.0	NaN	NaN	201.0	203.0	205.0	205.0	1	2020-02-03	False
1	15	15.0	FLPA	100	2020-02-05	1.0	m3	180.0	205.0	230.0	245.0	NaN	NaN	265.0	271.0	277.0	277.0	1	2020-02-03	False
2	144	144.0	FLPA	450	2020-02-05	1.0	m1	180.0	295.0	408.0	473.0	NaN	NaN	563.0	588.0	613.0	613.0	2	2020-02-04	False
3	101	101.0	FLPA	212	2020-02-05	1.0	m1	180.0	338.0	461.0	504.0	NaN	NaN	606.0	618.0	630.0	630.0	2	2020-02-04	False
4	79	79.0	FLPA	100	2020-02-05	1.0	m1	180.0	358.0	486.0	519.0	NaN	NaN	626.0	632.0	638.0	638.0	2	2020-02-04	False
5	125	125.0	FLPA	625	2020-02-05	1.0	m1	180.0	483.0	643.0	733.0	NaN	NaN	858.0	893.0	928.0	928.0	2	2020-02-04	False
6	164	164.0	FLPA	500	2020-02-05	1.0	m1	180.0	583.0	768.0	840.0	NaN	NaN	958.0	986.0	1014.0	1014.0	2	2020-02-04	False
7	135	135.0	FLPA	250	2020-02-05	1.0	m2	180.0	633.0	831.0	876.0	NaN	NaN	1008.0	1022.0	1036.0	1036.0	2	2020-02-04	False
8	198	198.0	GANCHO	237	2020-02-05	2.0	m1	360.0	681.0	NaN	910.0	934.0	1013.0	1061.0	1075.0	1089.0	1089.0	3	2020-02-05	False
9	16	16.0	FLPA	100	2020-02-05	2.0	m2	360.0	701.0	856.0	925.0	NaN	NaN	1081.0	1087.0	1095.0	1095.0	3	2020-02-05	False
10	167	167.0	FLPA	150	2020-02-05	2.0	m3	360.0	731.0	894.0	947.0	NaN	NaN	1111.0	1120.0	1129.0	1129.0	3	2020-02-05	False
11	17	17.0	FLPA	200	2020-02-05	2.0	m2	360.0	771.0	944.0	976.0	NaN	NaN	1151.0	1163.0	1175.0	1175.0	3	2020-02-05	False
12	212	212.0	GANCHO	250	2020-02-05	2.0	m1	360.0	821.0	NaN	1012.0	1037.0	1121.0	1201.0	1215.0	1229.0	1229.0	3	2020-02-05	False
13	165	165.0	GANCHO	500	2020-02-05	2.0	m1	360.0	921.0	NaN	1084.0	1134.0	1301.0	1401.0	1429.0	1457.0	1457.0	3	2020-02-05	False
14	172	172.0	FLPA	150	2020-02-05	3.0	m2	540.0	951.0	989.0	1106.0	NaN	NaN	1431.0	1440.0	1466.0	1466.0	3	2020-02-05	False
15	145	145.0	FLPA	25	2020-02-05	3.0	m1	540.0	956.0	996.0	1110.0	NaN	NaN	1436.0	1442.0	1468.0	1468.0	3	2020-02-05	False
16	20	20.0	FLPA	100	2020-02-05	3.0	m1	540.0	976.0	1021.0	1125.0	NaN	NaN	1456.0	1462.0	1474.0	1474.0	3	2020-02-05	False
17	45	45.0	FLPA	100	2020-02-05	3.0	m1	540.0	996.0	1046.0	1140.0	NaN	NaN	1476.0	1482.0	1488.0	1488.0	3	2020-02-05	False
18	23	23.0	FLPA	100	2020-02-05	3.0	m2	540.0	1016.0	1071.0	1155.0	NaN	NaN	1496.0	1502.0	1508.0	1508.0	3	2020-02-05	False
19	8	8.0	FLPA	100	2020-02-05	3.0	m3	540.0	1036.0	1096.0	1170.0	NaN	NaN	1516.0	1522.0	1528.0	1528.0	3	2020-02-05	False
20	98	98.0	GANCHO	200	2020-02-05	4.0	m1	720.0	1076.0	NaN	1199.0	1219.0	1368.0	1556.0	1568.0	1580.0	1580.0	3	2020-02-05	False
21	41	41.0	FLPA	100	2020-02-05	4.0	m2	720.0	1096.0	1121.0	1214.0	NaN	NaN	1576.0	1582.0	1588.0	1588.0	3	2020-02-05	False
22	21	21.0	FLPA	50	2020-02-05	4.0	m3	720.0	1106.0	1134.0	1222.0	NaN	NaN	1586.0	1589.0	1592.0	1592.0	3	2020-02-05	False

Mostrando registros del 1 al 23 de un total de 23 registros Anterior 1 Siguiete

Texto Footer

Figura 25. Interfaz gráfica configuración mejor cromosoma
Fuente: Elaboración propia

La tabla 7 muestra los resultados obtenidos de las mejores soluciones, la media del makespan, la desviación estándar muestral y la desviación porcentual en las simulaciones ejecutadas con cada una de las combinaciones de parámetros para los $\gamma=165$ trabajos. Teniendo en cuenta que una desviación estándar baja indica que la mayoría de los datos de una muestra tienden a estar agrupados cerca de su media, mientras que una desviación estándar alta indica que los datos se extienden sobre un rango de valores más amplio, se analiza que las desviaciones altas presentadas como resultado generan en su mayoría los mejores fitness, es decir, que las mejores soluciones se encontraron realizando una exploración más amplia del espacio de búsqueda

Las unidades de tiempo del makespan, en relación con la mejor solución y la medida real de la empresa en cada uno de los escenarios está dada en minutos.

Los tiempos computacionales del algoritmo varían de acuerdo con las configuraciones y la cantidad de trabajos (en un rango de 1 a 50), para aquellas que manejan 10 iteraciones y 2 subpoblaciones el tiempo requerido por el algoritmo para encontrar una solución, oscila entre 19 segundos y 2 minutos; al ejecutar los escenarios de 20 iteraciones y 10 subpoblaciones el tiempo aumenta y varía en un rango de 3 a 36 minutos dado que la búsqueda de soluciones se vuelve más rigurosa.

En la tabla 8 se visualizan los resultados de la mejor combinación escogida para cada programa de producción y el tipo de cromosoma que arrojó la mejor solución y se comparan con los datos reales proporcionados por la empresa. Se evidencia una mejora promedio de 911 minutos (78%) respecto al makespan real que tuvieron los 10 programas de producción y en ningún escenario ejecutado se presentan trabajos tardíos. En esta tabla es posible evidenciar que las mejores soluciones para el 80% de los escenarios se dieron con una configuración de 10 subpoblaciones, lo que quiere decir que este parámetro es el que mayor influencia tiene sobre una buena generación de resultados esto con base en los análisis de las desviaciones estándar presentadas (mayor exploración de resultados), adicional, los cromosomas que se seleccionaron por tener un mejor fitness fueron en su mayoría de tipo hijo, por el contrario, en 53 resultados de las 80 pruebas ejecutadas (66%), los cromosomas con menor calificación fueron de tipo mutación.

Se realizó un análisis de correlación para determinar la relación entre la cantidad de metros de los artículos y el makespan generado por el sistema diseñado, este análisis estadístico arrojó como resultado un coeficiente de correlación $R^2= 0,97$ lo que quiere decir que existe un alto grado de asociación entre las variables.

Fecha programación	Cantidad de trabajos	Cantidad de metros	Makespan pruebas (min)
enero 12 2020	5	431	465
enero 15 2020	10	9712	3758
enero 22 2020	23	4624	1597
enero 27 2020	46	13871	4456
febrero 03 2020	28	16084	5835
febrero 07 2020	10	2175	1165
febrero 14 2020	11	5200	2646
febrero 17 2020	8	864	670
febrero 24 2020	12	5350	1889
junio 04 2020	12	960	688

Tabla 6. Cantidad de metros procesados vs. makespan pruebas
Fuente: Elaboración propia

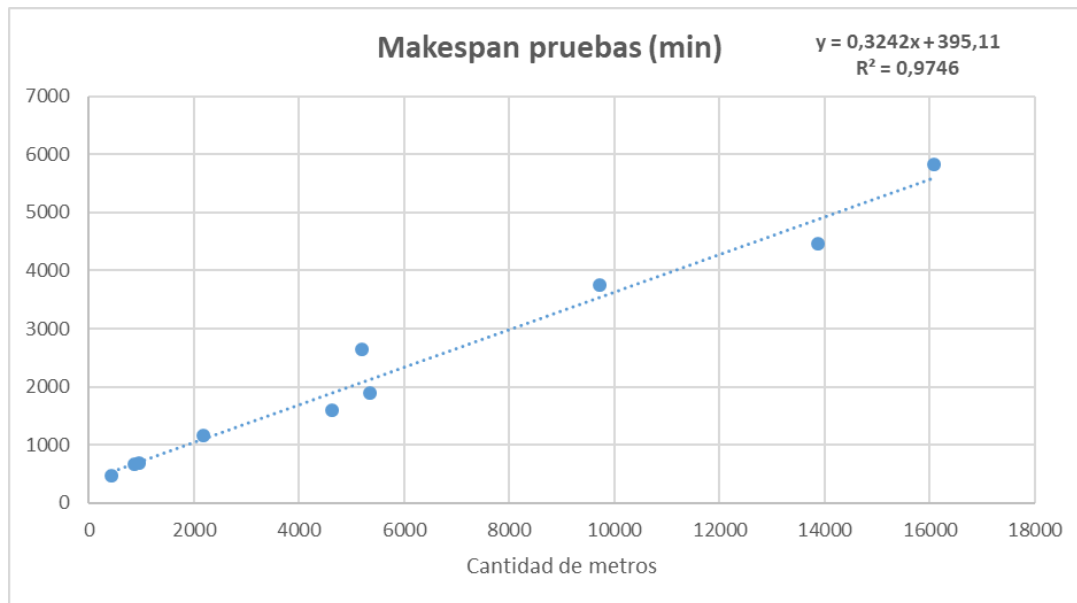


Figura 26. Gráfico análisis de correlación
Fuente: Elaboración propia

No se realizó un análisis de correlación para trabajos tardíos dado que en las pruebas sólo el 2.5% de los escenarios generaron un trabajo tardío.

La figura 27 muestra la comparación del makespan entre los resultados arrojados por el algoritmo y el makespan real para cada escenario.

En la figura 28 se puede observar que el tiempo de procesamiento del sistema, aunque depende de la cantidad de trabajos también se ve afectado por los parámetros que se apliquen a los escenarios, como se explicó anteriormente.

La figura 29 corresponde a la Gantt realizada según la secuenciación arrojada por el sistema correspondiente al programa de producción del día 27 de enero de 2020 con un total de 47 trabajos.

Fecha de prueba	Cantidad de trabajos	Iteraciones	Subpoblaciones	Cromosomas	% Mutación	Mejor ME pruebas	Media	Desvest	Desv%	TT pruebas
enero 12 2020	5	10	2	2	5	468	4.788	7,7	1,61%	0
		10	2	5	5	474	4.819	5,6	1,16%	0
		10	10	2	5	468	4.788	7,7	1,61%	0
		10	10	5	5	465	4.773	8,7	1,83%	0
		20	2	2	5	466	4.778	8,4	1,75%	0
		20	2	5	5	479	4.844	3,9	0,80%	0
		20	10	2	5	466	4.778	8,4	1,75%	0
		20	10	5	5	466	4.778	8,4	1,75%	0
enero 15 2020	10	10	2	2	5	3791	3.809	13	0,34%	0
		10	2	5	5	3758	3.792	24,6	0,65%	0
		10	10	2	5	3758	3.814	39,7	1,04%	0
		10	10	5	5	3758	4.051	207,6	5,12%	0
		20	2	2	5	3758	3.799	29,2	0,77%	0
		20	2	5	5	3758	3.823	46,6	1,22%	0
		20	10	2	5	3758	4.036	197	4,88%	0
		20	10	5	5	3758	4.071	221,6	5,44%	0
enero 22 2020	23	10	2	2	5	1617	1.814	139,9	7,71%	0
		10	2	5	5	1618	1.73	79,3	4,58%	0
		10	10	2	5	1597	1.804	146,7	8,13%	0
		10	10	5	5	1616	1.81	137,4	7,59%	0
		20	2	2	5	1622	1.803	128,2	7,11%	0
		20	2	5	5	1628	1818	134,6	7,40%	0
		20	10	2	5	1604	1.827	158	8,65%	0
		20	10	5	5	1617	1.826	148,1	8,11%	0
enero 27 2020	46	10	2	2	5	4456	4.46	3,2	0,07%	0
		10	2	5	5	4456	4729	193,5	4,09%	0
		10	10	2	5	4456	4.941	343,3	6,95%	0
		10	10	5	5	4456	5046	417,8	8,28%	0
		20	2	2	5	4456	4.742	202,4	4,27%	0
		20	2	5	5	4466	5.067	425	8,39%	0
		20	10	2	5	4456	5.203	528,4	10,16%	0
		20	10	5	5	4456	5.127	474,8	9,26%	0
febrero 03 2020	28	10	2	2	5	5869	5904	23,78	0,40%	0
		10	2	5	5	5999	6002	22,49	0,37%	1
		10	10	2	5	5835	6013	127,41	2,12%	0
		10	10	5	5	5873	6046	113,58	1,88%	0
		20	2	2	5	6058	6051	127,24	2,10%	1
		20	2	5	5	5856	6019	139,39	2,32%	0
		20	10	2	5	5854	6013	139,84	2,33%	0
		20	10	5	5	5853	6049	142,95	2,36%	0
febrero 07 2020	10	10	2	2	5	1170	1212	52,89	4,36%	0
		10	2	5	5	1165	1191	47,26	3,97%	0
		10	10	2	5	1170	1257	63,64	5,06%	0
		10	10	5	5	1165	1221	60,09	4,92%	0
		20	2	2	5	1165	1207	54,79	4,54%	0
		20	2	5	5	1165	1247	64,79	5,20%	0
		20	10	2	5	1165	1234	65,08	5,27%	0
		20	10	5	5	1165	1208	55,73	4,61%	0
febrero 14 2020	11	10	2	2	5	2651	2679	41,44	1,50%	0
		10	2	5	5	2646	2668	36,59	1,37%	0
		10	10	2	5	2646	2676	33,47	1,25%	0
		10	10	5	5	2646	2672	29,34	1,10%	0
		20	2	2	5	2651	2690	33,73	1,25%	0
		20	2	5	5	2651	2680	33,55	1,25%	0
		20	10	2	5	2646	2671	29,98	1,12%	0
		20	10	5	5	2646	2675	38,01	1,42%	0
febrero 17 2020	8	10	2	2	5	671	681	7,94	1,17%	0
		10	2	5	5	673	678	6,06	0,89%	0
		10	10	2	5	671	683	7,88	1,15%	0
		10	10	5	5	671	680	8,33	1,23%	0
		20	2	2	5	676	681	4,45	0,65%	0
		20	2	5	5	675	681	5,14	0,75%	0
		20	10	2	5	672	683	9,54	1,40%	0
		20	10	5	5	670	682	9,62	1,41%	0
febrero 24 2020	12	10	2	2	5	1928	1.99	44,2	2,22%	0
		10	2	5	5	2005	2.035	21,4	1,05%	0
		10	10	2	5	1948	2.155	146,7	6,81%	0
		10	10	5	5	1901	2.24	240,2	10,72%	0
		20	2	2	5	1969	2.163	137,5	6,35%	0
		20	2	5	5	1938	2060	86,4	4,19%	0
		20	10	2	5	1889	2.207	225,3	10,21%	0
		20	10	5	5	1944	2.157	151,1	7,00%	0
junio 04 2020	12	10	2	2	5	688	690	1,88	0,27%	0
		10	2	5	5	691	702	15,85	2,26%	0
		10	10	2	5	690	706	14,49	2,05%	0
		10	10	5	5	690	712	11,64	1,63%	0
		20	2	2	5	695	703	14,17	2,02%	0
		20	2	5	5	696	708	12,08	1,71%	0
		20	10	2	5	688	706	13,47	1,91%	0
		20	10	5	5	688	699	11,28	1,61%	0

Tabla 7. Resultados de las simulaciones

Fuente: Elaboración propia

Fecha de prueba	Cantidad de trabajos	Iteraciones	Subpoblaciones	Cromosomas	Mejor makespan pruebas	Makespan real	Trabajos tardíos pruebas	Trabajos tardíos real	Tipo de cromosoma
enero 12 2020	5	10	10	5	465	523	0	0	Hijo
enero 15 2020	10	10	10	5	3758	5940	0	1	Hijo
enero 22 2020	23	10	10	2	1597	1789	0	0	Mutación
enero 27 2020	46	20	10	2	4456	6380	0	6	Hijo
febrero 03 2020	28	10	10	2	5835	5986	0	3	Hijo
febrero 07 2020	10	20	2	5	1165	1534	0	0	Hijo
febrero 14 2020	11	20	10	5	2646	6238	0	1	Hijo
febrero 17 2020	8	20	10	5	670	779	0	0	Hijo
febrero 24 2020	12	20	10	2	1889	2359	0	0	Padre
junio 04 2020	12	20	10	2	688	766	0	0	Hijo

Tabla 8. Comparación mejores soluciones vs. datos reales

Fuente: Elaboración propia

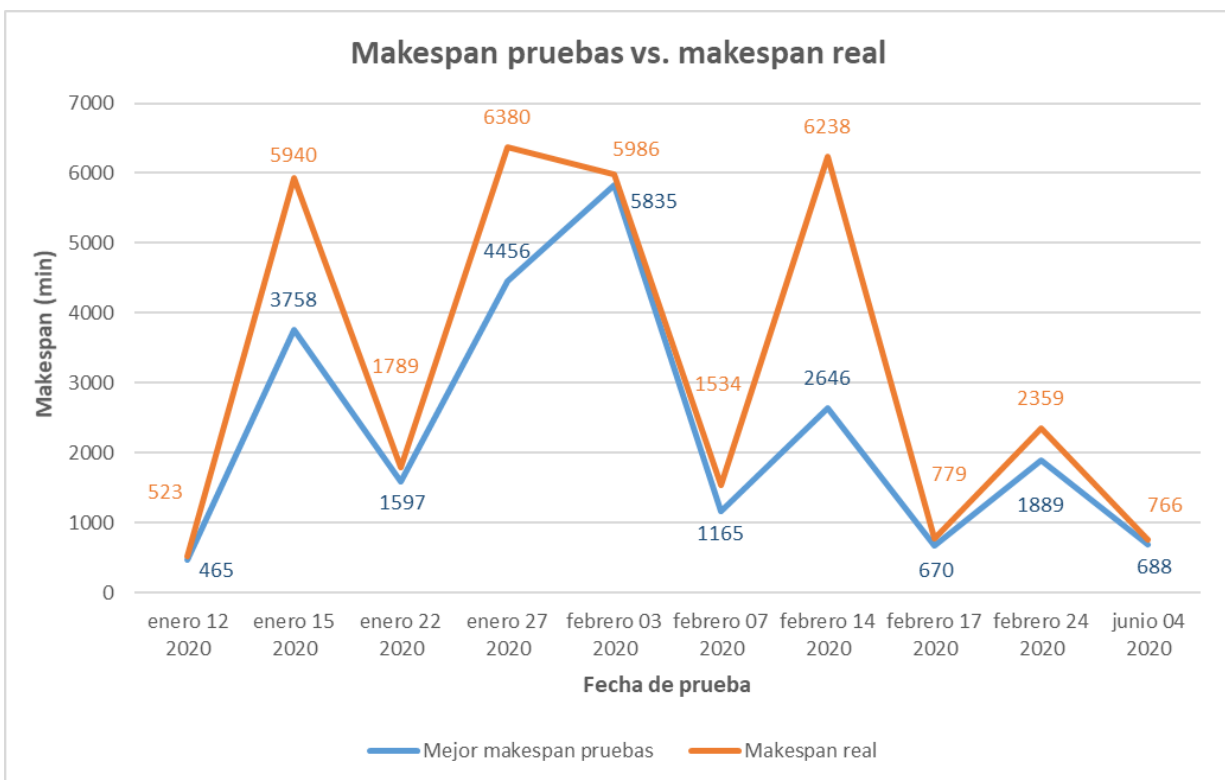


Figura 27. Comparación makespan pruebas vs. makespan real

Fuente: Elaboración propia

		35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68		
Tintorería	M1																																				
	M2																																				
	M3																																				
Mageba	M1	24	14;16	23;17;35		38	9;28	39		36,2		37;16;43;27	1;5,0	4;15																							
Cepillo	M1					23	17	38	9	28			36,2							37,27	15																
Termofijadora	M1	34;10	13;11	24;14;6		23;17;45		38	9	28	39			36,2;37							16;3;42	1;5,0	4;15														
Termofijadora2	M1		34;10;11;24	14;6							35		39									16;3;43	1;5	0;4													
Corte Monofilamento	M1			45	42		46;31				32		34	10	11	24	14	6;35	39;16	3;43	1	5															
Corte Ultrasonido	M1						45				22;44;40		30;42	46;31;7	32;8	12		34;10	13;11	24	14;6			23;17;35		38	9;28	39	36,2								
Enrollado	M1									45;22;44	40	30	42	46;31;7	32;8;12			34;10;13;11	24	14;6			23;17;35		38	9;28	39										
Empaque	M1											45	22;44;40	30;42	46;31;7	32;8	12		34;10;13;11	24	14;6			23;17;35		38	9;28	39									

		69	70	71	72	73	74	75
Tintorería	M1							
	M2							
	M3							
Mageba	M1							
Cepillo	M1							
Termofijadora	M1							
Termofijadora2	M1							
Corte Monofilamento	M1							
Corte Ultrasonido	M1	36,2			37;16;3;43	5,0	4;15	
Enrollado	M1			36,2;37	16;3;43;27	1;5,0	4;15	
Empaque	M1				36,2	37;16;3;43;27	4;15	

Figura 29. Diagrama de Gantt
Fuente: Elaboración propia

7 CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

1. La revisión del estado del arte del problema de scheduling en un entorno flow shop híbrido flexible sirvió como base para determinar las técnicas de solución aplicables, así como las complejidades existentes en la programación de producción en la industria textil.
2. Los resultados arrojados por el sistema desarrollado demuestran que la nueva metodología de programación logra reducir el makespan y la cantidad de trabajos tardíos en la empresa; este aspecto toma gran importancia teniendo en cuenta que el reducir estos dos elementos no sólo tiene impacto en la productividad de la compañía sino a nivel organizacional, ya que puede ser un factor importante de competitividad en la industria.
3. Una de las ventajas que ofrece el sistema radica en el tiempo que demora en generar una secuencia de programación comparado con un modelo de programación empírico.
4. Dada la complejidad del sistema productivo elegido como objeto de estudio se identificó que el modelo que se debió desarrollar no estaba contemplado dentro de los modelos matemáticos encontrados en las investigaciones que abarcan este tipo de problema.
5. En la investigación realizada se encontraron pocos proyectos que abordan sistemas reales de manufactura con configuraciones de producción de tipo flow shop híbrido flexible en la industria textil, en estos, se evidenció que la función objetivo común era la minimización del makespan a través de algoritmos genéticos probados con escenarios construidos por los autores;

es por eso que este trabajo aporta un valor a la investigación de este tipo de problemas a través de la solución de dos objetivos (makespan y trabajos tardíos) y el desarrollo de un sistema con la capacidad de procesar casos de en un ambiente real de manufactura (textiles técnicos).

6. Para este trabajo se eligió como base de la metodología diseñada el funcionamiento de los algoritmos genéticos dado de que es una metaheurística capaz de conseguir soluciones de alta calidad de forma eficiente en problemas de gran tamaño y complejidad.
7. Es posible mejorar los ANS teniendo en cuenta que el makespan disminuyó y con esto se pueden lograr tiempos de entrega menores a 14 días dependiendo de la cantidad de trabajos a procesar.
8. Según el análisis de correlación de los datos de prueba suministrados por la empresa y el análisis realizado a los resultados de las pruebas ejecutadas en el sistema desarrollado, se puede concluir que con la metodología propuesta el coeficiente de correlación presenta una mejora de $R^2 = 0,24$ lo cual indica una menor desviación entre la correlación lineal de las variables cantidad de metros vs. makespan; no se realizó un análisis de correlación de número de trabajos tardíos dado que de los 80 escenarios de prueba solamente dos presentaron cromosomas con un trabajo tardío.

7.2 RECOMENDACIONES

1. Se recomienda la implementación del modelo por parte de Kromia S.A.S, como herramienta para la programación de la producción que, además de mejorar la capacidad productiva de la planta puede ser útil para definir estrategias tácticas y operativas con el fin de lograr un mejoramiento continuo en niveles de servicio, aprovechamiento de recursos, entre otros.
2. En futuras investigaciones es posible lograr un mejor rendimiento del algoritmo en cuanto a secuencias de programación si se tienen en cuenta restricciones de la industria textil como tiempos de preparación

dependientes de la secuencia, presencia de máquinas paralelas no relacionadas en las diferentes etapas de procesamiento, lotes de transferencia variables y factores que puedan afectar el cumplimiento y resultados de la programación entre los cuales se pueden presentar: modificaciones de pedidos, disponibilidad de recursos, avería de máquinas, restricciones de precedencia entre otros.

Otro aspecto que se puede trabajar es la utilización de una metaheurística en la fase 1 de asignación de trabajos a las máquinas de tintorería ya que para el desarrollo de la metodología diseñada se utilizó el fundamento de la heurística LPT, el usar una metaheurística puede generar la exploración de espacios de búsqueda no examinados por el modelo propuesto.

REFERENCIAS

- Abdullah Konak, D. (2006). Multi-Objective Optimization Using Genetic Algorithms: A Tutorial. *Reliability Engineering system Safety*.
- Acuña, Domínguez, & Toro. (2012). Una comparación entre métodos estadísticos clásicos y técnicas metaheurística en el modelamiento estadístico. *Scientia et Technica*, 17(50), 68 – 77.
- Aguilar, R. (2016). *Algoritmo genético aplicado a sistemas de manufactura flexible*. Trabajo de grado, Universidad Nacional Autónoma de México, Ciudad de México, México.
- Allahverdi, A., & Al-Anzi, F. (2008). The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time. *The International Journal of Advanced Manufacturing Technology*(37), 166 - 177.
doi:doi:10.1007/s00170-007-0950-y
- Allahverdi, Ng, Chen, & Kovalyov. (2008). A survey of scheduling problems with setup time or costs. *European Journal of Operational Research*, 187, 985 – 1032.
- Arranz, P., & Parra, T. (s.f.). *Algoritmos Genéticos*. Trabajo de grado, Universidad Carlos III, Madrid, España.
- Arroyo, J., & Claudio, E. (2005). Genetic local search for multi-objective flowshop. *European Journal of Operational Research*, 717-738.

- Belén, M., Moreno Pérez, J., & Moreno Vega, J. (2003). Metaheuristics: A global view. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 7-28.
- Carlier, J., & Latapie, B. (1991). Une methode arborescente pour resoudre les problemes cumulatifs. *R.A.I.R.O. - Recherche Operationnelle / Operations Research*, 311-340.
- Chen, Pan, & Lin. (2008). A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem. *Expert Systems with Applications*, 34(1), 570 – 577.
- Chicano García, J. (2007). *Metaheurísticas e ingeniería del software*. Tesis doctoral, Universidad de Málaga, Málaga, España.
- Coello, C. (2004). *Introducción a la computación evolutiva*. Notas de curso, Centro de Investigación y de Estudios Avanzados del IPIN, Ciudad de México, México.
- Corral, A. (2017). *Desarrollo y evaluación del algoritmos genéticos multiobjetivo. Aplicación al problema del viajante*. Trabajo de grado, Escuela Técnica Superior de Ingeniería Informática. Universidad Politécnica de Valencia, Valencia, España.
- De Micheli, M. (2009). *Algoritmo de búsqueda local iterativa para la programación de piezas es un sistema flow shop híbrido*. Trabajo de grado, Universitat Politecnica de Catalunya, Barcelona, España.
- Esquivel, S., Ferrero, S., Gallard, R., Salto, C., Alfonso, H., & Schuts, M. (2002). Enhanced evolutionary algorithms for single and multiobjective optimization in the job shop scheduling problem. *Knowledge-Based Systems*, 13-25.

- Gestal, M. (2013). *Introducción a los algoritmos genéticos*. Universidade da Coruña, Coruña, España.
- Gestal, M., Rivero, D., Rabuñal, J., Dorado, J., & Pazos, A. (2010). *Introducción a los Algoritmos Genéticos y la Programación Genética*. Monografía, Universidade da Coruña, Coruña, España.
- Gómez, Mendes, Riveros, & Resende. (2012). Parallel hybrid heuristics for the permutation flow shop problem. *Annals of Operations Research*, 1-16.
- Gomez, Romano, & Cruz. (2012). An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimise makespan. *Expert Systems with Applications*, 39(9), 8095 - 8107.
- González, P. (2013). *Diseño de una metodología de programación de producción para la reducción de costos en un flow shop hírido flexible mediante el uso de algoritmos genéticos. Aplicación a la industria textil*. Tesis de maestría, Universidad Nacional de Colombia, Manizales, Colombia.
- Gupta, A., & Chauhan, S. (2015). A heuristic algorithm for scheduling in a fow shop environment to minimize makespan. *International Journal of Industrial Engineering Computations*(6), 173 - 184.
- Haral, U., Chen, R., Ferrell Jr, W., & Kurz, M. (2006). Multiobjetive single machine scheduling with nontraditional requirements. *International Journal of Production Economics*, 574-584.

- Hekmatfar, M., Fatemi, G., & Karimi, B. (2011). Two stage reentrant hybrid flow shop with setup times and the criterion of minimizing makespan. *Applied Soft Computing*(11), 4530 – 4539.
- Ibrahim, El-shaer, Al-rwasheda, & Nawarra. (2013). Flow shop scheduling using genetic algorithm: historical review and categorization of procedures. *The Egyptian International Journal of Engineering Science and Technology*, 16(3), 1671 - 1685.
- Jarboui, B., Eddaly, M., & Siarry, P. (2011). A hybrid genetic algorithm for solving no-wait flowshopscheduling problems. *The International Journal of Advanced Manufacturing Technology*, 54, 1129 - 1143. doi:doi:10.1007/s00170-010-3009-4
- Jiménez Vargas, D. C. (2008). *Desarrollo de una herramienta computacional para la programación de la producción en sistemas de fabricación con centros de trabajo dispuestos en paralelo*. Trabajo de grado, Universidad de los Andes, Bogotá, D.C.
- Jungwattanakit, Reodecha, Chaovalitwongse, & Werner. (2008). Algorithms for Flexible Flow Shop Problems with Unrelated Parallel Machines, Setup Times and Dual Criteria. *International Journal of Advanced Manufacturing Technology*, 37(3-4), 354 – 370. doi:10.1007/s00170-007-0977-0
- Karmakar, & Manhaty. (s.f.). *Minimizing makespan for a flexible flow shop scheduling problem in a paint company*. Indian Institute of Technology, Department of Industrial Engineering and Management, Kharagpur, India.

Kleinberg, J., & Tardos, E. (2006). *Algorithm Design*. Boston: Pearson.

Lagunas Jiménez, J., Moo Yam, V., & Ortiz Moctezuma, B. (2013). Optimización Multiobjetivo mediante Algoritmos Genéticos: Aplicación a Controladores PID Robustos. *Programación Matemática y Software*, 33-45.

López, J., & Arango, J. (2015). Algoritmo Genético para Reducir el Makespan en un Flow Shop Híbrido Flexible con máquinas paralelas no relacionadas y tiempos de alistamiento dependientes de la secuencia. *Revista Entramado*, 2(1), 250 – 262.

Mahapatra, S., Dash, R., & Pradhan, S. (2017). Heuristics Techniques for Scheduling Problems with Reducing Waiting Time Variance. *Intechopen*.

Mejía Azcárate, F. (Enero de 2015). <https://programadetextilizacion.blogspot.com/>.
Obtenido de <https://programadetextilizacion.blogspot.com/2015/02/capitulo-10-la-maquinaria-de-tintoreria.html>

Melián, Moreno P., & Moreno V. (2003). Metaheurísticas: una visión global. *Revista Iberoamericana de Inteligencia Artificial*, 7(19), 7 – 28.

Morillo, D., Moreno, L., & Díaz, J. (2014). Metodologías Analíticas y Heurísticas para la Solución del Problema de Programación de Tareas con Recursos Restringidos. *Ingeniería y Ciencia*, 247-291.

Moujahid, Inza, & Larrañaga. (s.f.). *Algoritmos genéticos*. Universidad del País Vasco, Lejona, España.

- Naik, A., & Gacén, J. (1985). Influencia del tratamiento y condiciones del termofijado en los parámetros de los tejidos de calada de poliéster. *Boletín Intextar del Instituto de Investigación Textil y de Cooperación Industrial*(87), 61 - 72.
- Oguz, C., & Ercan, M. (2005). A Genetic Algorithm for Hybrid Flow-shop Scheduling with Multiprocessor Tasks. *Journal of Scheduling*(8), 323 - 351.
- Osorio, J., Lasso, D., & Ruiz, G. (2012). Job shop scheduling biobjetivo mediante enfriamiento simulado y enfoque de pareto. *Revista Ingenierías Universidad de Medellín*, 113-126.
- Perea González, V. (2015). *Diseño de un procedimiento basado en algoritmos genéticos para dar solución al problema de scheduling admitiendo rechazos en escenarios flexible flow shop*. Tesis Mestría, Universidad Distrital Francisco José de Caldas, Bogotá, D.C.
- Ponnambalam, S. (2003). Genetic algorithms for sequencing problems in mixed model assembly lines. *Computers & Industrial Engineering*, 669-690.
- Quan-Ke, P., & Ruiz, R. (2014). An effective iterated greedy algorithm for the mixed no-idle flowshop scheduling problem. *Omega*, 44, 41 - 50.
doi:10.1016/j.omega.2013.10.002.
- Ranjan, & Mahapatra. (2011). A swarm optimization approach for flexible flow shop scheduling with multiprocessor tasks. *International Journal of Advanced Manufacturing Technology*. doi:10.1007/s00170-011-3807-3

Ren, Diao, & Luo. (2012). Optimal results and numerical simulations for flow shop scheduling problems. *Journal of Applied Mathematics*, 2012, 1 – 9.

doi:10.1155/2012/395947

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

Ruiz, & Vázquez. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 1-18.

Salazar Hornig, E., & Sarzuri Guarachi, R. (2015). Algoritmo genético mejorado para la minimización de la tardanza total en un flow shop flexible con tiempos de preparación dependientes de la secuencia. *Ingeniare. Revista Chilena de Ingeniería*, 23(1), 118 - 127.

Salazar, & Sarzuri. (2015). Algoritmo genético mejorado para la minimización de la tardanza total en un flowshop flexible con tiempos de preparación dependientes de la secuencia. *Revista Chilena de Ingeniería*, 23(1), 118 – 127.

Sánchez, S., & Salido Gregorio, M. (2017). *Desarrollo de técnicas de búsqueda metaheurísticas en problemas de scheduling multiobjetivo*. Trabajo fin de máster - MÁSTER UNIVERSITARIO EN INTELIGENCIA ARTIFICIAL, Departamento de sistemas informáticos y computación Universidad Politécnica de València, Valencia, España.

- Schaffer, J. (1985). Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of an international conference on genetic algorithms and their applications*, 92 - 101.
- Suárez, O. (2011). Una aproximación a la heurística y metaheurísticas. *INGEUAN*, 44-51.
- T'kindt, V., & Charles, B. (2006). *Multicriteria Scheduling*. New York.
- Tasgetiren, M., Liang, Y.-C., Sevkli, M., & Gencyilmaz, G. (2007). A particle swarm optimization algorithm for makespan and total flow time minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research*(177), 1930 - 1947.
- Testex. (27 de Diciembre de 2019). www.testextextile.com.es. Obtenido de <https://www.testextextile.com/es/resumen-del-método-de-teñido-textil-2-formas-y-5-pasos/>
- Toro Ocampo, E., Restrepo Grisales, Y., & Granada Echeverri, M. (2006). Algoritmo genético modificado aplicado al problema de secuenciamiento de tareas en sistemas de producción lineal - flow shop. *Scientia Et Technica*, XII(30), 285 - 290.
- Varadharajan, T., & Rajendran, C. (2004). A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs. *European Journal of Operational Research*, 167, 772-795.

- Zames, G., Ajlouni, N., Hollan, J., Hills, W., & Golberg, D. (1981). Genetic algorithms in search, optimization, and machine learning. *Information Technology Journal*, 301-302. Obtenido de <https://www.semanticscholar.org/paper>
- Zandieh, & Karimi. (2011). An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flowshop with sequence-dependent setup times. *Jornal of Intelligent Manufacturing*, 22(6), 979 - 989.
- Zobolas, Tarantilis, & Ioannou. (2009). Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. *Computers & Operations Research*, 36, 1249 - 1267.
- , N. P., López, J. M., & E., H. H. (2019). Una Metaheurística basada en el Algoritmo Genético de Ordenamiento No-Dominado II, aplicado al Problema de Ruteo de Vehículos de Productos Perecederos. *Información tecnológica*, 718-764.
- Abdullah Konak, D. W. (2006). *Multi-Objective Optimization Using Genetic Algorithms: A Tutorial*. New Jersey: Reliability Engineering system Safety.
- Acuña, Domínguez, & Toro. (2012). Una comparación entre métodos estadísticos clásicos y técnicas metaheurística en el modelamiento estadístico. *Scientia et Technica*, 17(50), 68 – 77.
- Aguilar, R. (2016). *Algoritmo genético aplicado a sistemas de manufactura flexible*. Trabajo de grado, Universidad Nacional Autónoma de México, Ciudad de México, México.
- Allahverdi, A., & Al-Anzi, F. (2008). The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time. *The International Journal*

- of Advanced Manufacturing Technology*(37), 166 - 177. doi:10.1007/s00170-007-0950-y
- Allahverdi, Ng, Chen, & Kovalyov. (2008). A survey of scheduling problems with setup time or costs. *European Journal of Operational Research*, 187, 985 – 1032.
- Arranz, P., & Parra, T. (s.f.). *Algoritmos Genéticos*. Trabajo de grado, Universidad Carlos III, Madrid, España.
- Arroyo, J., & Claudio, E. (2005). *Genetic local search for multi-objective flowshop*. 717-738.: *European Journal of Operational Research*.
- Baesler, F., Ceballos, L., & Ramírez, M. (2006). MULTIOBJECTIVE MOLDING MACHINE SCHEDULING USING MEMETIC ALGORITHM. *Maderas. Ciencia y tecnología*, 183-192.
- Belén, M., Moreno Pérez, J. A., & Moreno Vega, J. M. (2003). Metaheuristics: A global view. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 7-28.
- Carlier, J., & Latapie, B. (1991). Une methode arborescente pour resoudre les problemes cumulatifs . *R.A.I.R.O. - Recherche Operationnelle / Operations Research*, 311-340.
- Chen, Pan, & Lin. (2008). A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem. *Expert Systems with Applications*, 34(1), 570 – 577.
- Chicano García, J. (2007). *Metaheurísticas e ingeniería del software*. Tesis doctoral, Universidad de Málaga, Málaga, España.
- Coello, C. (2004). *Introducción a la computación evolutiva*. Notas de curso, Centro de Investigación y de Estudios Avanzados del IPIN, Ciudad de México, México.
- Coello, C. A. (1999). A comprehensive survey of evolutionary-based multiobjective optimizations techniques. *Knowledge and Information Systems. An International Journal*, 269–308.

- Corral, A. (2017). *Desarrollo y evaluación de algoritmos genéticos multiobjetivo. Aplicación al problema del viajante*. Trabajo de grado, Escuela Técnica Superior de Ingeniería Informática. Universidad Politécnica de Valencia, Valencia, España.
- De Micheli, M. (2009). *Algoritmo de búsqueda local iterativa para la programación de piezas es un sistema flow shop híbrido*. Trabajo de grado, Universitat Politecnica de Catalunya, Barcelona, España.
- Deb, K. (2001). Multi-Objective Optimization Using Evolutionary Algorithms. *Kanpur Genetic Algorithms Laboratory*, 1-24.
- Esquivel, S., Ferrero, S. W., Gallard, R., Salto, C., Alfonso, H., & Schuts, M. (2002). Enhanced evolutionary algorithms for single and multiobjective optimization in the job shop scheduling problem. *Knowledge-Based Systems*, 13-25.
- Gestal, M. (2013). *Introducción a los algoritmos genéticos*. Universidade da Coruña, Coruña, España.
- Gestal, M., Rivero, D., Rabuñal, J., Dorado, J., & Pazos, A. (2010). *Introducción a los Algoritmos Genéticos y la Programación Genética*. Monografía, Universidade da Coruña, Coruña, España.
- Glover, F., & Melián, B. (2003). Tabu Search. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 29-48.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Reading, Massachusetts.
- Gómez, Mendes, Riveros, & Resende. (2012). Parallel hybrid heuristics for the permutation flow shop problem. *Annals of Operations Research*, 1 - 16.
- Gomez, Romano, & Cruz. (2012). An agent-based genetic algorithm for hybrid flowshops with sequence dependent setup times to minimise makespan. *Expert Systems with Applications*, 39(9), 8095 - 8107.

- Gupta, A., & Chauhan, S. R. (2015). A heuristic algorithm for scheduling in a few shop environment to minimize makespan. *International Journal of Industrial Engineering Computations*(6), 173 - 184.
- Haral, U., Chen, R. W., Ferrell Jr, W., & Kurz, M. B. (2006). Multiobjective single machine scheduling with nontraditional requirements. *International Journal of Production Economics*, 574-584.
- Hekmatfar, M., Fatemi, G., & Karimi, B. (2011). Two stage reentrant hybrid flow shop with setup times and the criterion of minimizing makespan. *Applied Soft Computing*(11), 4530 – 4539.
- Henig, M., & Buchanan, J. T. (1997). Tradeoff directions in multiobjective optimization. *Mathematical Programming*, 357-374.
- Ibrahim, El-shaer, Al-rwasheda, & Nawarra. (2013). Flow shop scheduling using genetic algorithm: historical review and categorization of procedures. *The Egyptian International Journal of Engineering Science and Technology*, 16(3), 1671 - 1685.
- Jarboui, B., Eddaly, M., & Siarry, P. (2011). A hybrid genetic algorithm for solving no-wait flowshopscheduling problems. *The International Journal of Advanced Manufacturing Technology*, 54, 1129 - 1143. doi:10.1007/s00170-010-3009-4
- Jiménez Vargas, D. C. (2008). *Desarrollo de una herramienta computacional para la programación de la producción en sistemas de fabricación con centros de trabajo dispuestos en paralelo*. Trabajo de grado, Universidad de los Andes, Bogotá, D.C.
- Jungwattanakit, Reodecha, Chaovalitwongse, & Werner. (2008). Algorithms for Flexible Flow Shop Problems with Unrelated Parallel Machines, Setup Times and Dual Criteria. *International Journal of Advanced Manufacturing Technology*, 37(3-4), 354 – 370. doi:10.1007/s00170-007-0977-0
- Karmakar, & Manhaty. (s.f.). *Minimizing makespan for a flexible flow shop scheduling problem in a paint company*. Indian Institute of Technology, Department of Industrial Engineering and Management, Kharagpur, India.

- Kleinberg, J., & Tardos, E. (2006). *Algorithm Design*. Boston: Pearson.
- Lagunas Jiménez, J. R., Moo Yam, V., & Ortiz Moctezuma, B. (2013). Optimización Multiobjetivo mediante Algoritmos Genéticos: Aplicación a Controladores PID Robustos. *Programación Matemática y Software* , 33-45.
- López , J. (2013). *etodología de programación de producción en un flow shop híbrido flexible con el uso de algoritmos genéticos para reducir el makespan. Aplicación en la industria textil*. Tesis de Maestría, Universidad Nacional de Colombia, Manizales, Colombia.
- López, & Arango. (2015). Algoritmo Genético para Reducir el Makespan en un Flow Shop Híbrido Flexible con máquinas paralelas no relacionadas y tiempos de alistamiento dependientes de la secuencia. *Revista Entramado*, 2(1), 250 – 262.
- Mahapatra, S., Dash, R. R., & Pradhan, S. K. (13th de April de 2017). Heuristics Techniques for Scheduling Problems with Reducing Waiting Time Variance. intechopen.
- Martín, J. A. (2016). Mejora de tiempos de entrega en un flow shop híbrido flexible usando técnicas inteligentes. Aplicación en la industria de tejidos técnicos. Manizales, Colombia: Universidad Nacional de Colombia.
- Mejía Azcárate, F. (Enero de 2015). <https://programadetextilizacion.blogspot.com/>. Obtenido de <https://programadetextilizacion.blogspot.com/2015/02/capitulo-10-la-maquinaria-de-tintoreria.html>
- Melián, Moreno P., & Moreno V. (2003). Metaheurísticas: una visión global. *Revista Iberoamericana de Inteligencia Artificial*, 7(19), 7 – 28.
- Morillo, D., Moreno, L., & Díaz, J. (2014). Metodologías Analíticas y Heurísticas para la Solución del Problema de Programación de Tareas con Recursos Restringidos. *Ingeniería y Ciencia*, 247-291.
- Moujahid, Inza, & Larrañaga. (s.f.). *Algoritmos genéticos*. Universidad del País Vasco, Lejona, España.

- Naik, A., & Gacén, J. (1985). Influencia del tratamiento y condiciones del termofijado en los parámetros de los tejidos de calada de poliéster. *Boletín Intextar del Instituto de Investigación Textil y de Cooperación Industrial*(87), 61 - 72.
- Oguz, C., & Ercan, M. (2005). A Genetic Algorithm for Hybrid Flow-shop Scheduling with Multiprocessor Tasks. *Journal of Scheduling*(8), 323 - 351.
- Osorio, J. C., Lasso, D. F., & Ruiz, G. A. (2012). Job shop scheduling biobjetivo mediante enfriamiento simulado y enfoque de pareto . *Revista Ingenierías Universidad de Medellín*, 113-126.
- Perea González, V. A. (2015). *Diseño de un procedimiento basado en algoritmos genéticos para dar solución al problema de scheduling admitiendo rechazos en escenarios flexible flow shop*. Tesis Mestría, Universidad Distrital Francisco José de Caldas, Bogotá, D.C.
- PhD, J. B. (2011). *Non-dominated Sorting Genetic Algorithm*. Melbourne, Australia: Swinburne University of Technology. Obtenido de *Clever Algorithms: Nature-Inspired Programming Recipes*.
- Ponnambalam, S. A. (2003). *Genetic algorithms far sequencing problems in mixed model assembly fines*. 669-690: Computers & Industrial Engineering.
- Quan-Ke, P., & Ruiz, R. (2014). An effective iterated greedy algorithm for the mixed no-idle flowshop scheduling problem. *Omega*, 44, 41 - 50. doi:10.1016/j.omega.2013.10.002.
- Ranjan, & Mahapatra. (2011). A swarm optimization approach for flexible flow shop scheduling with multiprocessor tasks. *International Journal of Advanced Manufacturing Technology*. doi:10.1007/s00170-011-3807-3
- Ren, Diao, & Luo. (2012). Optimal results and numerical simulations for flow shop scheduling problems. *Journal of Applied Mathematics*, 2012, 1 - 9. doi:10.1155/2012/395947

- Ribas, Leisten, & Framiñan. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research*, 37, 1439 - 1454.
- Ruíz, & Vázquez. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205, 1 – 18.
- Salazar Hornig, E., & Sarzuri Guarachi, R. (2015). Algoritmo genético mejorado para la minimización de la tardanza total en un flow shop flexible con tiempos de preparación dependientes de la secuencia. *Ingeniare. Revista Chilena de Ingeniería*, 23(1), 118 - 127. (Carlier & Latapie, 1991)
- Salazar, & Sarzuri. (2015). Algoritmo genético mejorado para la minimización de la tardanza total en un flowshop flexible con tiempos de preparación dependientes de la secuencia. *Revista Chilena de Ingeniería*, 23(1), 118 – 127.
- Sánchez, S. F., & Salido Gregorio, M. Á. (2017). Desarrollo de técnicas de búsqueda metaheurísticas en problemas de scheduling multiobjetivo. *Trabajo fin de máster - MÁSTER UNIVERSITARIO EN INTELIGENCIA ARTIFICIAL*. Valencia, España: Departamento de sistemas informáticos y computación Universidad Politécnica de València.
- Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated denetic algoirhtms. *Proceedings of an international conference on genetic algorithms and their applications*, 92 - 101.
- Suárez, O. d. (2011). Una aproximación a la heuristica y metaheurísticas. *INGEUAN*, 44-51.
- T'kindt, V., & Charles, B. J. (2006). *Multicriteria Scheduling*. New York: Springer.
- Tasgetiren, M., Liang, Y.-C., Sevkli, M., & Gencyilmaz, G. (2007). A particle swarm optimization algorithm for makespan and total flow time minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research*(177), 1930 - 1947.

- Testex. (27 de Diciembre de 2019). www.testextextile.com.es. Obtenido de <https://www.testextextile.com/es/resumen-del-método-de-teñido-textil-2-formas-y-5-pasos/>
- Toro Ocampo, E., Restrepo Grisales, Y. S., & Granada Echeverri, M. (2006). Algoritmo genético modificado aplicado al problema de secuenciamiento de tareas en sistemas de producción lineal - fow shop. *Scientia Et Technica*, XII(30), 285 - 290.
- Varadharajan, T., & Rajendran, C. (2004). A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs. *European Journal of Operational Research* 167, 772-795.
- Yaurima-Basaldúa, V. H., Tchernykh, A., Villalobos-Rodríguez, F., Salomon-Torres, & Ricardo. (2018). Hybrid Flow Shop with Unrelated Machines, Setup Time, and Work in Progress Buffers for Bi-Objective Optimization of Tortilla Manufacturing. *Algorithms MDTI*, 1-22.
- Zames, G., Ajlouni, N., Hollan, J., Hills, W., & Golberg, D. (1981). Genetic algorithms in search, optimization, and machine learning. *Information Technology Journal*, 301-302. Obtenido de <https://www.semanticscholar.org/paper>
- Zandieh, & Karimi. (2011). An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flowshop with sequence-dependent setup times. *Jornal of Intelligent Manufacturing*, 22(6), 979 - 989.
- Zobolas, Tarantilis, & Ioannou. (2009). Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. *Computers & Operations Research*, 36, 1249 - 1267.