Optimización en tiempos de producción de sistemas de manufactura automatizados con algoritmos heurísticos usando software libre

Izquierdo Córdova, Fabián Eduardo

Vicerrectorado De Investigación Innovación y Transferencia De Tecnología

Centro de Posgrados

Maestría En Manufactura Y Diseño Asistidos Por Computador

Trabajo de titulación, previo a la obtención del título de Magíster En Manufactura Y Diseño Asistidos Por Computador

Ing. Escobar Carvajal, Luis Fernando

19 de septiembre del 2020

URKUND

Document Information

Analyzed document	TESIS IZQUIERDO CORDOVA FABIAN EDUARDO MMDAC (1).docx (D82354462)
Submitted	10/22/2020 12:39:00 AM
Submitted by	Cortez Pazmiño Byron Hernan
Submitter email	bhcortez@espe.edu.ec
Similarity	10%
Analysis address	bhcortez.espe@analysis.urkund.com

Sources included in the report

SA	**Universidad de las Fuerzas Armadas ESPE / Tesis MDAC Edwin García Biblioteca.docx** Document Tesis MDAC Edwin García Biblioteca.docx (D26124903) Submitted by: elgarcia@espe.edu.ec Receiver: lfescobar.espe@analysis.urkund.com	22
W	URL: https://www.coursehero.com/file/68202691/FormatosDocumentosPregrado2020docx/ Fetched: 10/22/2020 12:40:00 AM	1
W	URL: https://portal.espe.edu.ec/biblioteca/wp-content/uploads/2019/11/TitulacionPregrad ... Fetched: 10/22/2020 12:40:00 AM	2
W	URL: https://repositorio.espe.edu.ec/bitstream/21000/12856/1/T-ESPE-053854.pdf Fetched: 9/30/2019 1:30:12 AM	2
W	URL: https://docplayer.es/53340939-1a-ed-iniciativa-latinoamericana-de-libros-de-texto- ... Fetched: 10/21/2019 1:07:23 PM	1
W	URL: https://ieeexplore.ieee.org/abstract/document/6721642/authorsGarc Fetched: 10/22/2020 12:40:00 AM	2
SA	**1A_Mugruza_Torres_Roberto_Guillermo_titulo profesional_2018.doc** Document 1A_Mugruza_Torres_Roberto_Guillermo_titulo profesional_2018.doc (D43548974)	6
SA	**CACERES BRIONES RONALD ANDRES_PT-010517.pdf** Document CACERES BRIONES RONALD ANDRES_PT-010517.pdf (D29676954)	3

Escobar Carvajal, Luis Fernando

DIRECTOR

Vicerrectorado De Investigación, Innovación Y Transferencia De Tecnología

Centro de Posgrados

Certificación

Certifico que el trabajo de titulación, "**Optimización en tiempos de producción de sistemas de manufactura automatizados con algoritmos heurísticos usando software libre**" fue realizado por el señor **Izquierdo Córdova Fabián Eduardo** el mismo que fue revisado y analizado en su totalidad, por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 19 de Julio del 2020

Escobar Carvajal, Luis Fernando

Director

C.C.: 1002403200

Vicerrectorado De Investigación, Innovación y Transferencia De Tecnología

Centro de Posgrados

Responsabilidad de autoría

Yo, **Izquierdo Córdova, Fabián Eduardo**, con cédula de ciudadanía n° 1717528168, declaro que el contenido, ideas y criterios del trabajo de titulación: **Optimización en tiempos de producción de sistemas de manufactura automatizados con algoritmos heurísticos usando software libre**, es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 19 de julio de 2020

Izquierdo Córdova, Fabián Eduardo

C.C.: 1717528168

Vicerrectorado De Investigación, Innovación y Transferencia De Tecnología

Centro de Posgrados

Autorización de publicación

Yo, **Izquierdo Córdova, Fabián Eduardo** con cédula de ciudadanía n° 1717528168, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Optimización en tiempos de producción de sistemas de manufactura automatizados con algoritmos heurísticos usando software libre**, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 19 de julio de 2020

Izquierdo Córdova Fabián Eduardo

C.C.: 1717528168

Dedicatoria

Este trabajo lo dedico a mi familia, en especial a mis padres, que a pesar de que pasen los años, siempre están pendientes de que cumpla las metas propuestas y me brindan su apoyo en cada etapa de mi vida.

A mi hermana Diana, quién con su ejemplo me ha impulsado seguir sus pasos en bien de superarme y conseguir un futuro mejor.

A mi hermana Patricia, quién me ha demostrado, que lo que tienes que ponerle a la vida es la voluntad de cumplir tus objetivos y no perder la fe en tus capacidades.

Fabián Eduardo Izquierdo Córdova

Agradecimiento

A Dios, por bendecirme en cada paso de esta experiencia

A mi madre, por cuidarme y su amor incondicional.

A mi padre, por ser un ejemplo a seguir, por todas las enseñanzas de vida, y por mantenerse al pendiente de mí.

Al Ing. Edwin García, por su amistad y apoyo incondicional, durante este proceso.

Un especial agradecimiento a mi Director de Tesis Ing. Luis Fernando Escobar Carvajal por sus consejos, su guía, su amistad y por creer en mí a pesar de las limitaciones ajenas que se han producido en el transcurso del desarrollo de este proyecto.

Fabián Eduardo Izquierdo Córdova

Índice de contenido

Carátula.....1

Urkund.....2

Certificación.....3

Responsabilidad de autoría.....4

Autorización de publicación.....5

Dedicatoria.....6

Agradecimiento.....7

Índice de contenido.....8

Resumen.....16

Abstract.....17

Introducción.....18

Problema.....18

Desarrollo del problema.....18

Planteamiento del problema.....20

Formulación del problema a resolver.....20

Justificación e importancia.....21

Objetivos.....22

Objetivo general.....22

Objetivos específicos.....22

Hipótesis.....23

Descripción del sistema de producción y algoritmos metaheurísticos de optimización.....24

Sistemas de manufactura.....24

Definición.....24

Tipos de sistemas de manufacturas.....24

Tipos de distribución de fms dentro de una planta.....25

Funcionamiento de los sistemas de manufactura.....29

Sistemas de producción.....30

Definición.....30

Tipos de procesos productivos.....29

Factores que integran un sistema de producción.....31

Métodos heurísticos o de optimización.....32

Escalada (hill climbing).....33

Enfriamiento simulado (simulated annealing).....34

Optimización basada en nubes de partículas (particle swarm optimization).....35

Algoritmos de colonias de hormigas (ant colony optimization).....36

Algoritmos genéticos (genetic algorithms).....37

Codificación de problemas.....40

Estructura de un algoritmo.....41

Definición de la solución.....42

Problema.....42

Introducción a jaamsim.....43

Requisitos de instalación.....44

Presentación de jaamsim.....44

Guía para ejemplo básico de simulación.....46

Creación de objetos modelo.....47

Algoritmos genéticos.....51

Proceso para crear un algoritmo genético.....52

Población Inicial.....53

Codificación.....53

Adaptabilidad.....54

Selección de individuos.....55

Torneo.....56

Selección.....57

Cruce.....57

Cruce basado en un punto.....58

Cruce punto a punto.....58

Cruce multipunto.....58

Mutación.....59

Implementación de la solución.....61

Modelado fms.....61

Modelado fms en el simulador.....64

Desarrollo del algoritmo metaheurístico.....67

Código para Iniciar la población.....71

Código para convertir la población.....73

Código para calidad del individuo.....74

Código para selección de parejas.....75

Código para copiarse.....75

Código para combinación y mutación.....76

Código para ver población.....77

Diagrama de flujo.....79

Resultados..... 81

Resolución del caso de estudio.....81

Comparación de resultados.....83

Análisis de resultados.....85

Conclusiones y recomendaciones.....88

Conclusiones.....88

Recomendaciones.....89

Proyectos futuros.....90

Bibliografía.....91

Apéndice A. Código fuente del algoritmo genético.....96

Índice de tablas

Tabla 1. *Objeto SimEntity a crear*.....48

Tabla 2. *Ajustes de vista*.....49

Tabla 3. *Objetos adicionales para modelos de ejemplo*.....50

Tabla 4. *Comparación de cromosoma y adapatabilidad*.....54

Tabla 5. *Selección de individuos en base*.....55

Tabla 6. *Re numeración según adaptabilidad de torneo*.....56

Tabla 7. *Codificación para cromosoma de secuencia*.....73

Tabla 8. *Especificaciones de productos*.....81

Tabla 9. *Análisis de resultados*.....83

Índice de figuras

Figura 1. *Distribución de FMS en línea*.....26

Figura 2. *Distribución de FMS en ciclo*.....26

Figura 3. *Distribución FMS en escalera*.....27

Figura 4. *Distribución FMS en campo abierto*.....27

Figura 5. *Distribución FMS centrado en un robot*.....28

Figura 6. *Enfriamiento simulado*.....34

Figura 7. *Tipos de entornos*.....36

Figura 8. *Comportamiento de hormigas*.....37

Figura 9. *Pasos de un algoritmo genético*.....39

Figura 10. *Individuo genético binario*.....40

Figura 11. *Codificación de una red de neuronas artificiales*.....40

Figura 12. *Interfaz inicio Jaamsim*.....45

Figura 13. *Ejemplo de simulación en Jaamsim* 46

Figura 14. *SimEntity insertada en espacio de trabajo*.....47

Figura 15. *Vista de ventana configurada*.....49

Figura 16. *Vista final del paso 1*.....51

Figura 17. *Cadena cromosoma*.....53

Figura 18. *Cruce basado en un punto*.....58

Figura 19. *Cruce multipunto*.....59

Figura 20. *Mutación*.....59

Figura 21. *Diagrama de bloque FMS*.....61

Figura 22. *Diagrama de flujo para la simulación en Jaamsim*.....63

Figura 23. *Modelo desarrollado en Jaamsim*.....63

Figura 24. *Bloques de funcionamiento Jaamsim.*.....65

Figura 25. *Diagrama de flujo algoritmo genético*.....68

Figura 26. *Función principal código genético sección A.*.....70

Figura 27. *Función principal código genético sección B.*.....71

Figura 28. *Código población inicial* 72

Figura 29. *Código convertir la población.*.....73

Figura 30. *Código calidad individuo.*.....74

Figura 31. *Código para selección de parejas.*.....75

Figura 32. *Código para copiarse.*.....76

Figura 33. *Código para Combinación y mutación…….*.....77

Figura 34. *Código para ver la población.*.....78

Figura 35. *Diagrama de flujo metaheurístico…….*.....79

Figura 36. *Ingreso de orden de producción a modelo en Jaamsim.*.....82

Figura 37. *Verificación de valores en simulación……..*.....83

Figura 38. *Evaluación de tiempo por lote……..*.....85

Figura 39. *Reducción porcentual absoluta en tiempos de producción.*.....86

Resumen

Esta investigación desarrolla la integración entre la simulación de un sistema de manufactura flexible y la optimización de uno de sus parámetros resultante mediante la aplicación de un método meta heurístico basado en algoritmos genéticos. El estudio del proyecto, se enfoca en la aplicación de software libre para la ejecución de este propósito, en razón que los costes en la implementación de esta integración con software comercial resultan elevados para la mayoría de empresas existentes en el país. En la etapa de simulación se opta por Jaamsim, un potente simulador con interfaz gráfica amigable al programador, desarrollado en Java. Para la construcción del algoritmo se ha optado por el mismo lenguaje de programación. La investigación proporciona un referente para la aplicación de una herramienta que permita mejorar la productividad y competitividad de las empresas en el mercado, concentrada en el análisis de los recursos, la planificación y la observación en el comportamiento de un sistema de manufactura flexible con el fin de evaluar los tiempos acumulados en sus procesos.

Palabras Clave:

- **SISTEMA DE MANUFACTURA FLEXIBLE**
- **ALGORITMO GENÉTICO**
- **MÉTODOS META HEURÍSTICOS**
- **SOFTWARE LIBRE**

Abstract

This research develops the integration between the simulation of a flexible manufacturing system and the optimization of one of its resulting parameters by applying a metaheuristic method based on genetic algorithms. The study of the project, focuses on the application of open source software for the execution of this purpose, because the costs in the implementation of this integration with commercial software are high for the majority of existing companies in the country. In the simulation stage, Jaamsim is chosen, a powerful simulator with a programmer-friendly graphical interface, developed in Java. For the construction of the algorithm, the same programming language was chosen. The research provides a reference for the application of a tool that improves the productivity and competitiveness of companies in the market, focused on the analysis of resources, planning and observation in the behavior of a flexible manufacturing system in order to evaluate the accumulated times in their processes

Keywords:

- **FLEXIBLE MANUFACTURING SYSTEM**
- **GENETIC ALGORITHM**
- **METAHEURISTIC METHODS**
- **OPEN SOURCE SOFTWARE**

CAPITULO 1

INTRODUCCIÓN

1.1. Problema

El siguiente capítulo presenta de manera general el problema para la optimización de tiempos de producción en los sistemas de manufactura flexible basado en métodos meta heurísticos usando software libre.

1.1.1. Desarrollo del problema

En el mundo donde índices de competitividad crecen cada día, un requerimiento primordial es la correcta planificación de los procesos por medio de la optimización esto con el fin de un aumento en la productividad. Para los sistemas de manufactura, además de una buena planificación, es requisito que sean flexibles para adaptarse a las necesidades del mercado; solo las empresas con un excelente manejo de sus procesos pueden mantenerse líderes en sus sectores de producción.

Dentro de las investigaciones analizadas previamente como el *Desarrollo de un software académico para programar la producción en los sistemas de manufactura flexible* por Flores K., 2014, la *Reconfiguración autónoma de sistemas de manufactura mediante la optimización de funciones de desempeño del proceso,* según Valtierra, 2014 y *A two-phase iterative heuristic approach for the Production Routing Problem realizado por* Absi, 2014*,* han identificado que existen tres puntos a considerar para sobrellevar lo anterior expuesto desde la visión técnica de la manufactura:

Primero: el problema radica en que cada caso de manufactura genera sus propias condiciones de producción, esto a su vez conlleva una planificación y flexibilidad únicas para el o los procesos de manufactura desarrollados dentro de una empresa.

Segundo: en la actualidad existen paquetes de software que simulan los entornos de producción conocidos como software de eventos discretos, sin embargo, en su mayoría son de paga y con costes elevados de implementación, poco rentables para la mayoría de empresas en el Ecuador, donde el 89.6% son microempresas, el 8.2% pequeñas, el 1.7% medianas y el 0.5% grandes empresas. Si bien, existen también opciones de Código Abierto o Software Libre de este tipo de programas no están enfocados a un usuario promedio, ya que requiere conocimientos medios a altos de programación e ingeniería en procesos de manufactura. Así mismo, el esporádico uso de este tipo de software y/o capacitación del personal a cargo no justifica los costes generados durante la implementación para las empresas locales a las cuales está enfocado el proyecto.

Tercero: la elección de los algoritmos a implementar para tratar de abarcar varias soluciones a los casos expuestos ya que a nivel de programación de un algoritmo heurístico de optimización existen varios métodos entre ellos los que se puede citar: Búsqueda local, Inteligencia de enjambre (Swarm Intelligence), Procedimiento de búsqueda adaptativa aleatoria codiciosa (Greedy Randomized Adaptative Search procedure GRASP), Búsqueda Reactiva (Reactive Search), Redes Neuronales (Neuronal Networks), Hacia atrás / Hacia adelante (Forward Heuristic BFH), Combinación Integral de Programación Lineal (Mixed Integer Linear Programming Formulation MILPF), Recocido Simulado (Simulated Annealing), Búsqueda Tabú (Tabu Search), Redes de Petri (Petri Networks), se debe considerar que las últimas investigaciones se basaron en los seis últimos algoritmos citados.

1.1.2. Planteamiento del problema

El proyecto analiza la factibilidad de programar la solución mediante implementación de uno o varios algoritmos enfocándose principalmente en métodos meta heurísticos sobre una plataforma de código abierto que permite la simulación grafica de sistemas de manufactura flexible dirigido al usuario final, a su vez el software permitirá presentar una interfaz amigable con el usuario disminuyendo el nivel de complejidad y el tiempo al momento de modelar el sistema.

La investigación de Valtierra, 2014 ofrece una metodología para optimizar los procesos de producción a través de métodos meta heurísticos basado en los factores críticos de la planeación y del control de la producción dentro de los sistemas de manufactura como: la selección óptima de las rutas de procesos y secuencias óptimas, otra investigación dada por Flores en el 2014 determina que se debe balancear la carga tanto en los lotes como en cada máquina disponible para la producción, garantizando un ahorro y facilitando la flexibilidad respondiendo a imprevistos de cambios de número de piezas, máquinas, herramientas entre otros, enfocándose en la disminución del tiempo de producción basado en el algoritmo heurístico de Greddy que resuelve problemas de selección de partes, formación de lotes y carga.

1.1.3. Formulación del problema a resolver

1. ¿Se podría optimizar cualquier software de código abierto para simulación de sistema de manufactura automatizados?

2. ¿Se podría usar los métodos meta heurísticos para optimizar software de simulación?

3. ¿Con métodos meta heurísticos se podría minimizar los tiempos de producción según las necesidades del usuario?

Así, el problema científico de la presente investigación es:

¿Cómo optimizar los tiempos de producción de los Sistemas de Manufactura Automatizados mediante métodos meta heurísticos, al considerar los criterios de factores críticos en las operaciones de manufactura, contribuyendo con una disminución de la competitividad en el mercado ecuatoriano?

Como Objetivo de la Investigación queda definido los Sistema de Manufactura Automatizados y su Campo de Acción, la optimización de las operaciones de producción de los sistemas de manufactura automatizados mediante métodos meta heurísticos.

1.2. Justificación e Importancia

Hay varias investigaciones que tratan sobre la Optimización de la Producción de Sistemas de Manufactura Automatizados basados en métodos meta heurísticos como la investigación de Frank M., 2014 con su tema “Reducing computation time in simulation-based optimization of manufacturing systems”, en el cual se realiza una análisis de los sistemas de producción buscando la mejor configuración de los parámetros posibles para un modelo de simulación con técnicas combinatorias de simulación como algoritmos heurísticos y genéticos determinando soluciones óptimas, otra investigación generada por el autor Chen J., 2013, con el tema “A study of a heuristic capacity planning algorithm for weapon production system”, habla sobre la planificación de la capacidad de producción de un sistema de producción de armas basado en algoritmos heurísticos programados en Visual Basic, que controlan el tiempo apropiado de la liberación de maquinaria dentro de la fábrica.

Se busca generar un código basado en métodos meta heurísticos que permita la adaptabilidad de los procesos de una empresa, optimizando las herramientas de análisis, diseño

y modelación para Sistemas de Manufactura, enfocándose en complementar un software de código abierto destinado a empresas manufactureras ecuatorianas en el rango de microempresas a medianas empresas que no pueden invertir en software con costos muy altos y licencias limitadas sobre los $20.000 USD aproximadamente.

Este proyecto pretende aportar a la comunidad de software libre y la tendencia mundial que genera aplicaciones tipo Open Source una herramienta destinada al usuario final con costes bajos de implementación para las microempresas y medianas empresas de fácil uso, que ayudará a disminuir los tiempos de producción, optimizando las líneas de producción, número de lotes, secuencias y rutas de fabricación, además de poder ampliarse en futuras investigaciones.

1.3. Objetivos

En el siguiente contenido se detalla tanto el objetivo general como los objetivos específicos de la presente investigación.

1.3.1. Objetivo general:

- Implementar un método meta heurístico de optimización aplicado a sistemas de manufactura automatizados sobre una base de software libre.

1.3.2. Objetivos específicos:

- Investigar las características de métodos heurísticos de optimización y su factibilidad de aplicación a los sistemas de manufactura automatizados.
- Analizar ventajas, limitantes técnicas y operativas del método heurístico elegido durante la investigación

- Diseñar un método heurístico que se acople a un simulador de código abierto que pueda optimizar los procesos de producción de los sistemas de manufactura automatizados.
- Obtener un modelo en el simulador de código abierto adaptado con métodos Heurísticos y comparar contra la metodología de "Programación adaptativa de sistemas de fabricación flexibles con restricciones de manejo de materiales utilizando algoritmos genéticos y redes coloreadas temporizadas de Petri" desarrollada por el tutor del proyecto de tesis.

1.4. Hipótesis

Un software de código abierto puede ser modificado mediante métodos heurísticos brindando la posibilidad de cambiar su estructura para optimizar procesos de producción centrados en la disminución de los tiempos de fabricación.

CAPÍTULO 2

DESCRIPCIÓN DEL SISTEMA DE PRODUCCIÓN Y ALGORITMOS METAHEURÍSTICOS DE OPTIMIZACIÓN

2.1. Sistemas de Manufactura

En este apartado se fundamenta los contenidos referentes a los sistemas de manufactura flexible que están relacionados con los sistemas de producción

2.1.1. Definición

Un sistema de manufactura flexible o su equivalente en inglés flexible manufacture system obteniendo las siglas en inglés como FMS, se lo puede definir según Cruz J., 2004 como "un conjuto de herramientas que ofrecen a la empresa de transformación industrial diferentes alternativas para el incremento de la productividad y como consecuencia mejorar la competitividad de la empresa".

Técnicamente un FMS según el autor Bernal, M., 2015 "es un grupo de máquinas o estaciones de trabajo relacionadas entre sí y que realizan una tarea específica", el autor Groove, 1990 define a los FMS como "grupos de estaciones de trabajo interconectadas por medio de un sistema de control que tiene la capacidad de procesar diferentes órdenes simultáneamente bajo un código de control numérico por cada estación".

2.1.2. Tipos de sistemas de manufacturas

La investigación de (García & Escobar, 2018) nos indica que existen cuatro factores de clasificación para un sistema de manufactura:

1. Por el tipo de operación realizada
    - Operaciones de procesamiento versus operación de ensamble.
    - Tipo de procesamiento u operación de ensamble.
2. El número de estaciones de trabajo y el layout
    - Una estación versus más de una estación.
    - Para más de una estación, ruteado variable versus ruteado fijo.
3. El nivel de automatización
    - Una estación versus más de una estación.
    - Para más de una estación, ruteado variable versus ruteado fijo.
    - Estaciones de trabajo manual o semi-automático que requieren tiempo completo del operador versus completamente automatizada que requiere solamente atención periódica del operador.
4. La diversidad de las partes o productos
    - Todas las unidades de trabajo idénticas versus variaciones en las unidades de trabajo que requieren diferencias en procesamiento.

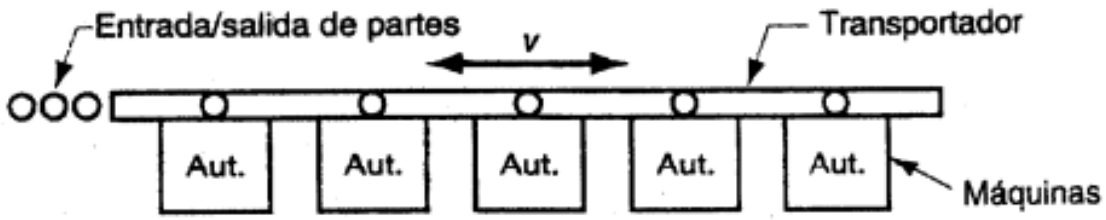
2.1.3. Tipos de distribución de FMS dentro de una planta

Los autores Alquilano & Graw, 2014 indican que para la elección de un tipo de distribución se debe tener en consideración la planeación, especificaciones de diseño, la programación y las estrategias de control de fabricación para operar el sistema, tomando en cuenta esos criterios establecen las siguientes distribuciones de planta:

a) En línea: usa un sistema de transferencia para mover las partes entre las estaciones de procesamiento, realizando carga y descarga, como se presentar en la Figura 1, existen movimiento bidireccional como un limitado flujo y secuencias básicas.

Figura 1

Distribución de FMS en línea

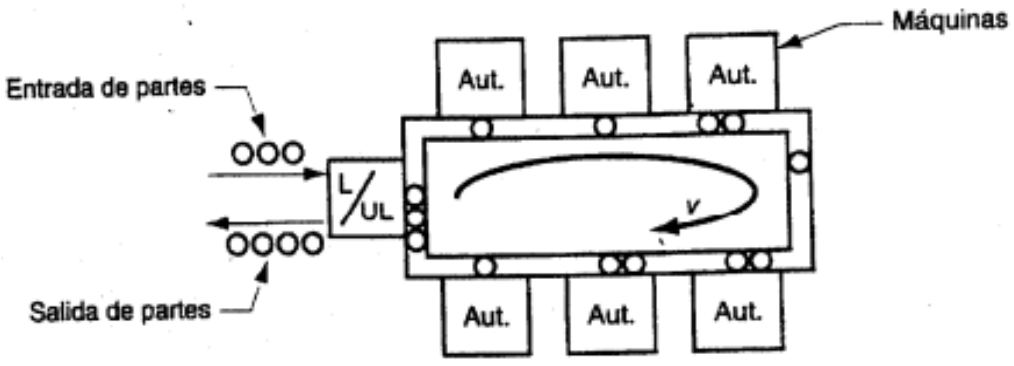


Nota: (Alquilano & Graw, 2014)

b) En ciclo: consiste en un trasportados con estaciones de trabajo ubicadas a su alrededor como se puede observar en la Figura 2., está configuración permite realizar diferentes secuencias de maquinado ya que se puede acceder a cualquier máquina desde cualquier otra.

Figura 2

Distribución de FMS en ciclo

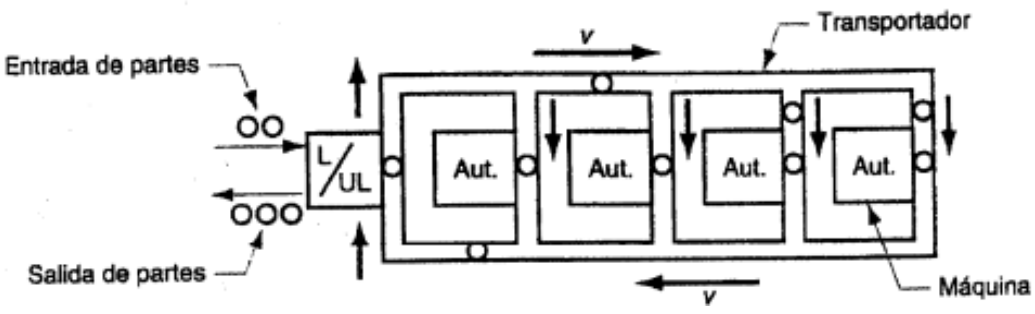


Nota: (Alquilano & Graw, 2014)

c) En escalera: en esta configuración se puede realizar secuencias de procesamiento, sus estaciones de trabajo se encuentran en peldaños con forma de escalera como se observa en la Figura 3.

Figura 3

Distribución FMS en escalera

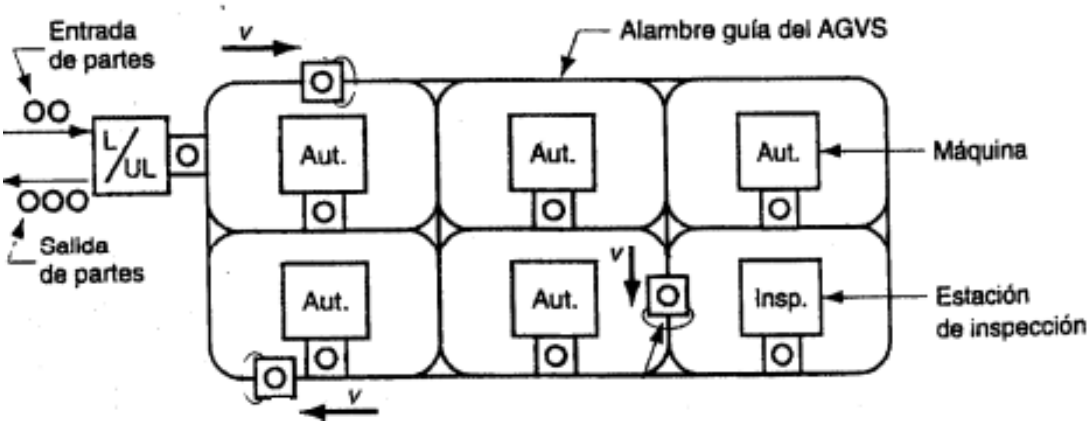


Nota: (Alquilano & Graw, 2014)

d) En campo abierto: trata sobre una configuración más compleja que consiste en varios ciclos anidados como se puede ver en la Figura 4.

Figura 4

Distribución FMS en campo abierto

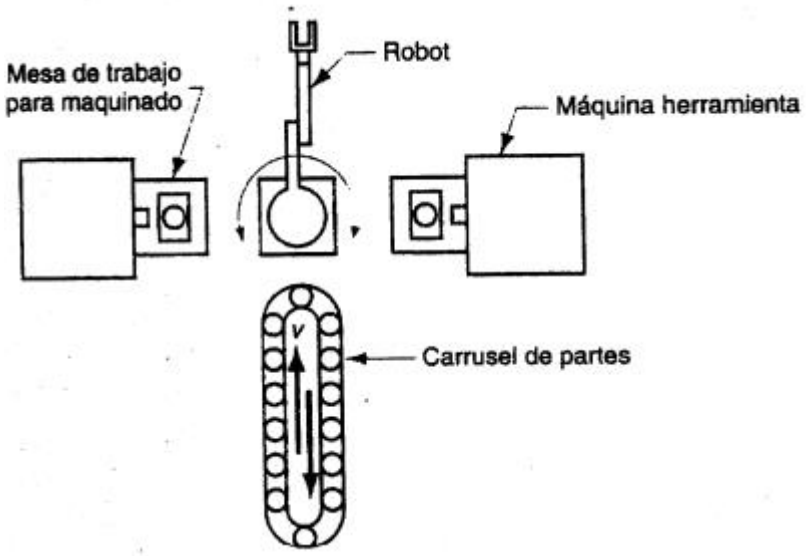


Nota: (Alquilano & Graw, 2014)

e) Centrada en un robot: un robot es el encargado de la carga y descarga de las máquinas en la celda, observar la Figura 5.

Figura 5

Distribución FMS centrado en un robot



Nota: (Alquilano & Graw, 2014)

Los sistemas de manufactura se pueden clasificar también de acuerdo a su espectro o rango de flexibilidad.

a) Espectro reducido: produce un número limitado de partes pequeñas de diferentes geometrías o diseños.

b) De alto espectro: producen familias de partes numerosas con variaciones sustanciales en la configuración de las partes y en la secuencia de operación.

2.1.4. Funcionamiento de los sistemas de manufactura

La funcionalidad de los sistemas de manufactura viene definida por los grupos de tecnología de manufactura, manufactura integrada por computador (CIM) y robots, el proceso de funcionamiento inicial con la carga de la materia prima y las herramientas en la maquinaria, el sistema de cómputo inicializa las máquinas con los programas de control numérico, de acuerdo a las tareas en la planta y ordena el proceso de transporte de la materia hacia cada proceso, una vez finalizada la operación designada se transporta devuelta la pieza (materia prima transformada) a la banda transportadora automáticamente a otra sección de trabajo hasta tener un producto terminado.

Un factor que caracteriza a un sistema de manufactura es el grado en el cual es capaz de lidiar con variaciones en las partes o productos que produce. Las posibles variaciones para un sistema de manufactura son:

- Variaciones en el tipo y/o color de las partes plásticas moldeadas en un inyector.
- Variaciones en componentes electrónicos colocados sobre un circuito impreso estándar.

- Variaciones en el tamaño de circuitos impresos manejados por una máquina colocadora de componentes.
- Variaciones en la geometría de partes de máquinas.
- Variaciones en partes y opciones en un producto ensamblado sobre una línea final ensamble.

2.2. Sistemas de producción

Este punto se tratará sobre definiciones relacionadas con los sistemas de producción.

2.2.1. Definición

Los sistemas de producción se definen como la actividad o conjunto de actividades que se realizan para producir y distribuir un producto o servicio, se compone de entradas (insumos o materia prima), proceso de conversión (transforma la materia prima en productos para el mercado) y salidas (producto para consumo intermedio o producto final). Celeberrima, 2019.

Los sistemas de producción son diseñados y mejorados para optimizar utilidades del productor, lograr la satisfacción de los clientes y mejorar las condiciones laborales.

2.2.2. Tipos de procesos productivos

Según EAE Business School, 2018 nos indican que, para aprovechar los recursos, flexibilidad para responder a los cambios y el trabajo en condiciones de calidad existen cuatro procesos productivos que se detallan a continuación:

- **Producción por trabajo:** también llamada producción bajo pedido, tiene la finalidad de concentrar todos los esfuerzos en un solo producto a la vez, cada resultado es diferente ya que se trata de un concepto asociado a un uso intensivo

de mano de obra. Los procesos pueden ser combinados entre métodos manuales y mecánicos

- **Producción por lotes:** está producción se basa en la producción de una pequeña cantidad de productos idénticos y limitada, su modalidad puede ser intensiva en mano de obra, aunque generalmente se introduce el concepto de plantillas o modelos, que agilizan la producción, reducen el factor de personalización. Este tipo de producción produce más dificultades a la hora de organizar el funcionamiento
- **Producción en masa:** se diferencia por la producción de cientos de productos idénticos, por lo general en una línea de producción cuenta con un sistema automatizado de tareas que permite dar una salida a un volumen de productos más elevado utilizando menos trabajadores, está opción implica el montaje de un número indeterminado de componentes individuales, piezas requeridas de un proveedor.
- **Producción de flujo continuo:** la diferencia entre ésta y la producción en masas es que, en este caso, la línea de producción se mantiene en funcionamiento los siete días de la semana a 24 horas. Consiguiendo maximizar la producción, reduciendo costos de inicio y paro de procesos. Se debe considerar que está producción cuenta con procesos altamente automatizados y en la que se requiere de menos trabajadores, con productos con menos fallas.

2.2.3. Factores que integran un sistema de producción

Según la investigación de (Romero, 2016) indica que para garantizar el éxito de un sistema de produccón deben ser gestionados cinco elementos, que son conocidos en el área de la ingeniera como: mano de obra, maquinarias, materiales, métodos y mediciones, en la prática

estos elementos presentan gran incertidumbre haciendo complejo y dificil gestionar un sistema de producción.

- **Mano de obra:** es el recurso más valioso dentro de un sistema de producción, forma parte activa del proceso de fabricación, contirbuye a optimizar el uso de recursos materiales y técnicos.
- **Maquinarias y equipos:** son las herramientas empleadas por los operarios para lograr la transformación de las materias primas, el correcto funcionamiento está sujeto a la aparición de desperfectos que no pueden ser corregidos con el mantenimiento preventivo.
- **Materiales:** se hace referencia tanto a la materia prima como a insumos indirectos y productos terminados, los fallos en relación al abastecimiento de materiales en los sistemas de producción ocasionan altos costos.
- **Método:** son las secuencias detalladas de los procesos y la ruta de operaciones que deben seguirse dentro de los sistemas de producción para garantizar la fabricación de los productos terminados, las operaciones son divididas en una serie de tareas o actividades que deben ser realizadas para culminar satisfactoriamente la operación.
- **Mediciones:** en todo sistema de producción se realizan mediciones para determinar si las materias primas y los insumos satisfacen los requirimientos de calidad, adicionalmente, también se realizan mediciones a la producción en proceso para controlar que sean respetados los rangos de tolerancia admitidos en las diferentes etapas del sistema de producción.

2.3. Métodos heurísticos o de optimización

Según Caldas et. al., 2014, los“métodos herurísticos permiten obtener soluciones de alta calidad con un coste computacional razonable”.

Los métodos de optimización sirven para intentar alcanzar la mejor solución posible en un tiempo razonable, ya sea para minimizar o maximizar la solución, con ayuda de una función que evalúa la calidad de la solución en cada momento. (Rabanal, Rodríguez, & Rubio, 2010)

Los métodos meta heurísticos se aplican cuando las soluciones son muy complejas, con espacios de búsqueda muy grandes y no se puede realizar una exploración sistemática del sistema para hallar la mejor solución, por lo cual el método elimina las rutas de búsqueda que generen la peor solución basadas en un punto inicial definido por el usuario, a continuación se enlista los métodos heurísticos o de optimización basados en sus principales características como nos indican los autores Rabanal, Rodríguez, & Rubio, 2010

2.3.1. Escalada (Hill climbing)

Método conocido en inglés como hill climbing, el cual consiste en aplicar técnicas de mejora iterativa para solucionar problemas de espacio de estado, el método escoge a cada paso el sucesor con mejor valor en función del estado inicial seleccionado, existen dos tipos de algoritmos de escalada:

- **Escalada simple:** se busca aplicar a la solución actual un operador de referencia que se compare con los operadores disponibles, reemplazando el valor de referencia y desechando la solución anterior, conservando la nueva.
- **Escalada por máxima pendiente:** se aplican a la solución actual todos los operadores posibles y se actualiza la solución con el valor que mejore a la solución actual después de

estudiar todos los resultados, se diferencia de la simple ya que cuando ésta determina una mejor solución deja de probar el resto.

Las ventajas que presenta este método son:

- Es muy eficiente en tiempo
- Consigue ahorro de memoria

La desventaja más significativa que presenta el método es que no siempre determina la solución óptima.

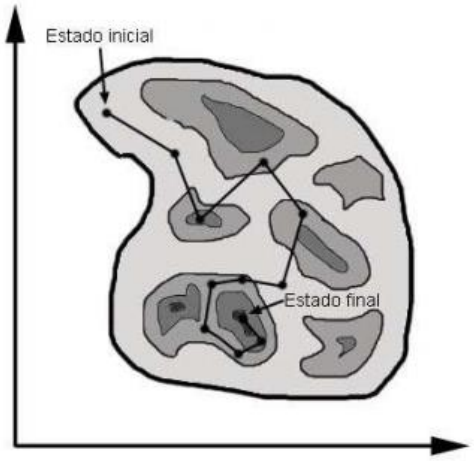
2.3.2. Enfriamiento simulado (Simulated annealing)

Método con su nombre en inglés como simulated annealing, consiste en elegir un sucesor de entre todos los posibles según una distribución de probabilidad, dicha elección permite empeorar la solución actual, es decir, puede escogerse (probabilísticamente) estados peores, de este modo se dan pasos parcialmente aleatorios por el espacio de soluciones buscando la mejor solución, esto permite al algoritmo poder salir de óptimos locales.

En la figura 6 se puede observar cómo en ciertos puntos de la búsqueda se elige una solución peor que la actual para poder huir de los máximos locales y poder encontrar el máximo global (estado final), las regiones más oscuras representan soluciones mejores, mientras que las regiones más claras representan soluciones peores.

Figura 6

Enfriamiento simulado



Nota: (Rabanal, Rodríguez, & Rubio, 2010)

2.3.3. Optimización basada en nubes de partículas (Particle swarm optimization)

Este tipo de método se lo conoce en inglés como particule swarm optimization, inspirado en el comportamiento social del vuelo de bandadas de aves y el movimiento de banco de peces, estos sistemas se componen de varias partículas que se mueven por el espacio de búsqueda durante la ejecución de algoritmo, se compone de entidades simples con acciones locales (incluyendo interacciones con el ambiente). El resultado de la combinación de comportamientos simples es la aparición de comportamientos complejos y la capacidad de conseguir buenos resultados como un equipo.

Las técnicas de nubes de partículas con inteligencia son métodos estocásticos basados en población usados en problemas de optimización combinatoria, donde el comportamiento colectivo de los individuos surge de sus interacciones locales con el medio para generar la aparición de patrones globales funcionales, este método se aplica típicamente en problemas de optimización numérica y se le atribuyen las siguientes características:

- Asume un intercambio de información entre los agentes de búsqueda, lo que se identifica con las interacciones sociales de aves o peces.

- La idea básica es utilizar la información del mejor resultado propio de cada partícula y la información del mejor resultado global.

- La implementación es muy sencilla ya que se manejan pocos parámetros

- La convergencia del algoritmo es rápida y se encuentra buenas soluciones.

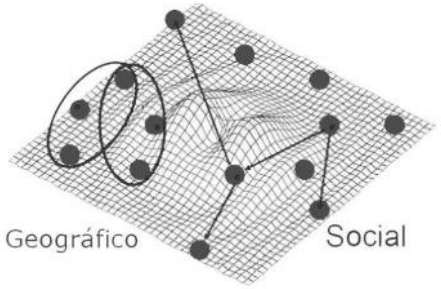
Las nubes de partículas pueden formar distintas topologías, que definen el entorno de cada partícula individual y se diferencias en dos tipos como se observa en la figura 7:

- Geográfico: se calcula la distancia de la partícula actual al resto y se toman la más cercana para componer su entorno.

- Sociales: se tiene una prioridad basado en una lista de partículas, independientemente de su posición en el espacio.

Figura 7

Tipos de entornos



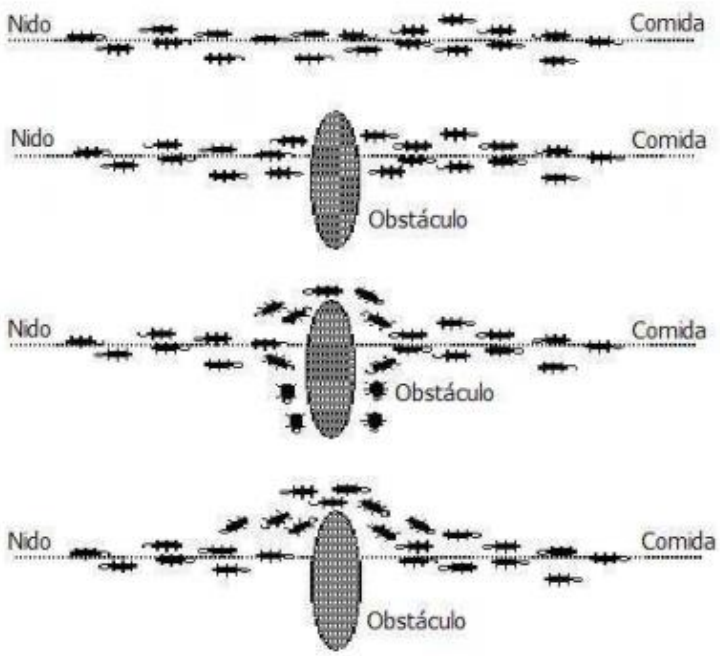
Nota: (Rabanal, Rodríguez, & Rubio, 2010)

2.3.4. Algoritmos de colonias de hormigas (Ant colony optimization)

Método conocido en inglés como ant colony optimization, es una técnica probabilística para resolver problemas computacionales que pueden reducir al problema de encontrar buenos caminos a través de grafos, inspirados en el comportamiento de las hormigas, que son insectos sociales que viven en colonias dirigido al desarrollo grupal como un todo más que a un desarrollo individual.

Figura 8

Comportamiento de hormigas



Nota: (Rabanal, Rodríguez, & Rubio, 2010)

El comportamiento de las hormigas funciona según el principio de que cada hormiga se mueve de manera aleatoria; depositan feromonas en el camino que recorren las hormigas detectan el camino principal (el que tienen más feromonas) inclinándose a seguirlo, en la figura

8, se puede observar como todas las hormigas de una colonia tienden a seguir el mismo camino para ir desde su nido hasta la comida, considerando que si existe un obstáculo buscarán por donde evitarlo rápidamente y finalmente seguirán la ruta más corta, por el cual pasarán las hormigas más rápido y más veces dejando un rastro de feromonas más fuerte..

2.3.5. Algoritmos genéticos (Genetic algorithms)

El autor Gestal, R, et al., 2010, nos indica que los algoritmos genéticos son métodos adaptativos, generalmente usados en problemas de búsqueda y optimización de parámetros, basados en la reproducción sexual y en el principio de supervivencia del más apto entre secuencias con intercambio de información estructurado, aunque aleatorio, para construir así un algoritmo de búsqueda.

Este tipo de método consiste en realizar una escalada en paralelo inspirada en los siguientes mecanismos de selección natural, como se puede observar en la figura 7:

- La adaptación de los seres vivos al entorno en el que viven.

- La supervivencia, reproducción y sus posibilidades según las características de cada individuo.

- La combinación de individuos, que pueden llevar a individuos mejor adaptados.

Las analogías de la selección natural con la búsqueda local son las siguientes:

- Las posibles soluciones se representan como los seres vivos, es decir, como los individuos que se adaptan al medio.

- La función que evalúa la calidad de cada solución calcula la adaptación de cada individuo.

- Basándose en la reproducción de los individuos, se supone que combinando buenas soluciones se puede obtener soluciones mejores.

- Apoyándose en el mecanismo de evolución, se selecciona las mejores soluciones sucesivamente.

Los parámetros del algoritmo genético son:

- El número total de individuos de la población (que es constante en cada generación)

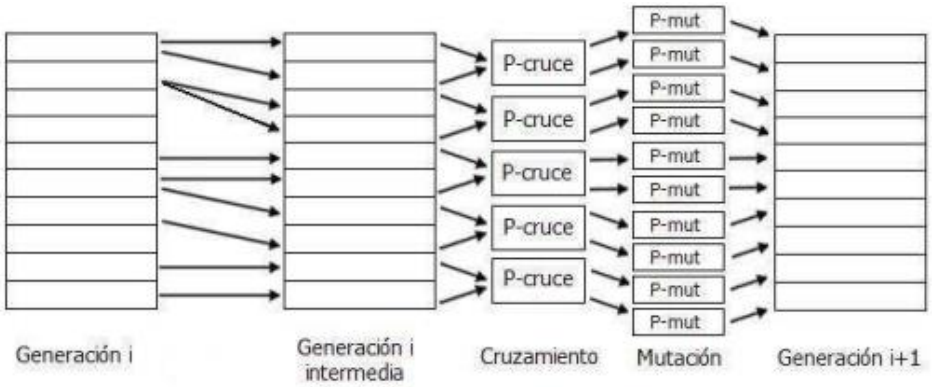
- La proporción del total de los individuos que intervienen en la reproducción en cada generación, es decir, la proporción de individuos que serán padres.

- La proporción del total de los padres que se escogen de entre los mejores individuos de la población, es decir, la proporción de padres que se eligen por su nivel dentro de la población.

- La probabilidad de que ocurra una mutación, que generalmente es un número muy bajo.

Figura 9

Pasos de un algoritmo genético

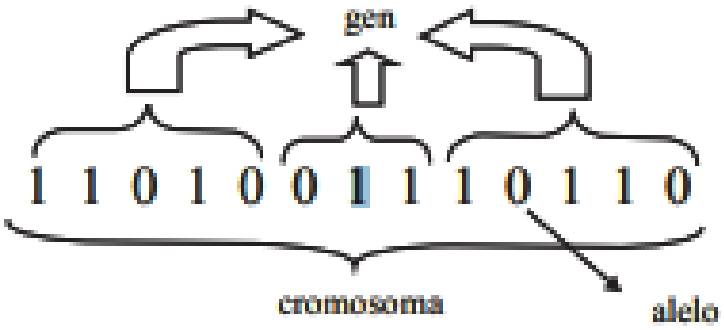


Nota: Rabanal, Rodríguez, & Rubio, 2010

2.3.5.1. Codificación de problemas. Cualquier solución potencial a un problema puede ser presentada dando valores a una serie de parámetros (genes: conjunto de parámetros) y se codifica en una cadena de valores denominada cromosoma como se observa en la Figura 10.

Figura 10

Individuo genético binario

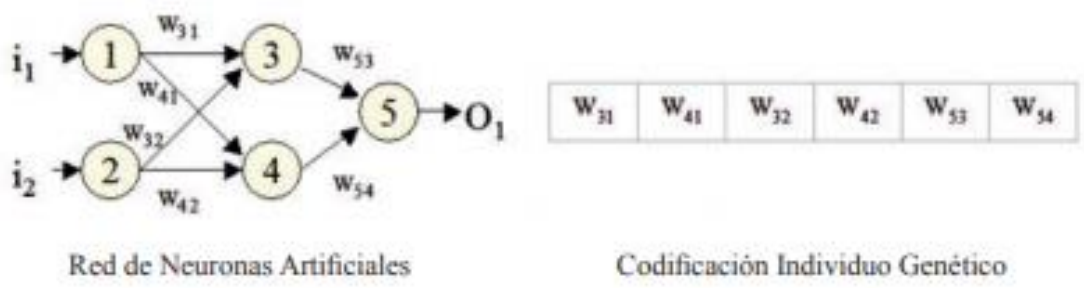


Nota: (Gestal, R, et al., 2010)

Los algoritmos genéticos se basan en la representación y aplicación de redes neuronales artificiales como se observa en la Figura 11, codificar una red neuronal es tan sencillo como asignar un gen del cromosoma a cada uno de los pesos de la red, también se puede añadir genes que indiquen el número de capas y el número de elementos de procesado en cada una.

Figura 11

Codificación de una red de neuronas artificiales



Red de Neuronas Artificiales

Codificación Individuo Genético

Nota: (Gestal, R, et al., 2010)

2.3.5.2. Estructura de un algoritmo. Según el autor Gestal, R. 2010, los algoritmos genéticos trabajan sobre una población de individuos, cada uno de ellos representa una posible solución al problema que se desea resolver. Una generación se obtiene a partir de la anterior por medio de los operadores de reproducción como:

- **Cruce:** se trata de una reproducción de tipo sexual, se genera una descendencia a partir del mismo número de individuos (generalmente 2) de la generación anterior.
- **Copia:** se trata de una reproducción asexual, un determinado número de individuos pasa sin sufrir ninguna variación directamente a la siguiente generación.

CAPITULO 3

DEFINICIÓN DE LA SOLUCIÓN

3.1. Problema

En este capítulo, se detalla los problemas en la Simulación de Sistemas de Eventos Discretos centrados en los Sistemas Flexibles de Manufactura Automatizados y su Optimización en los tiempos de producción con la aplicación de un algoritmo meta heurístico.

Para el presente proyecto se utilizará software libre como herramienta para encontrar una solución y considera los siguientes aspectos:

- El desarrollo de una estación FMS; el diseño, planificación, simulación y ejecución, se contemplan varios componentes, por ejemplo: máquinas, robots, vehículos automatizados, entradas, salidas; que configurados en conjunto, deben fabricar productos a través de secuencias determinadas en escenarios complejos, razón por lo cual obtener una herramienta de simulación gráfica que considere la toma de decisiones en la ejecución de todas las operaciones, en una forma correcta y entendible al usuario final, es un punto que se debe tener en cuenta.
- El software de simulación debe presentar las siguientes características: flexibilidad, facilidad de uso e integración, coste razonable, capacidad de animación y depuración, poseer capacidades estadísticas y de manejo de datos, contar con documentación y soporte.
- El resultado de la simulación de un sistema de eventos discretos, deberá ser evaluado mediante la aplicación del algoritmo meta heurístico, que optimice la producción de los productos generados.

Dentro de las opciones investigadas para la simulación en software libre, la que mejor se ajusta a los requerimientos que se han especificado, es Jaamsim, (Java Animation Modelling and Simulation) un software desarrollado por la empresa Ausenco; dirigido a una amplia audiencia de profesionales, investigadores y estudiantes. Es multiplataforma y su ejecutable, el manual del usuario, el manual de programación y los ejemplos están disponibles a solicitud de los autores.

3.2. Introducción a Jaamsim

Los autores King & Harrison, 2013 nos indican que Jaamsim es un paquete gratuito de simulación de código abierto escrito en el lenguaje de programación Java. Se proporciona, con una moderna interfaz gráfica para el usuario (GUI), que es comparable al software comercial; incluye opciones de arrastrar y soltar (Drag & Drop), construcción de modelos, un editor de entrada, visor de salida y gráficos en 3D. Los usuarios pueden crear sus propias paletas de objetos de alto nivel utilizando Java estándar y herramientas de programación modernas como Eclipse.

Jaamsim no es la primera oferta de simulación de código abierto en Java. Los motores de simulación de código abierto anteriores en Java incluyen: CSIM, DESMO-J, DEUS, DSOL, JavaSim, JiST, Jsim, JSL, SimJava, Simkit, SSJ y Tortuga. Aunque estos paquetes ofrecen muchas características útiles que los hacen buenas opciones para algunas aplicaciones, cada uno proporciona solo una biblioteca de herramientas de simulación y el usuario puede preparar su modelo a partir del código Java. Sin una GUI, estos paquetes atraen solo a personas con el tiempo y las habilidades de programación necesarias para construir modelos de esta manera. Una excepción a esta observación es Simkit, que proporciona el generador de gráficos de eventos visuales VisSim; sin embargo, esta GUI se limita a construir modelos a partir de componentes de

gráficos de eventos de bajo nivel, que es solo una parte de la funcionalidad necesaria para que el software tenga un atractivo amplio.

Finalmente, los autores desean dejar completamente claro que están ofreciendo Jaamsim como un proyecto genuino de código abierto para el beneficio de la comunidad de simulación. Es un software gratuito licenciado bajo GPLv.3. No tienen planes de presentar una versión "premium" es decir de paga de Jaamsim.

3.2.1. Requisitos de instalación

Los requisitos para su instalación acorde al manual de usuario desarrollado por King & Harrison, 2017 son:

- El ordenador debe tener instalada una máquina virtual JAVA 7 o superior. Disponible en la página oficial de java(http://www.java.com), para sistemas operativos Windows, Linux y OS X.
- La tarjeta gráfica debe cumplir con la norma OpenGL 3.0 o superior para que el renderizado de los modelos en 2 y 3 dimensiones presenten fluidez. Puede funcionar también con OpenGL 2.1 aunque su rendimiento. Se recomienda una tarjeta gráfica NVIDIA GeForce. aunque es posible usar una computadora con una tarjeta gráfica AMD, las tarjetas NVIDIA generalmente brindan un mejor soporte para gráficos OpenGL.
- Contar con procesador Intel Core i3, i5 e i7 (mínimo de segunda generación) o posterior

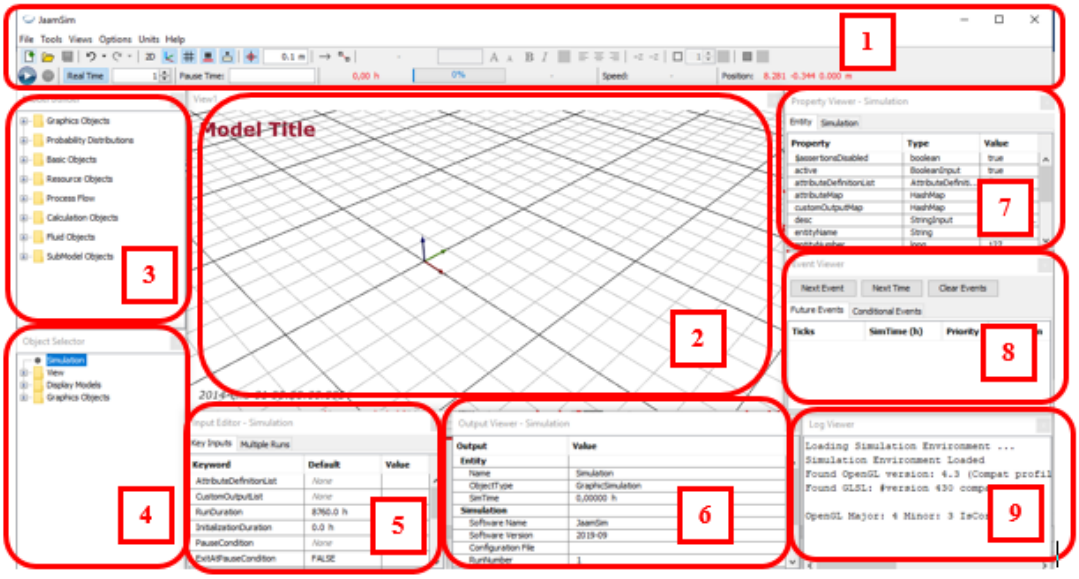
3.2.2. Presentación de Jaamsim

La introducción en el uso de Jaamsim según lo descrito por King & Harrison, 2017; detalla que la interfaz gráfica de usuario (GUI) proporciona todas las herramientas necesarias para la

construcción de modelos. La interfaz de inicio puede es expuesta de la siguiente manera (ver Figura 12):

Figura 12

Interfaz inicio Jaamsim



1. **Panel de control (Control Panel):** La interfaz principal de Jaamsim que controla la ejecución de modelos y proporciona acceso a los otros componentes de la GUI.
2. **Ventana de vista (View Window):** Una o más ventanas que muestran vistas en 3D del universo modelo.
3. **Constructor de Modelos (Model Builder):** Herramienta para arrastrar y soltar componentes del modelo. Proporciona bloques configurables de: herramientas gráficas, distribuciones de probabilidad, flujo de proceso, recursos, sistemas de matemáticas y control, flujo de fluidos.
4. **Selector de objetos (Object Selector):** Proporciona acceso a cada objeto en el modelo.

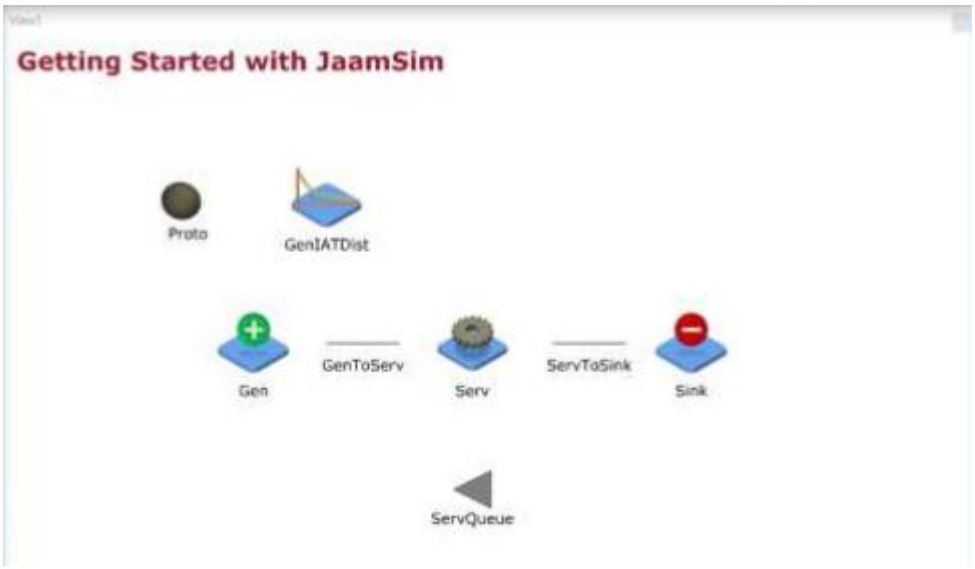
5. **Editor de entrada (Input Editor):** Muestra las entradas del objeto seleccionado y permite la edición.
6. **Visor de salida (Output Viewer):** Muestra el valor actual de las salidas del objeto seleccionado.
7. **Visor de propiedades (Property Viewer):** Herramienta de depuración que muestra todas las propiedades internas del objeto seleccionado.
8. **Visor de eventos (Event Viewer):** Herramienta de depuración que muestra los avances en la simulación por evento o por siguiente avance en el tiempo.
9. **Visor de Log (Log Viewer):** Herramienta que permite visualizar la ejecución de las instrucciones.

3.2.3. Guía para ejemplo básico de simulación

El modelo de ejemplo (Figura 13) se simula un sistema típico de colas de un solo servidor, con entidades siendo generado, procesado y consumido. El modelo terminado se muestra a continuación.

Figura 13

Ejemplo de simulación en Jaamsim



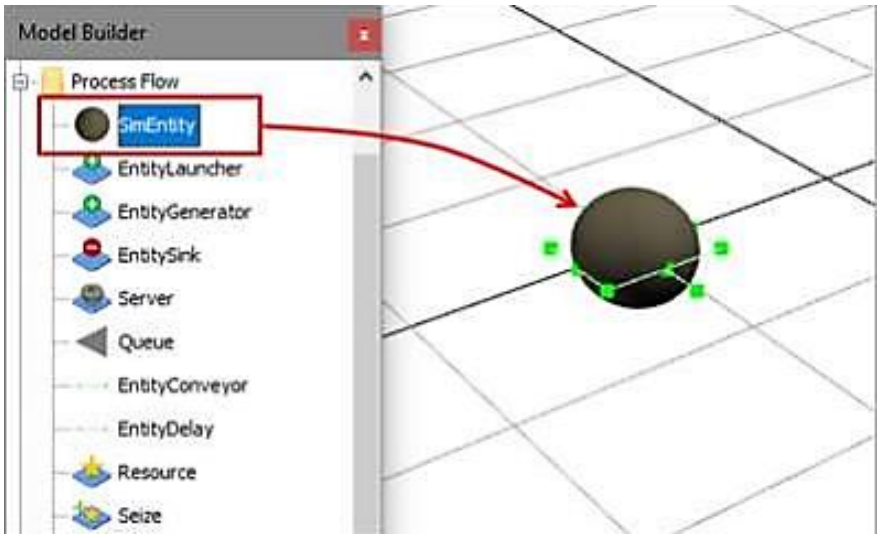
Este ejemplo se divide en cuatro pasos:

1. Creación de objetos modelo
2. Relación entre los objetos
3. Agregar una distribución de probabilidad al modelo

3.2.3.1. Creación de objetos modelo. En el constructor de modelos, se expande la paleta Flujo de proceso y luego arrastre y suelte una SimEntity en el Espacio de Trabajo (Figura 14). Esto crea un objeto SimEntity con un nombre predeterminado (SimEntity1) y forma (Esfera) que se selecciona automáticamente, denotado por el resaltado verde como se muestra a continuación.

Figura 14

SimEntity insertada en espacio de trabajo



Este objeto servirá como prototipo para las entidades que se procesarán en el modelo. En el objeto selector (Object Selector), seleccione SimEntity1 y presione F2 para cambiar el nombre del objeto como "Proto" (ver Tabla 1).

Tabla 1

Objeto SimEntity a crear

Constructor de modelo (Model Builder)	**Tipo de Objeto (Object Type)**	**Nombre (Name)**
Process Flow	Sim Entity	Proto

Como los gráficos del modelo serán 2D, a partir de la entidad Proto, con un click en el botón “2D”, en el panel de control (Control Panel), para que la vista se convierta en un ojo de pájaro (Vista superior). Ajuste la posición de visualización utilizando las acciones enumeradas abajo (ver Tabla 2).

Tabla 2

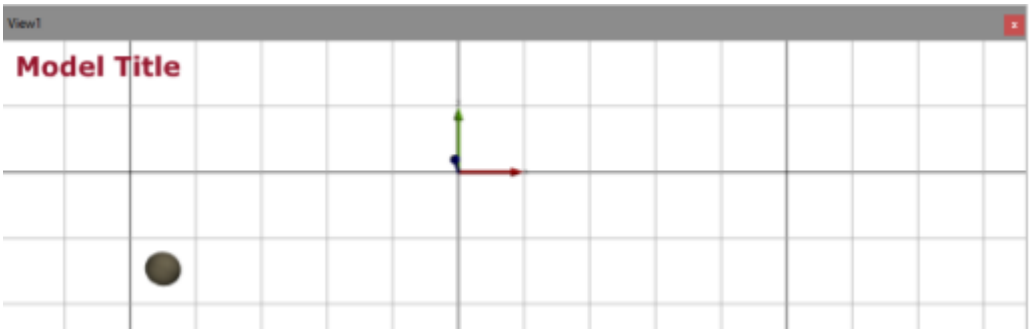
Ajustes de vista

Acción	Descripción
Click Izquierdo presionado	Desplazamiento (Pan) en el plano XY
Rueda de desplazamiento(ratón)	Zoom

Después de ajustar la posición de vista y el zoom, la ventana debería aparecer similar a la siguiente figura 15.

Figura 15

Vista de ventana configurada



En función de lo descrito anteriormente se crea los siguientes elementos que se puede observar en la Tabla 3.

Tabla 3

Objetos adicionales para modelos de ejemplo

Constructor de modelo (Model Builder)	**Tipo de Objeto (Object Type)**	**Nombre (Name)**
Process Flow	EntityGenerator	Gen
Process Flow	EntityConveyor	GenToServ
Process Flow	Server	Serv
Process Flow	EntityConveyor	ServToSink
Process Flow	EntitySink	Sink
Process Flow	Queue	ServQueue

A menudo es más fácil cambiar el nombre de un objeto activando su etiqueta, que luego puede editarse. Comience haciendo click derecho en el objeto y seleccionando "Mostrar etiqueta" (Show Label). Luego haga doble click en la etiqueta e ingrese el nuevo nombre. Cambiar la etiqueta de un objeto hace que se renombre como corresponde. El cambio de nombre se ejecuta con presionar el botón Enter o haciendo click en otro lugar de la ventana. La etiqueta se puede dejar en lugar o se puede desactivar haciendo clic derecho en el objeto y deseleccionando "Mostrar etiqueta".

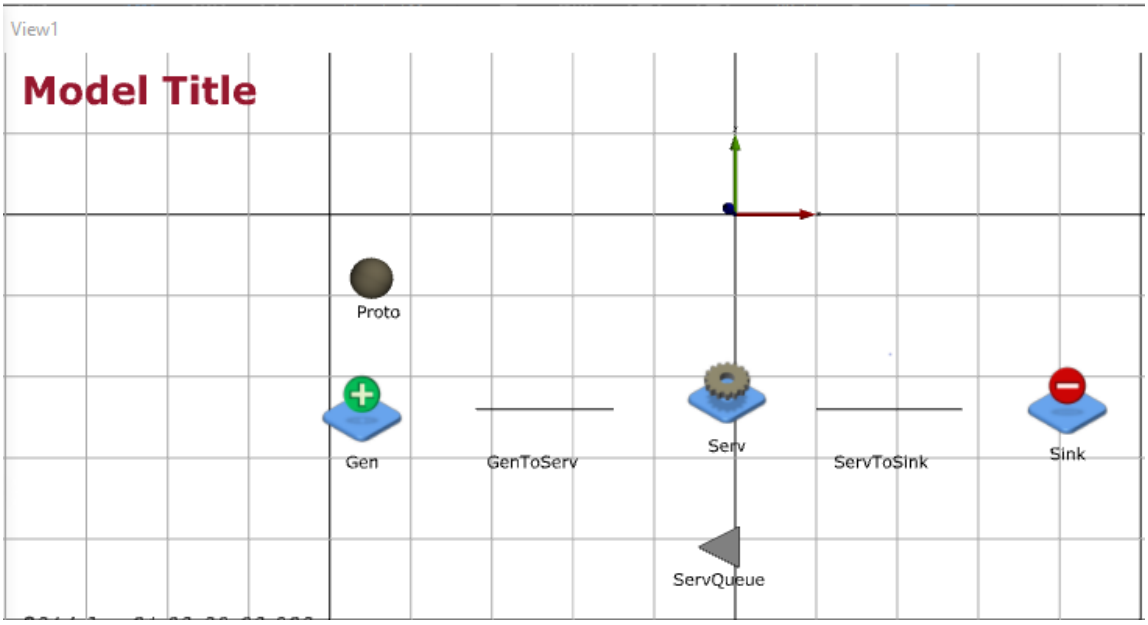
Un objeto se puede mover seleccionándolo y usando CTRL + Click izquierdo + Arrastrar. La posición de una etiqueta. relativo a su objeto se puede cambiar utilizando el mismo método. Coloque los primeros cinco objetos de izquierda a justo en el siguiente orden:

Gen - GenToServ - Serv - ServToSink – Sink

Se debe colocar el objeto ServQueue, debajo de Serv. Posterior a posicionar todos los objetos el modelo debería ser similar a la siguiente figura:

Figura 16

Vista final del paso 1



3.3. Algoritmos genéticos

Centrado en la investigación previa, se opta por el uso de un algoritmo genético para el desarrollo de una solución de optimización de secuencias de producción por lo cual se detalla en esta sección la metodología del funcionamiento de dicho algoritmo.

El autor Calderón, 2009, no indica que los algoritmos genéticos combinan la supervivencia de los individuos mejor adaptados junto con operadores de búsqueda genéticos como la mutación y el cruce, de ahí que sean comparables a una búsqueda biológica, además son mecanismos de búsqueda basados en las leyes de la selección natural y de la genética.

Características:

- Trabajan con un conjunto de puntos
- No están sujetos a restricciones
- Utilizan operadores probabilísticos
- Resultan fácil de utilizar en arquitecturas paralelas modernas
- Son algoritmos estocásticos es decir dos ejecuciones distintas pueden dar dos soluciones distintas.
- Utilizan una función objetivo que da la información de lo adaptados que están los individuos, y no las derivadas u otra información auxiliar.
- Las reglas de transición son probabilísticas, no determinísticas.

3.3.1. Proceso para crear un algoritmo genético

Los autores Caceres & Rivas, 2017 en su investigación nos detallan las secuencias que debe cumplir un algoritmo genético y las cuales se especifican a continuación:

1. Generar una población inicial
2. Iterar hasta un criterio de parada
3. Evolucionar cada individuo de la población
4. Seleccionar los progenitores
5. Aplicar el operador de cruce y mutación a esos progenitores

6. Incluir la nueva descendencia para formar la nueva generación

3.3.2. Población Inicial

Los algoritmos genéticos tienen varios modelos de producir una población inicial, pero para conseguir el máximo desempeño del algoritmo, la población debe ser producida aleatoriamente no obstante evolucionará para así obtener una buena solución.

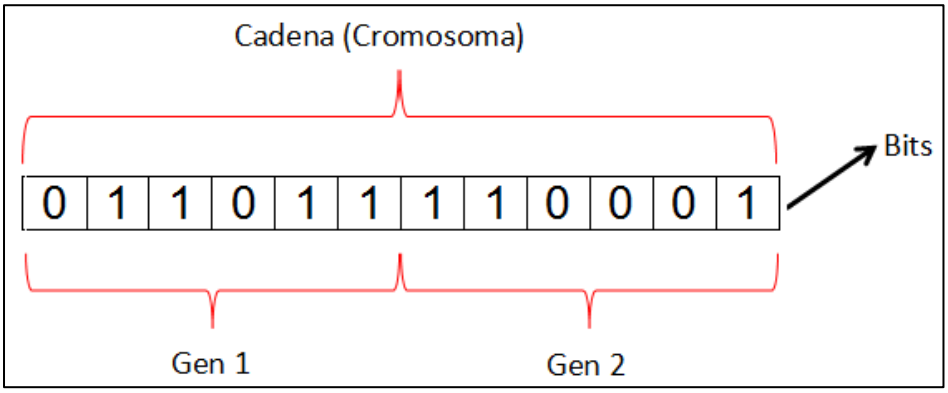
3.3.3. Codificación

Para poder emplear un algoritmo genético se tiene que definir el método de codificación para cada solución, las cadenas de bits con longitud fija son las que más se emplean al momento de desarrollar los algoritmos.

La codificación binaria del algoritmo genético de cada individuo es representada por una cadena de bits, (ver Figura 17). Las variables a resolver normalmente son un número entero o decimal, que se identifica con un gen. El conjunto de genes es representado por un cromosoma. Se debe de asignar una longitud a la cadena de bits de los individuos, para la solución esperada, por lo tanto, cada individuo debe tener una representación única.

Figura 17

Cadena cromosoma



3.3.4. Adaptabilidad

Es adaptación de un cromosoma que se basa en la probabilidad de que este sobreviva a la siguiente generación:

$$f_i = \frac{q_i}{\sum_j q_j} \qquad \textbf{Ecuación 1. Adaptabilidad}$$

Donde:

- El cromosoma es igual a f_i
- q_i representa la calidad de los candidatos
- $\sum_j q_j$ es la suma de todos los candidatos

Tabla 4

Comparación de cromosoma y adaptabilidad

	Cromosoma (f_i)	**Calidad (q_i)**	**Adaptabilidad estándar**
1	4	4	0.4
3	1	3	0.3

1	2	2		0.2
1	1	1		0.1

3.3.5. Selección de individuos

Dependiendo del orden de los cromosomas, se puede determinar individuos diferentes y por lo cual se puede calcular la adaptabilidad como se observa en tabla 5, mientras mayor sea su resultado mejor grado de adaptabilidad.

Tabla 5

Selección de individuos en base a adaptabilidad

	Número individuo	**Codificación**	**Número**	**X^2**	**Adaptabilidad** $f_i=q_i/\sum_j q_j$	**Pareja**
MEJOR INDIVIDUO	1	(0, 1, 1, 0, 0)	12	144	0,07399	6
	2	(1, 0, 0, 1, 0)	18	324	0,16649	3
	3	(0, 1, 1, 1, 1)	15	225	0,11562	2
	4	(1, 1, 0, 0, 0)	24	576	0,29599	5
	5	(1, 1, 0, 1, 0)	26	676	0,34737	4
	6	(0, 0, 0, 0, 1)	1	1	0,00051	1

			1946		

3.3.6. Torneo

Dentro de la población se rota a los individuos asignándole una pareja y el mejor de los dos es seleccionando según su grado de adaptabilidad, maximizando la función y pasando al proceso de copia.

Tabla 6

Re numeración según adaptabilidad de torneo

	Número individuo	**Codificación**	**Número**	**X^2**	**Adaptabilidad** $f_i=q_i/\sum_j q_j$
1	1	(0, 1, 1, 0, 0)	12	144	0,07399
2	1	(0, 1, 1, 0, 0)	12	144	0,07399
3	2	(1, 0, 0, 1, 0)	18	324	0,16649
4	2	(1, 0, 0, 1, 0)	18	324	0,16649
5	5	(1, 1, 0, 1, 0)	26	676	0,34737
6	5	(1, 1, 0, 1, 0)	26	676	0,34737

3.3.7. Selección

Este proceso consiste en elegir a individuos como progenitores o padres y se ampara en la adaptabilidad de los individuos, es decir, aquellos individuos con mayor aptitud tendrán mayor probabilidad de ser escogidos para su reproducción. Este proceso crea una población intermedia.

Existen métodos estocásticos para encontrar la solución óptima, a continuación, se presentan tales métodos:

- Ruleta
- Torneo
- Rango (basado)
- Estocástico

3.3.8. Cruce

Este proceso se lo realiza aleatoriamente, un individuo comparte parte de sus genes con otros y de aquí debe nacer un individuo nuevo. El cruce sucede aleatoriamente, seleccionado el número de genes menos uno (n-1). Un individuo hereda una cantidad de información del padre y otra de la madre.

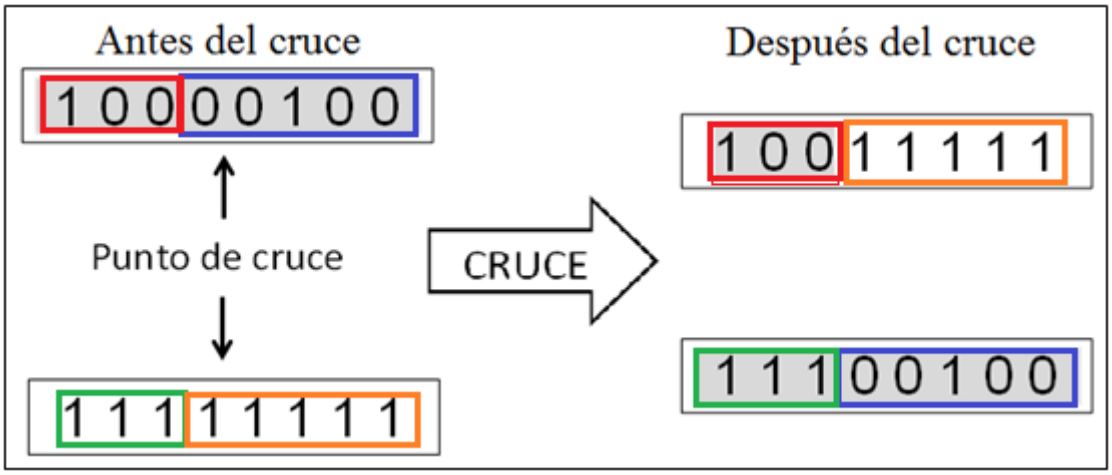
Se debe considerar que este proceso de mutación se lo realiza a partir de la combinación de los mejores individuos, para lo cual se selecciona un punto de cruce aleatorio dependiendo de la cantidad de información genética del cromosoma.

Existen diversos métodos de cruce, pero los más utilizados son los siguientes:

3.3.8.1. Cruce basado en un punto. Los dos individuos seleccionados para jugar el papel de padres, son recombinados por medio de la selección de un punto de corte, para posteriormente intercambiar las secciones que se encuentran a la derecha de dicho punto. Es decir, los genes del padre1 a la izquierda del punto de corte forman parte del hijo1 y los situados a la derecha formaran parte del hijo2, mientras que con el padre2 sucederá lo contrario.

Figura 18

Cruce basado en un punto

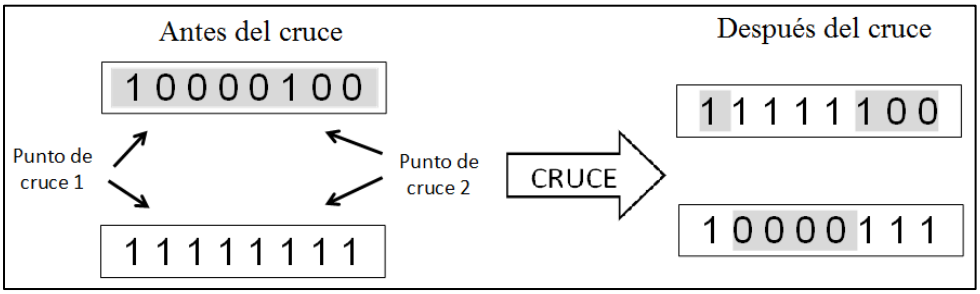


3.3.8.2. Cruce punto a punto. Este tipo de cruce es similar al anterior, pero realizándose para cada gen de los padres. Por tanto, en este cruce los genes pares del padre1 formarán parte del hijo1 y los genes impares formarán parte del hijo2, mientras que para el padre2 sucederá lo contrario.

3.3.8.3. Cruce multipunto. En este tipo de cruce se selecciona aleatoriamente la cantidad de puntos que se van a utilizar para el cruce. De esta forma, y de manera análoga al anterior cruce, se irán intercambiando los genes para formar los dos nuevos hijos.

Figura 19

Cruce multipunto



3.3.9. Mutación

La mutación se considera un operador básico, que proporciona un pequeño elemento de aleatoriedad en los individuos de la población. Si bien se admite que el operador de cruce es el responsable de efectuar la búsqueda a lo largo del espacio de posibles soluciones, el operador de mutación es el responsable del aumento o reducción del espacio de búsqueda dentro del algoritmo genético y del fomento de la variabilidad genética de los individuos de la población. Existen varios métodos para aplicar la mutación a los individuos de una población, pero el más comúnmente utilizado es el de mutar un porcentaje de los genes totales de la población.

Figura 20

Mutación

Este porcentaje de genes a mutar se puede seleccionar de dos maneras, de forma fija, especificando el mismo porcentaje de mutación a todas las generaciones del algoritmo genético

y de forma variable, es decir, modificando el porcentaje de mutación de una generación a otra, por ejemplo, reduciéndolo. De esta manera, se consigue hacer una búsqueda más amplia y global al principio e ir reduciéndola en las siguientes generaciones.

CAPITULO 4

IMPLEMENTACIÓN DE LA SOLUCIÓN

4.1. Modelado FMS

En el presente capitulo se toma como base el caso de estudio planteado por Escobar L., 2015 para la simulación en Jaamsim y posterior evaluación de su desempeño por medio de un algoritmo genético desarrollado.

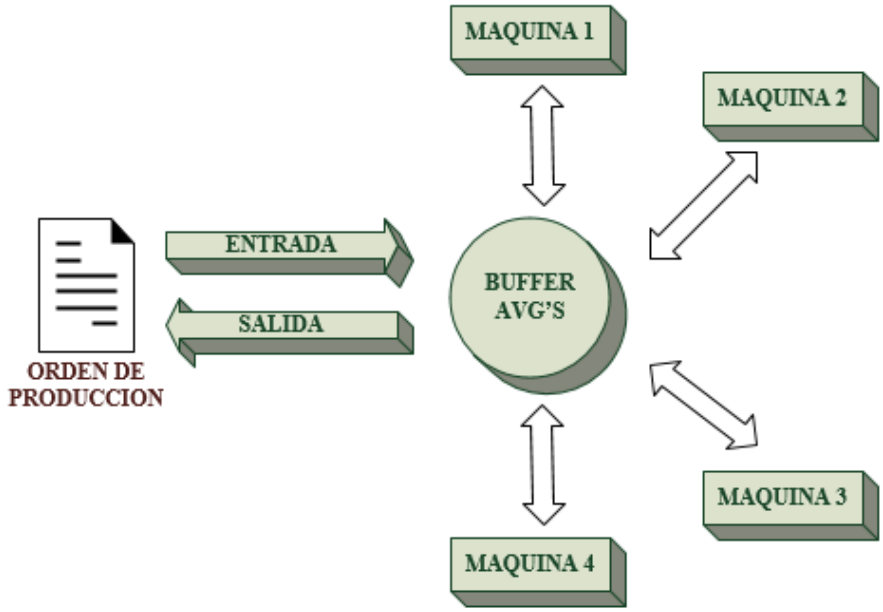
Para este experimento, se considera una FMS que cuenta con:

- Una estación de entrada/salida de materia prima y producto terminado,
- Cuatro estaciones con máquinas para procesamiento de los productos,
- Un buffer para los AVG's (Automatic Guided Vehicles/ Vehículos de guiado automático) que transportará los productos entre estaciones.

El diagrama de bloques correspondiente se expone en la siguiente gráfica:

Figura 21

Diagrama de bloque FMS



Basado en los puntos anteriores se considera la necesidad de métodos para afrontar los problemas para lo cual se toma en cuenta las siguientes restricciones:

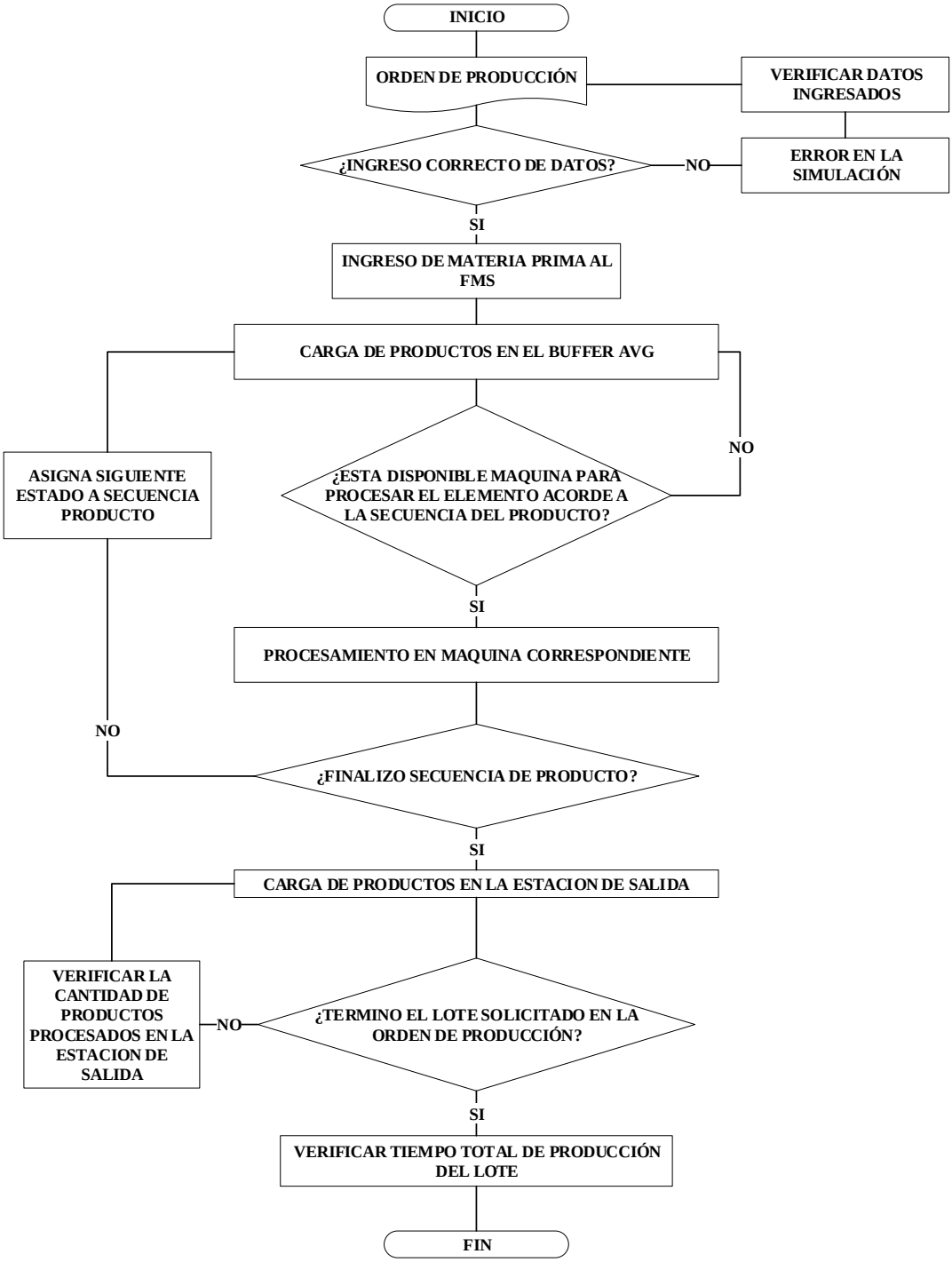
- Todos los productos serán procesados en el mismo FMS.
- Toda la materia prima está a punto para procesar y tienen una tasa de llegada constante a la estación de entrada/salida.
- Cada Máquina puede manejar una operación para un trabajo a la vez.
- Todos los trabajos pasarán por el Buffer para su distribución entre las máquinas hasta que sean procesados por completo
- Las distancias entre estaciones son equidistantes.
- Están predefinidas las secuencias de cada trabajo y los tiempos de servicio para cada producto en las estaciones.
- Se asignará el primer trabajo en cola a la estación disponible correspondiente
- Una vez iniciada la operación en una máquina no pueda ser detenida.

La simulación dentro del entorno Jaamsim, es factible para el modelo planteado gracias a las herramientas que proporciona el software, sin embargo, se debe tener en cuenta que el desarrollador debe tener conocimientos medios a altos en el lenguaje de programación Java para realizar la concatenación entre los elementos de la simulación. La habilidad del programador permitirá realizar una interfaz amigable con el usuario final.

El diagrama de flujo mostrado en la figura siguiente explica la metodología de trabajo de la celda FMS planteada. El proceso de fabricación del lote de productos, inicia al recibir una orden de producción. La orden consta de una variedad de elementos con sus respectivas secuencias y tiempos de operación. Si dicha orden, no contiene los parámetros correctos, la simulación rechazará la orden y se deberá verificar los valores ingresados nuevamente, para continuar con el proceso. Una vez que se ha validado la orden de producción adecuadamente, se ingresa la materia prima al buffer central, en donde los AVG's se encargarán de repartir los pedidos en función de la disponibilidad de las máquinas procesadoras y la secuencia de producción asignada a cada elemento. Los elementos que han terminado su secuencia de producción, serán trasladados a la estación de salida, en la que se confirmará si el lote completo fue fabricado. En este último paso se evaluará el tiempo total de producción como producto de la simulación.

Figura 22

Diagrama de flujo para la simulación en Jaamsim



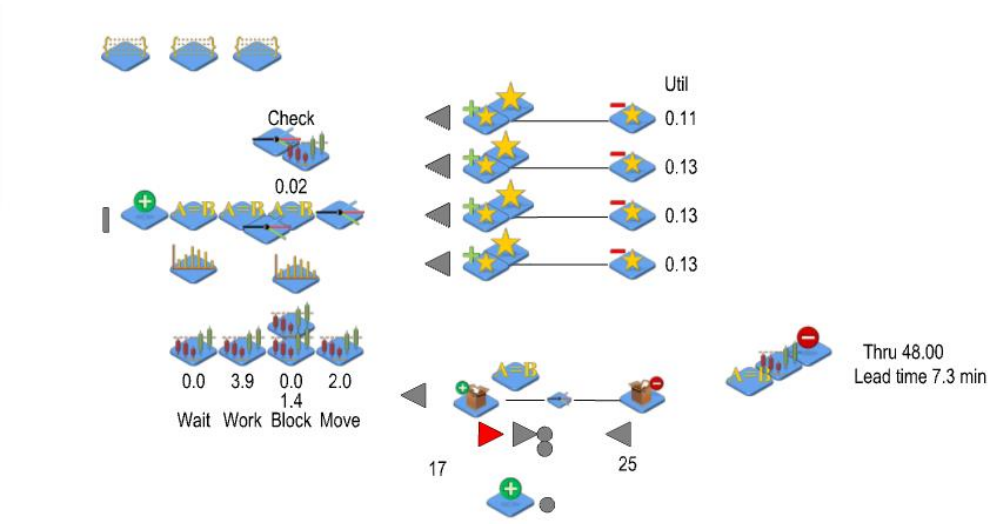
4.1.1. Modelado FMS en el simulador

La investigación ha tomado como referencia el trabajo de Teruel & Aragüés, 2017, en donde se simula un modelo similar al que se requiere. El modelo está a disposición del público en general

y se adaptado al caso mostrado en Escobar L., 2015. En la figura 23 se muestra el modelo desarrollado.

Figura 23

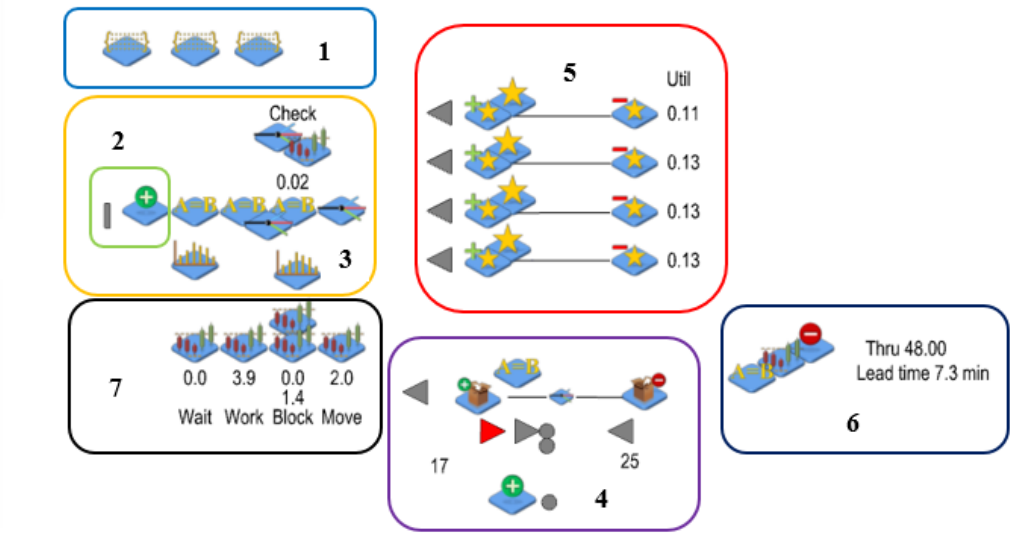
Modelo desarrollado en Jaamsim



Por motivos didácticos, se divide en 7 secciones al modelo simulado obtenido, el cual describe la relación presentada en al diagrama de flujo presentado en la sección 4.1. La Figura 24 describe el desglose de secciones.

Figura 24

Bloques de funcionamiento Jaamsim



La descripción de funciones se presenta de la siguiente manera:

1. Los objetos, FileToMatrix (Archivo a matriz), ingresarán tres bloques de parámetros para la simulación, a través de archivos de texto con extensión txt, que contienen las secuencias asignadas a cada producto. Estos representan las secuencias de producción, los tiempos asignados a cada estación y la distancia entre estaciones.
2. SimEntity (Entidad de simulación) y EntityGenerator (Generador de Entidades) son los objetos que generan la materia prima
3. Mediante la combinación de bloques: Assign (Asignación), Discrete Distibution (Distribución Discreta), Branch (Ramificador) y Statistics (Estadístico), se setea la asignación de los elementos generados en el anterior punto para que sean evaluados en el buffer de los AVG's y sean distribuidos para la producción.
4. El Buffer de AVG's el cual se encarga de validar las secuencias procesadas de cada producto y trasladar a su correspondiente estación. Verifica la disponibilidad en la asignación de operaciones y despacha hacia la estación de salida los productos terminados. Su funcionamiento es similar a la de un servidor que mediante la interacción

de bloques Assign (Asignación), AddTo (Añadir), Branch (Ramificador) y RemoveFrom (Retirar de) asigna los AVG's para las tareas de distribución.

5. Representan las estaciones en donde se encuentran las 4 máquinas por las que debe pasar los productos a obtener, cabe indicar que, en el modelo simulado, el proceso de entrada/salida (2/6) es también considerado una estación. Adicional se muestra una media de utilización por estación. Los bloques que se interactúan en esta sección son Seize (Receptor), Resource (Recursos) y Release (Liberador), en donde se simula la recepción, proceso y despacho de un producto.
6. La estación de salida, muestra el final de elementos procesados en un determinado rango de tiempo.
7. Esta sección refleja datos estadísticos de la simulación mediante validación de propiedades en el flujo del proceso. Se logra en función de bloques Statistics (Estadístico).
8. La información de la configuración asignada a cada bloque que se tomó desde el Constructor de Modelos, se encuentra disponible en el manual de usuario de Jaamsim, en donde se profundiza a detalle los parámetros y estructuras de las variables que pueden ser seteadas.

4.2. Desarrollo del algoritmo metaheurístico

La base para determinar la optimización de secuencias de producción se la realizó a partir de un algoritmo genético desarrollado en lenguaje de programación java, con la versión JDK 9.0.4 (Java Development Kit).

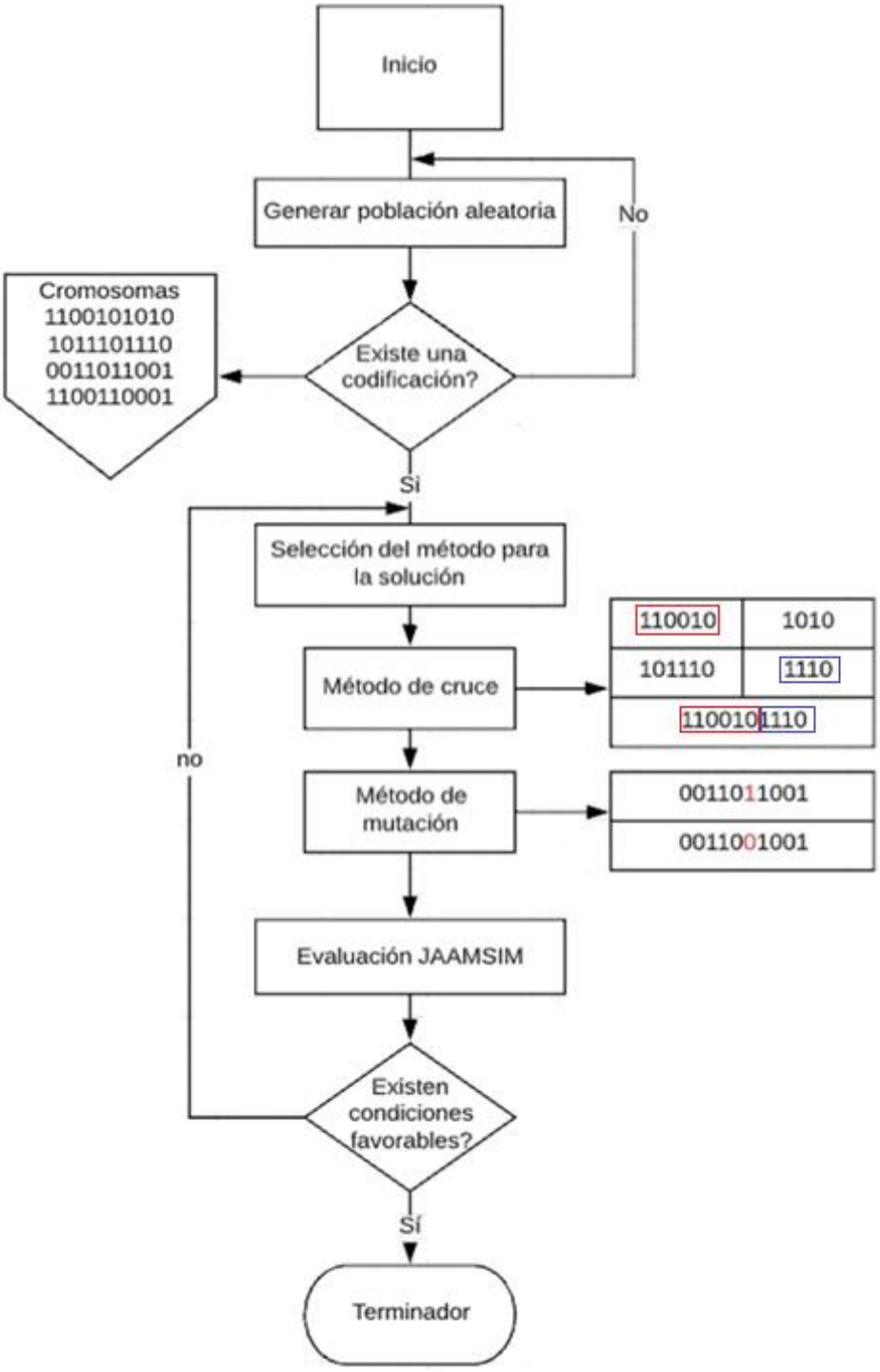
Se debe considerar que el código fue generado a partir de las condiciones y características que un algoritmo genético como se detalla en el diagrama de flujo de la Figura 27, y como se

describió en el Capítulo 3 de esta investigación, ese proceso se lo puede observar también en el código programado en la Figura 25, el cual es la función principal del programa destino a llamar a cada una de las demás funciones en un orden predeterminado como se muestra a continuación:

- Iniciar población
- Convertir población
- Calidad Individuo
- Selección Parejas
- Torneo
- Copiarse
- Combinación – Mutación
- Ver Población

Figura 1

Diagrama de flujo algoritmo genético



El código principal del programa llama a cada una funciones del proceso de código genético como: crear población, adaptabilidad, torneo, duplicar imprirmir, mutación y selección

de secuencia optima como se observa en las figuras 26 y 27, se debe considerar que este proceso es secuencial.

Figura 2

Función principal código genético sección A

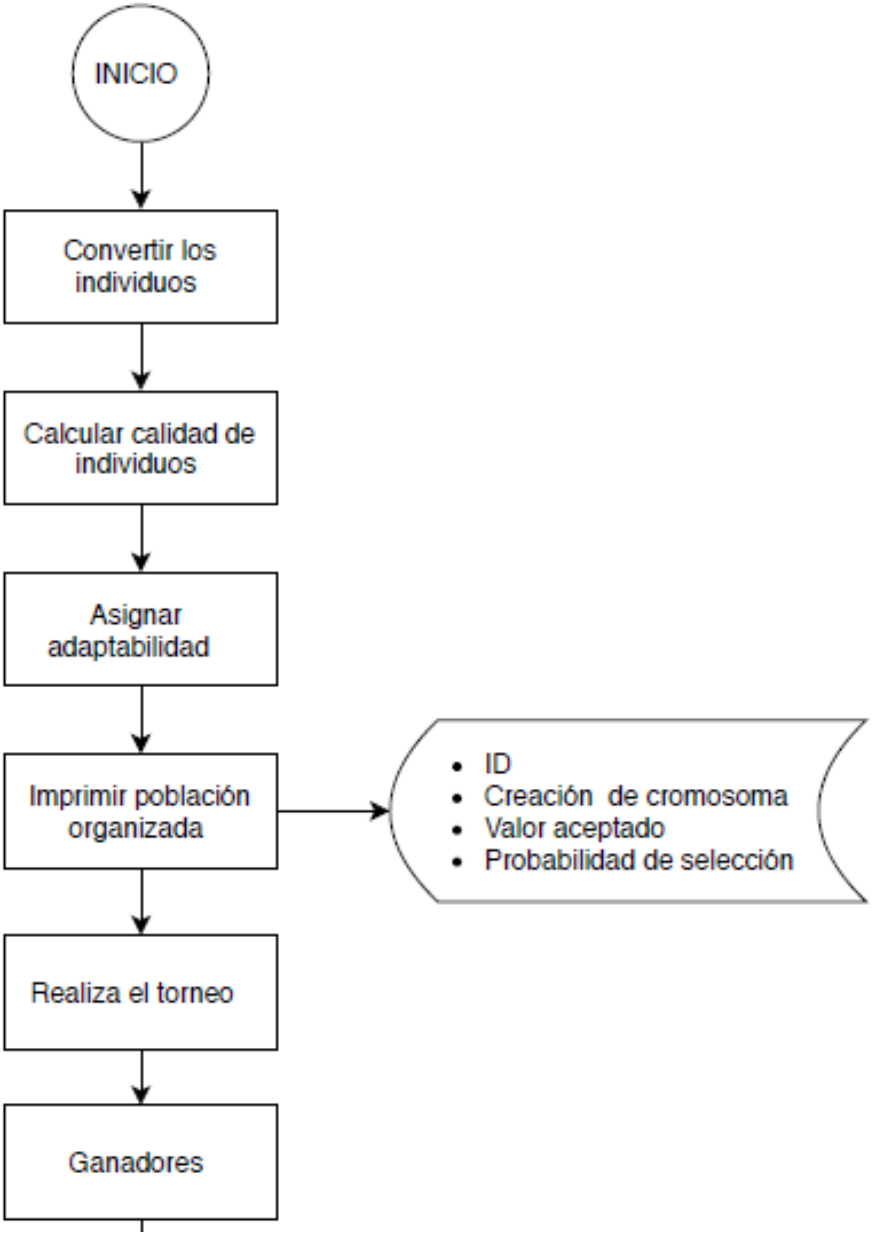
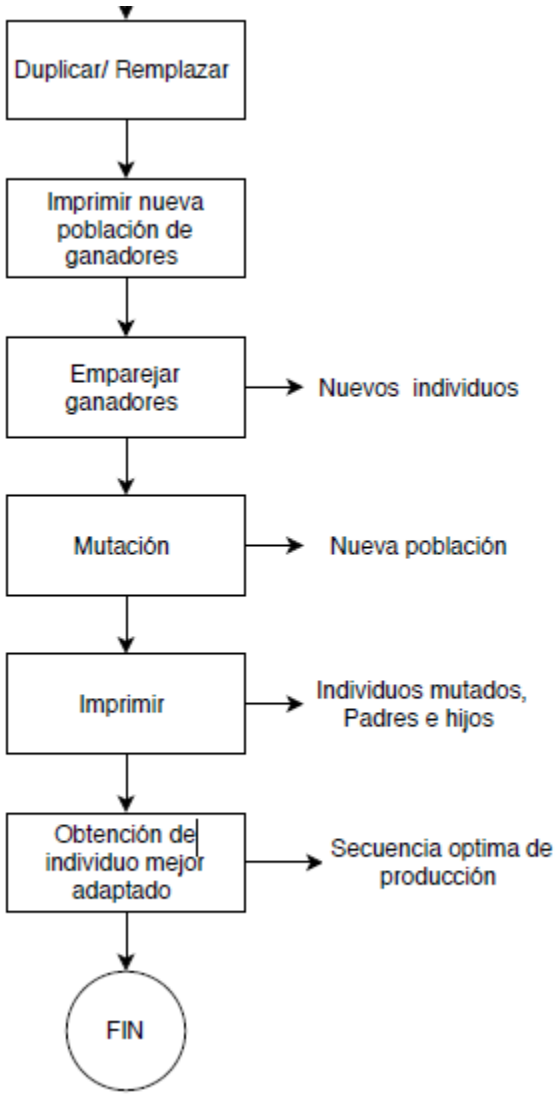


Figura 3

Función principal código genético sección B



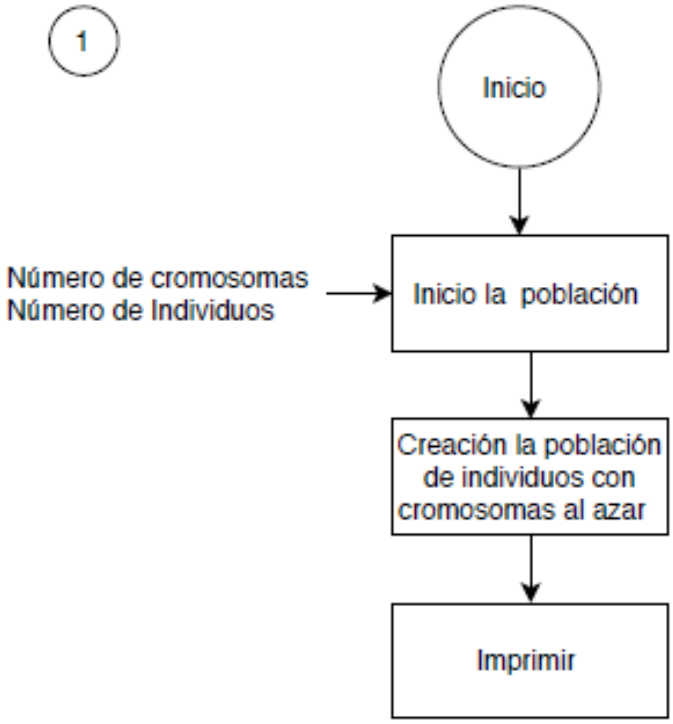
4.2.1. Código para Iniciar la población

Esta sección de código puede generar una población aleatoria de 1 y 0 o configurar una población específica en función de los lotes de producción, en donde las filas representan la creación del total de individuos y las columnas representan la construcción de los cromosomas generando una población inicial, como se puede observar en la Figura 28, que se define la función

Iniciar Población para generar una población aleatoria o se tiene la opción de ingresar manualmente los individuos para generar cada una de las secuencias combinadas por lotes de producción.

Figura 4

Código población inicial



4.2.2. Código para convertir la población

Esta parte del código se encarga de transformar el cromosoma o la secuencia de lotes de cada individuo a un tiempo total de producción basado en un formato de codificación y tiempos de entrega por producto como se detalla en la Tabla 7, la función encargada de realizar este proceso tiene el nombre de convertir_individuo (ver Figura 29).

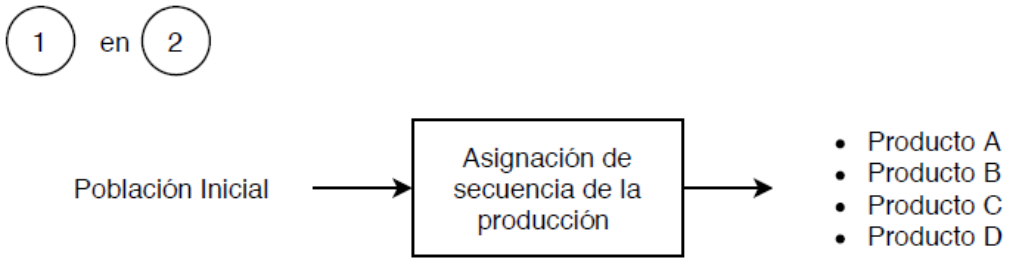
Tabla 7

Codificación para cromosoma de secuencia

Información de cromosoma	Representación	Tiempo de entrega
1	Producto A	8
2	Producto B	10
3	Producto C	12
4	Producto D	14

Figura 29

Código convertir la población

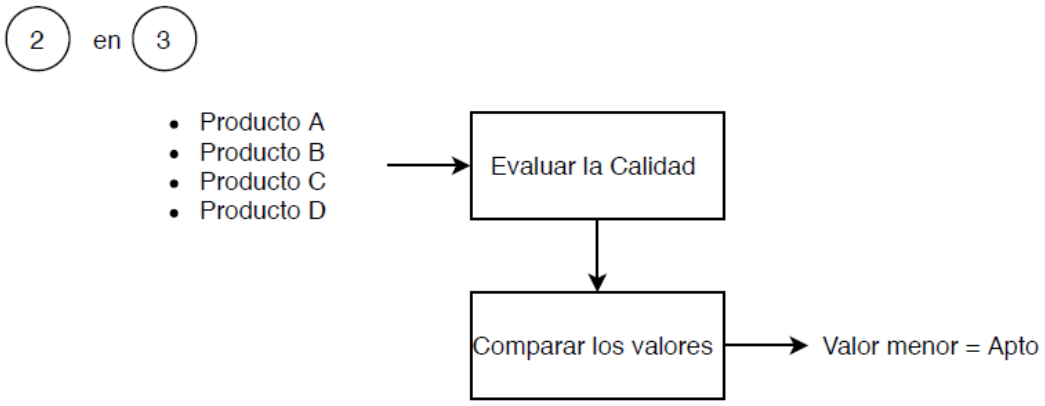


4.2.3. Código para calidad del individuo

La función encargada de evaluar la calidad de un individuo tiene el nombre de Calidad_Individuo (Figura 30), en la cual se compara los valores obtenidos de cada individuo por la función convertir_individuo, buscando el menor valor, ya que al tratarse de tiempos de producción el valor más apto es el menor.

Figura 5

Código calidad individuo

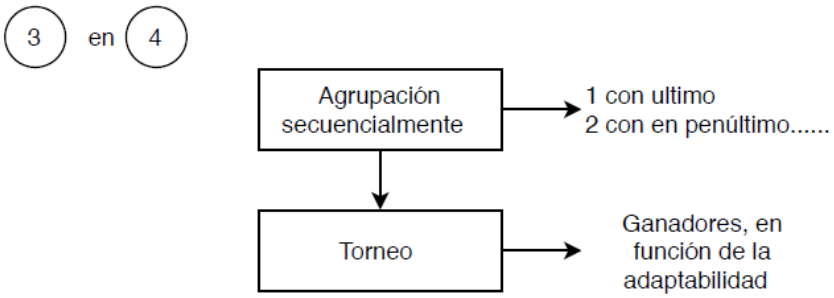


4.2.4. Código para selección de parejas

Para realizar la selección de parejas, se usa el proceso de agrupar el primero con el último, el segundo con el antepenúltimo, sucesivamente hasta que todos los individuos tengan una pareja, la función asociada a dicho proceso tiene el nombre de Selección_Parejas como se observa en la Figura 31. Dentro de este código se ejecuta también el Torneo que se encarga de realizar una competencia basado en la selección de parejas previamente elegidas, obteniendo de ahí los mejores individuos o los individuos más aptos

Figura 6

Código para selección de parejas

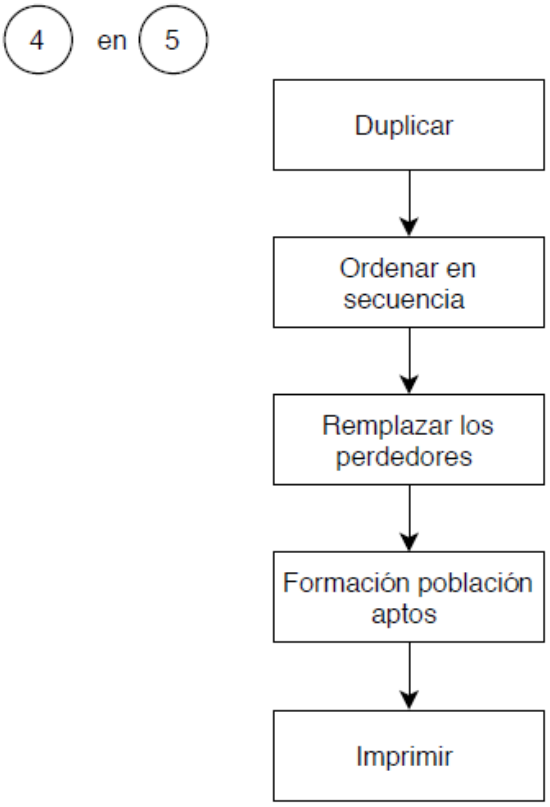


4.2.5. Código para copiarse

Función con el nombre de Copiarse, tiene por objetivo duplicar y ordenar a los ganadores del torneo reemplazando a los perdedores y formando una población con individuos más aptos (Figura 32).

Figura 7

Código para copiarse

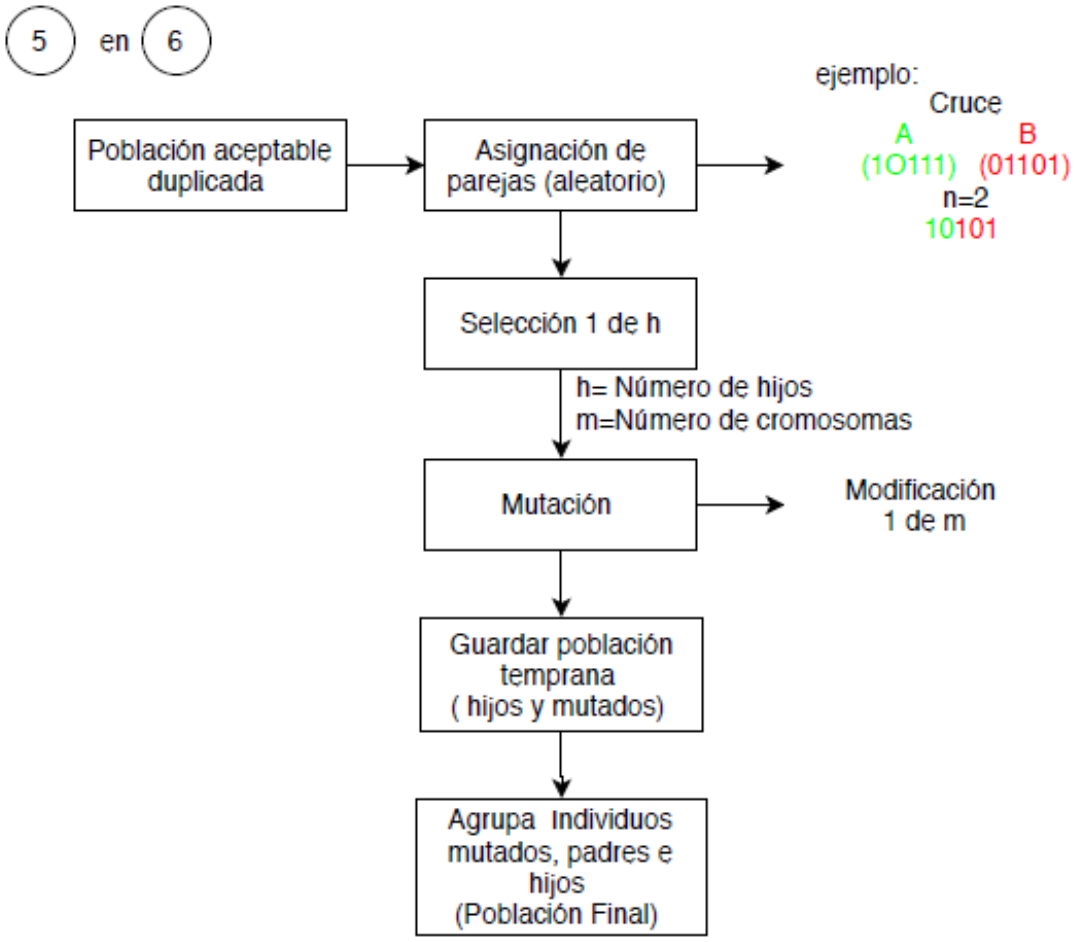


4.2.6. Código para combinación y mutación

En esta parte del código se específica el cruce de individuos, se debe considerar que este proceso es aleatorio tanto para la selección de individuos como para la elección del punto de cruce de la división de información genética de los padres, otra parte aleatoria de este código es la selección del hijo de entre los resultantes del cruce para la mutación, además el gen elegido para ser mutado también se obtiene aleatoriamente, esto se pude observar en la función con nombre Combinación_Mutación, tanto como los hijos y el individuo mutado son guardados en una población temporal como se puede observar en Figura 33.

Figura 8

Código para Combinación y mutación



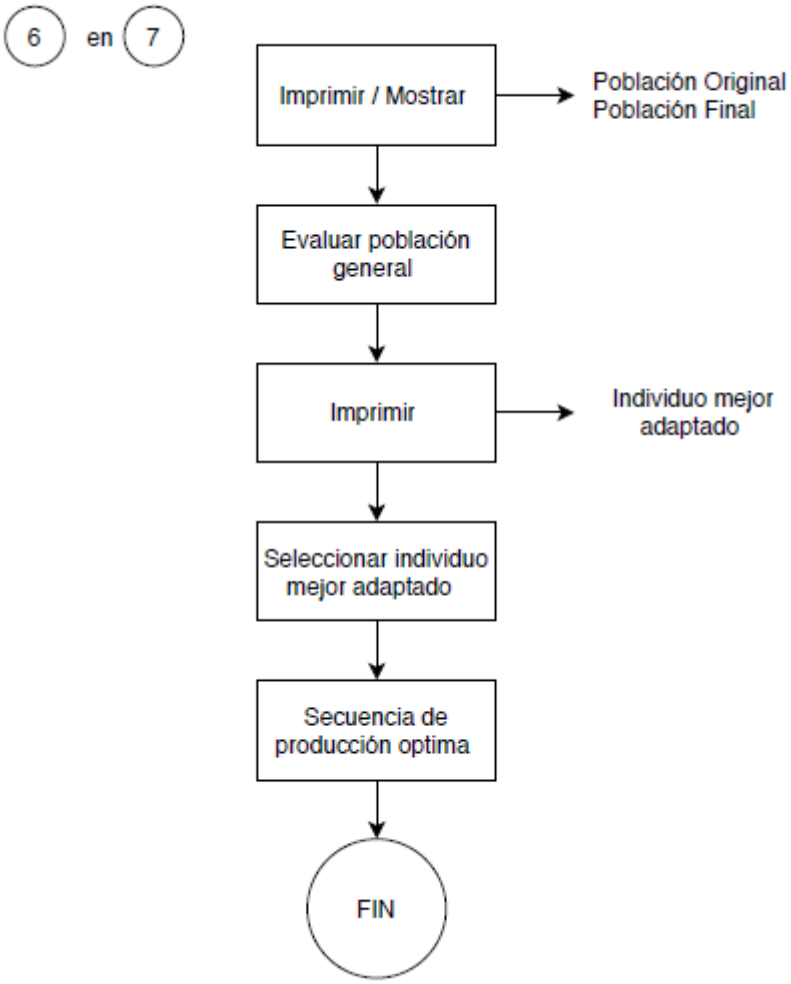
4.2.7. Código para ver población

Esta última función es la encargada de imprimir en pantalla o mostrar los datos tanto de la población original de individuos como la población final: formada por los mejores individuos, hijos y el individuo mutado, considerando que para ese proceso existe una asociación con la población temporal generada en la función Combinación_Mutación (ver Figura 34). Al final como resultado

de la evaluación de los individuos dentro del algoritmo genético el programa entre la mejor secuencia de entre todos los individuos considerando el menor tiempo de secuencia.

Figura 9

Código para ver la población

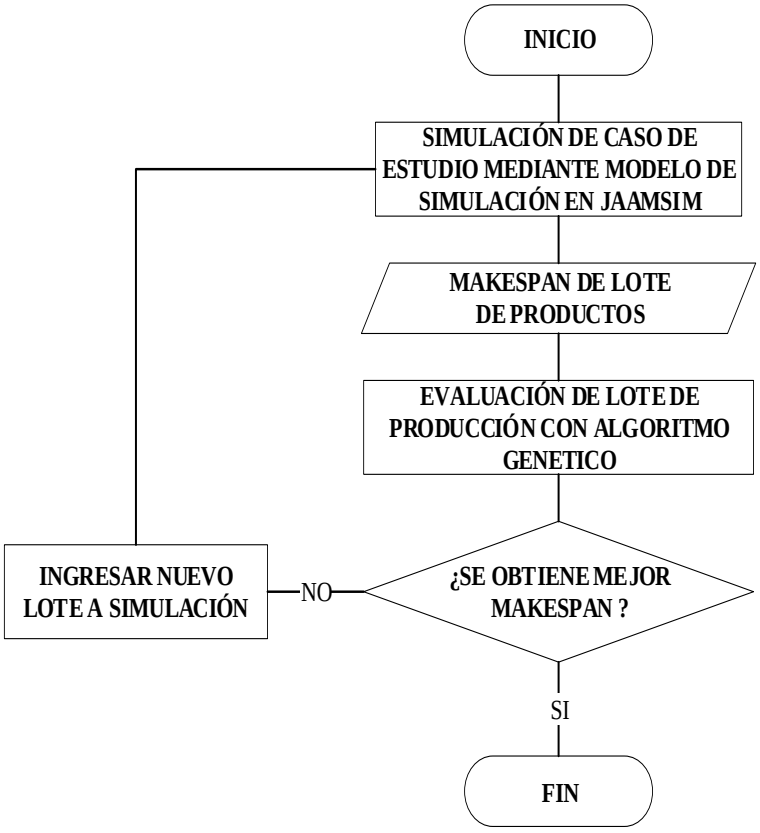


4.3. Diagrama de flujo

La relación entre las secciones 4.1 y 4.2 de la presente investigación, se demuestra mediante el siguiente diagrama de flujo, el proceso y unión de los dos procesos meta heurísticos tanto de Jaamsim como del algoritmo genético.

Figura 10

Diagrama de flujo metaheurístico



Dentro del entorno de Jaamsim, se simulará un caso de estudio, que se detalla en el Capítulo 5. Esta simulación reflejará un tiempo de producción general por lote inicial. (Makespan). El lote se evaluará por el algoritmo genético, que se propone en la sección 4.2.; este algoritmo producirá un nuevo lote, que debe ser evaluado en la simulación. Sí el nuevo lote produce un mejor Makespan, se habrá logrado uno de los objetivos de esta investigación.

CAPITULO 5

RESULTADOS

5.1. Resolución del caso de estudio

Este capítulo, se centra en la prueba de aplicación del método meta heurístico a un Caso de Estudio citado en Escobar, 2015; que desea producir un lote de 10 unidades de 4 tipos diferentes de productos, en una FMS de características análogas al que se describe en el capítulo 4. La siguiente tabla se especifica las rutinas para los diferentes productos.

Tabla 8

Especificaciones de productos

Producto	Operación 1		Operación 2		Operación 3		Operación 4		Cantidad
	Máquina	Tiempo	Máquina	Tiempo	Máquina	Tiempo	Máquina	Tiempo	
A	1	2	2	2	3	2	4	2	3
B	2	2	4	4	3	2	1	2	4
C	1	3	2	2	3	3	4	4	2
D	3	5	1	2	4	4	2	3	1

Se establece un lote inicial de producción, de acuerdo a las cantidades solicitadas en la orden de trabajo; que se considera la población inicial para inicio de las pruebas. Los productos están codificados de la siguiente manera:

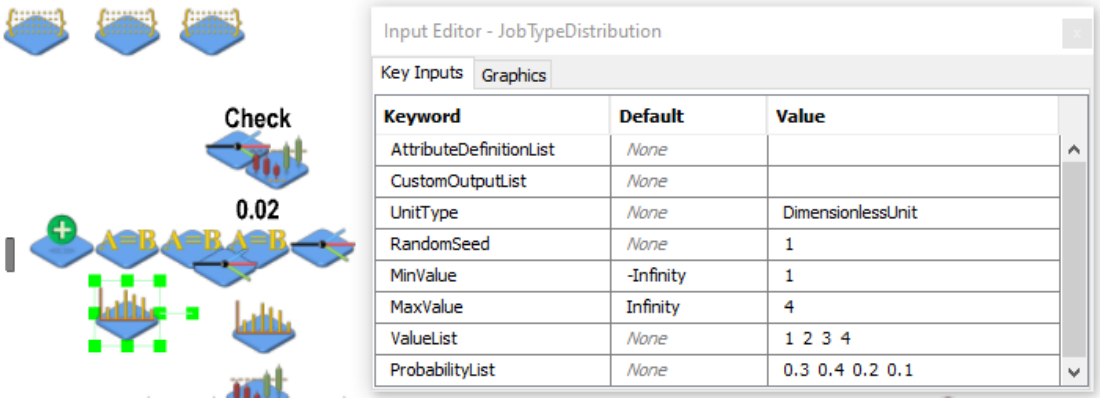
- A = 1
- B = 2

- C = 3
- D = 4

La población debe ser ingresada en el Editor de Entradas que corresponde a la distribución discreta de nombre JobTypeDistribution, en donde se setea la orden de producción requerida. Este valor se puede tomar a manera de un vector. En el campo ValueList se detalla los tipos de producto y en ProbabilityList, la cantidad.

Figura 11

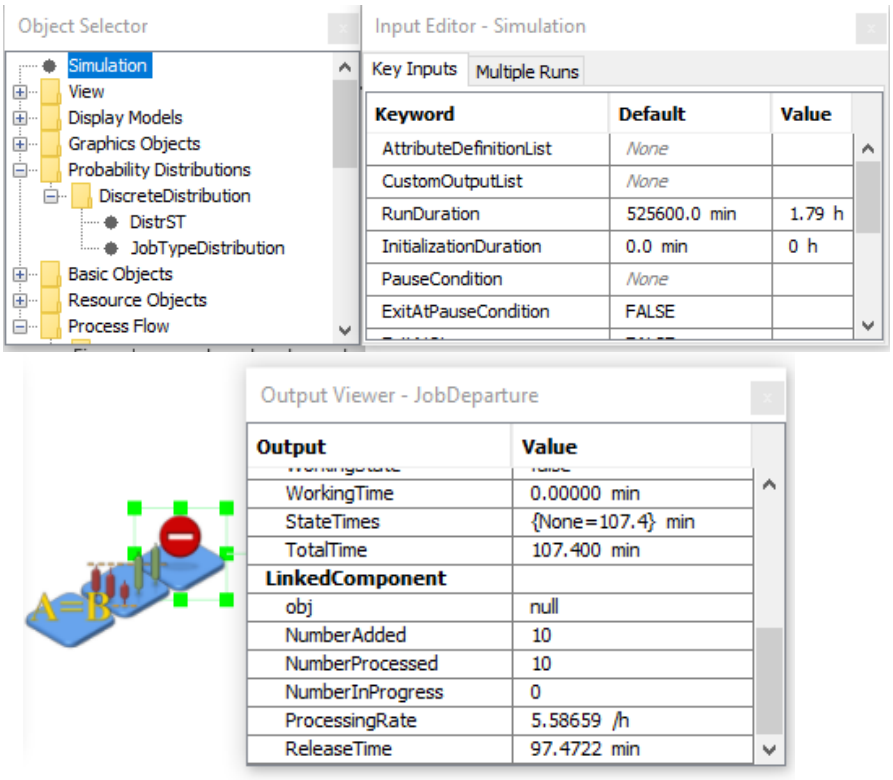
Ingreso de orden de producción a modelo en Jaamsim



Se debe correr la simulación con los datos ingresados; y verificar la cantidad producida en un determinado tiempo hasta ajustar las 10 unidades analizadas para el caso de estudio en el objeto JobDeparture.

Figura 12

Verificación de valores en simulación



5.2. Comparación de resultados

Los resultados que se presentaron durante las pruebas son los siguientes:

Tabla 9

Análisis de resultados

	TIPO DE PRODUCTO	PRODUCTO CODIFICADO	PROBABILIDAD DE PRODUCCIÓN	EVALUACIÓN DE TIEMPO EN SIMULADOR (HORAS)	PORCENTAJE DE REDUCCIÓN DE TIEMPO (%)	TIEMPO REDUCIDO (H)
POBLACIÓN INICIAL (ORDEN DE PRODUCCION)	[A A A B B B B C C D]	[1 1 1 2 2 2 2 3 3 4]	[0.3 0.4 0.2 0.1]	**1.79**	**0.00**	**0.000**
INTERACCION ALGORITMO GENETICO 1	[A A A A B C B B A D]	[1 1 1 1 2 3 2 2 1 4]	[0.5 0.3 0.1 0.1]	**1.625**	**-9.22**	**0.165**
INTERACCION ALGORITMO GENETICO 2	[A A A A B C B B A D]	[1 1 1 1 2 3 2 2 1 4]	[0.5 0.3 0.1 0.1]	**1.625**	**-9.22**	**0.165**
INTERACCION ALGORITMO GENETICO 3	[A A A A B C B B A D]	[1 1 1 1 2 3 2 2 1 4]	[0.5 0.3 0.1 0.1]	**1.625**	**-9.22**	**0.165**
INTERACCION ALGORITMO GENETICO 4	[A A A B B C B B A D]	[1 1 1 2 2 3 2 2 1 4]	[0.4 0.4 0.1 0.1]	**1.71**	**-4.47**	**0.080**
INTERACCION ALGORITMO GENETICO 5	[A A A B B C B B A D]	[1 1 1 2 2 3 2 2 1 4]	[0.4 0.4 0.1 0.1]	**1.71**	**-4.47**	**0.080**

Se verifica que las secuencias generadas a través del algoritmo se repiten. Se debe a que nuestra población inicial, no ha presentado una mayor mutación en el paso del tiempo y la secuencia se ajusta al caso de estudio referencial. A mayor número de interacciones se aprecia el mismo comportamiento al tiempo final operativo.

Los lotes de producción dispuestos por medio del algoritmo, generaron mejores makespan al lote base. La reducción presente no es significativa, sin embargo, considerando esta variación a gran escala, se podría apreciar una diferencia distintiva bajo este criterio.

El modelo y el algoritmo han sido diseñados de forma escalable y flexible; pudiendo analizar modelos análogos con pocos cambios en la configuración.

5.3. Análisis de resultados

Dentro de los objetivos establecidos en el proyecto de tesis, se determina utilizar Software Libre que permita al usuario modelar un sistema de manufactura flexible y optimizar uno o varios parámetros de su diseño; este objetivo se ha conseguido dado que toda la ejecución del proyecto de tesis ha sido totalmente desarrollada con herramientas Open Source.

En los capítulos anteriores se ha explicado la forma en que se ha hecho la integración hacia un proceso meta heurístico. Se destaca que el objeto de estudio, ha cumplido sus propósitos, denotando una reducción posterior a la aplicación del algoritmo como herramienta de análisis retroalimentando el modelo simulado para su evaluación.

La Tabla 9 muestra las interacciones que se han presentado de manera mas concurrente como resultado de las distintas ejecuciones del algoritmo, las cuales se ha evaluado en el modelo de simulación creado, denotan una reducción de los tiempos de producción al ejemplo analizado. Los resultados cuantitativos son mejor apreciables en las siguientes figuras.

Figura 13

Evaluación de tiempo por lote

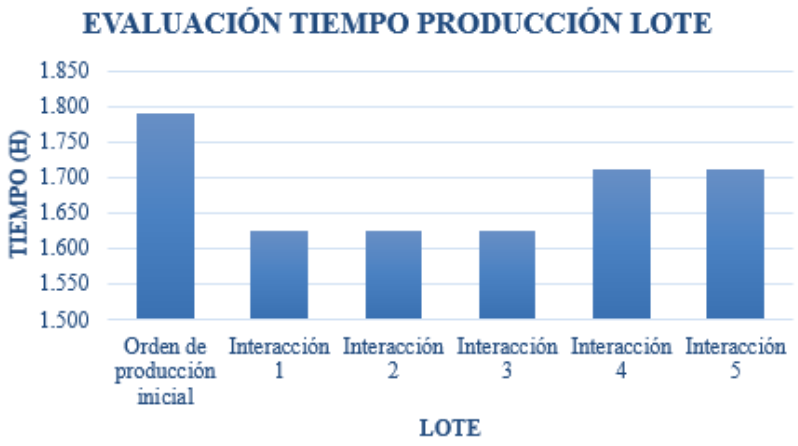
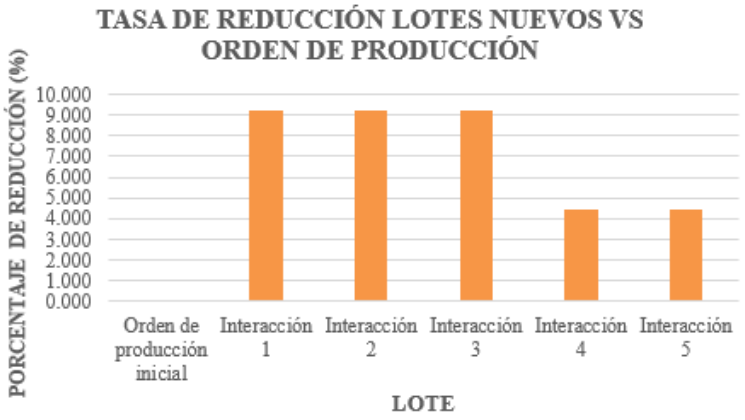


Figura 14

Reducción porcentual absoluta en tiempos de producción



Se aprecia que la media porcentual entre las pruebas simuladas establece una reducción de un 5.23% al tiempo de ejecución total inicial. La Figura 39 evidencia que las soluciones que ha proporcionado la ejecución del algoritmo disminuyen los tiempos de producción de los lotes fabricados.

Dado que una de las mayores limitantes es la singularidad de cada proceso, su simulación es única. El caso de estudio que se presentó, mostraría una tendencia diferente en sus valores al existir un cambio en: variación de las restricciones del modelo, modificación en las secuencias del

producto, alteración en la tasa de llegada de la materia prima, el diferente tratamiento en la prioridad de las colas para cada máquina o el uso de distribuciones estadísticas diferentes.

Otro factor a considerar es el número de interacciones del producto, o el grado de mutación para las generaciones de la población, que determinará individuos con mejores características. El enfoque de las pruebas se basó en obtener una secuencia con mejor desempeño en su reducción de makespan; no siempre obteniendo los productos iniciales esto se interpreta que existirá producto en inventario que servirá para atender nuevos pedidos y dado que el diseño es flexible bastará con cambiar la orden de producción para los valores deseados.

CAPITULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- A partir de la información investigada sobre los métodos heurísticos de optimización, se determinó que se puede crear herramientas para facilitar la modelación y análisis de sistemas de manufactura automatizados en las que se debe considerar la particularidad para cada caso de estudio y realizar una correcta elección de los parámetros para obtener un resultado favorable.
- Los métodos heurísticos de optimización son herramientas robustas que permiten modelar y analizar con flexibilidad los sistemas, existiendo una amplia variedad de métodos. Su elección requiere de experiencia en campo de manejo de sistemas de manufactura para que bajo un excelente criterio analítico se realice la mejor elección del método para que se adapte al modelo propuesto.
- En la presente investigación se desarrolló un algoritmo heurístico en código abierto, capaz de optimizar secuencias de producción basado en algoritmos genéticos de n individuos con n cromosomas y determinar la secuencia más óptima para sistema de manufactura automatizados, así como hallar la mejor solución de problemas combinatorios muy grandes; resultados que pueden ser acoplados y comprobados en la herramienta Jaamsim.
- La comparativa con el caso de estudio, citada en Escobar L. ,2015; en el modelo aplicado con AVG's; presenta un comportamiento análogo al descrito en la investigación de

referencia; cabe indicar, que al igual que se analiza en Escobar L., 2015 se debe considerar las restricciones para hacer al modelo cercano a la realidad.

- Con el número de interacciones evaluadas para la misma población inicial, se logró determinar que las mejores secuencias y disminución de tiempos de producción, en donde los resultados dominantes se obtuvieron en los cruces de individuos (hijos), también se presentaron soluciones en los individuos mutados, pero a diferencia de los anteriores presentaron con un porcentaje demasiado bajo.
- Considerando que se dieron variaciones aleatorias, las nuevas poblaciones obtenidas no presentaron variaciones drásticas al desempeño de la población inicial, no obstante, mejoraron un porcentaje a considerar, entre todas las iteraciones, determinando que el programa cumple con su función de optimizar.
- Jaamsim, como simulador de eventos discretos, resulto ser una herramienta potente y de interfaz amigable al usuario, su diseño de bloques permite un entendimiento del modelado de procesos fácil de asimilar. Sin embargo, una de las limitantes es la configuración e integración de los elementos, la cual demanda un conocimiento medio a alto del lenguaje de programación Java para el desarrollador de la simulación.

6.2. Recomendaciones

- El apoyo de la comunidad en línea de Jaamsim es primordial en el entendimiento del software; se recomienda la interacción con los miembros del foro disponible en la internet para solventar dudas e inquietudes al respecto de futuros desarrollos.
- Es importante realizar un estudio concreto sobre el sistema de eventos discretos a optimizar, la particularidad que presenta cada sistema, limita el alcance al investigador en la implementación de soluciones globales para varios sistemas.

- Con el propósito de alimentar la documentación disponible sobre el entorno de simulación Jaamsim para la comunidad que emplea herramientas open source, se recomienda proponer más proyectos con la implementación de este software libre. Durante el desarrollo de la investigación se pudo apreciar que a pesar de que Jaamsim se encuentra lleva varios años en línea para su uso no se dispone de mucha información clara accesible para nuevos usuarios.
- En base a la flexibilidad presentada tanto en el simulador como en el método heurístico empleado, pueden ser modificados para cambiar la cantidad de productos a fabricar en el FMS.

6.3. Proyectos futuros

- Describir una investigación para la optimización de sistemas de manufactura empleando otro de los métodos heurísticos disponibles y verificar su funcionamiento en el simulador de código abierto.
- Evaluar y definir las mejores reglas para ser utilizadas en la implementación del análisis meta heurístico de sistemas de manufactura flexible.
- Proponer un modelo de sistemas de manufactura flexible con gran escala de producción y explicar estrategias empleadas para el tratamiento de las colas en los servidores producidas por la alta demanda en las operaciones.
- Realizar el análisis variando el tipo de distribuciones estadísticas para la asignación de tareas sobre el modelo presentado y verificar el comportamiento del sistema.

BIBLIOGRAFIA

Absi, N., Archetti, C., Dauzère-Pérès, S., & Feillet, D. (2014). *A two-phase iterative heuristic approach for the Production Routing Problem*. Obtenido de Informs pubs online: https://doi.org/10.1287/trsc.2014.0523

Alquilano , C., & Graw, M. (2014). *En qué consiste el sistema de manufactura flexible?* Obtenido de http://cdcsoftwarebackoffice.blogspot.com/2014/09/en-que-consiste-el-sistema-de.html

Bernal, M., Cock, G., & Restrepo, J. (2015). Productividad en una celda de manufactura flexible simulada en promodel utlizando path networks type crane. *Dialnet*, 133-144.

Caceres , R., & Rivas, W. (2017). *Desarrollar un software planificador de rutas, para encontrar una ruta óptima, mediante algoritmos genéticos (JPAP) con interfaz desarrollada en java*. Obtenido de Unidad Académica de Ingeniería Civil - UTMACH: http://repositorio.utmachala.edu.ec/bitstream/48000/10736/1/TUAIC_2017_IS_CD0003.pdf

Caceres., R., & Rivas, W. (2017). *Desarrollar un software planificador de rutas, para encontrar una ruta óptima, mediante algoritmos genéticos (JPAP) con interfaz desarrollada en java*. Obtenido de Unidad Académica de Ingeniería Civil - UTMACH: http://repositorio.utmachala.edu.ec/bitstream/48000/10736/1/TUAIC_2017_IS_CD0003.pdf

Caldas, A., Lorenzo, S., & Carpente, M. (4 de septiembre de 2014). *Aplicación de algoritmos heurísticos para optimizar el coste de.* Obtenido de http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_759.pdf

Calderón, J. (2009). *Algoritmos Genéticos*. Obtenido de Redes Neuronales Artificiales I.

Celeberrima. (2019). *Qué es un sistema de producción – definición*. Obtenido de https://www.celeberrima.com/que-es-un-sistema-de-produccion-definicion/

Chen, J., Chen, K., Lin, C., Chen, C., & Yang, C. (2013). A study of a heuristic capacity planning algorithm for weapon production system. *International Journal of Electronic Business Management*, 46-57.

Chryssolouris, G. (2006). *Manufacturing Systems: Theory and Practice.*

Cruz, J., & Badii, H. (2004). SMED: El camino a la flexibilidad total . *Innovaciones de negocioes* , 277-283.

Díaz, F. (2011). *Tipos de Sistemas de Manufactura .* Cuautitlán Izcalli: México.

EAE Business School. (2018). *Tipos de sistemas de producción industrial y sus características*. Obtenido de Retos en Supply Chain: https://retos-operaciones-logistica.eae.es/tipos-de-sistemas-de-produccion-industrial-y-sus-caracteristicas/

Escobar, L. (2015). Adaptive scheduling of flexible manufacturing systems with material handling constraints using GA and Colored Timed Petri Net. Japón.

Flores, k., & Medina, P. (2014). *Desarrollo de un software académico para programar la producción en los sistemas de manufactura flexible*. Obtenido de Universidad Tecnológica

de Pereira Facultad de Ingeniería Industrial: http://recursosbiblioteca.utp.edu.co/tesisd/textoyanexos/0053F634.pdf

Frank , M., Laroque, C., & Uhlig, T. (2014). *Reducing computation time in simulation-based optimization of manufacturing systems*. Obtenido de IEEE Xplore: https://ieeexplore.ieee.org/abstract/document/6721642/authors

García , E., & Carvajal, E. (2018). Desarrollo de un simulador de sistemas de manufactura con interfaz gráfica basado en redes de petri. *Revista iberoamericana de ingeniería mecánica - Dialnet*, 69-79.

Gestal, M., Rivero, D., Ramón, J., Dorado, J., & Pazos , A. (2010). *Introducción a los algoritmos genéticos y la programación genética* . Obtenido de Universidad da Coruña: http://www.galeon.com/dantethedestroyer/algoritmos.pdf

Granda, M. (2012). *Redes de Petri: Definición, Formalización y Ejecución.* España: Universidad de Cantabria.

Groove. (1990). *Sistemas Integrados de Manufactura.*

Groover, M. P. (2002). *Automation, Production System, and Computer Integrated Manufacturing.* New Jersey: Prentice Hall Press Upper Saddle River.

Inmaculada, G. P. (2002). *Java Threads (Hilos Java).* Salamanca: Universidad de Salamanca.

King, D., & Harrison, H. (2013). *Open-sorce simulation software "Jaamsim"*. Obtenido de Proceedings of the 2013 Winter Simulation Conference: https://informs-sim.org/wsc13papers/includes/files/190.pdf

King, D., & Harrison, H. (s.f.). *User Manual Jaamsim*. Obtenido de 2017: https://jaamsim.com/docs/JaamSim%20User%20Manual%202017-10.pdf

Lara, J. (2012). *Dinámismo en Redes de Petri Coloreadas.* Castilla: Universidad de Castilla.

M. Azaranga, E. G. (2013). *Simulación y Análisis de Modelos Estocásticos.* México: Unamerida.

Murillo, L. D. (2010). *Simulación de un sistema de Manufactura Flexible con Redes de Petri Coloreadas.* Costa Rica: Escuela de Ingeniería Electromecánica .

Rabanal, P., Rodríguez, I., & Rubio, F. (2010). *Algoritmos heurísticos y aplicaciones a métodos formales* . Obtenido de Universidad Complutensis: https://eprints.ucm.es/12027/1/T32515.pdf

Romero, M. (2016). *Sistema de Producción: Los 3 Tipos y Factores que lo Integran*. Obtenido de Lifeder.com: https://www.lifeder.com/sistema-de-produccion/#

Shannon, R. (1988). *Simulación de Sistema Diseño, Desarrollo e Implantación .* España: Trillas.

Sthipal, S. (1990). Flexibility in Manufacturing. *A survey International Journal of Flexible Manufacturing Systems*, Vol 2.

Teruel, E., & Aragüés, R. (2017). *Aprendiendo simulación de eventos discretos con JaamSim*. Obtenido de XXXVIII Jornadas de automática: https://dialnet.unirioja.es/servlet/articulo?codigo=6591655

Valtierra , J., & Sausedo, J. (2014). *Reconfiguración autónoma de sistemas de manufactura mediante la optimización de funciones de desempeño del proceso*. Obtenido de Latin

American and Caribbean Conference for Engineering and Technology: http://www.laccei.org/LACCEI2014-Guayaquil/RefereedPapers/RP016.pdf

Y.Ramirez, J. M. (2014). Manufactura y Robótica. *Emaze*, 1-23.

Zhou, M. (2005). *Petri Net Systhesis for Discrete Evente Control of Manufacturing Systems.* Boston: Kluver Academic Publishers.

Apéndice A. Código fuente del algoritmo genético