

Comparación del desempeño de dos algoritmos genéticos híbridos para el problema de formación de celdas de manufactura considerando el movimiento de los trabajadores (PFCMT)

Dilan Jhoanny Mogollón Carreño

Sebastián Elías Páez Becerra

Trabajo de Grado para Optar al título de Ingeniero Industrial

Director

M.Sc. Laura Yeraldín Escobar Rodríguez

Codirector

M.Sc. Edwin Alberto Garavito Hernández

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Estudios Industriales y Empresariales

Bucaramanga

2022

**Dedicatoria**

A mi familia, por cada gesto de amor,  
enseñanza y valor tarazado en mi vivir.

A mi pareja, por ser quien escucha y apoya  
cada idea y sueño que dispongo realizar en mi vida.

*Sebastián Elías Páez Becerra*

A mis padres, mis abuelos,  
por enseñarme a ser una mejor persona  
y a ti, por enseñarme a confiar en mí mismo  
y apoyarme en cada idea que tenía.

*Dilan Jhoanny Mogollón Carreño*

### **Agradecimientos**

Iniciar agradeciendo a la vida que me permite estar en este preciso instante a punto de cumplir un primer objetivo y sueño de vida. En segundo lugar, agradecer a mis padres que confiaron en mis capacidades y me enseñaron que todo lo bueno en la vida llega con esfuerzo y dedicación. Agradecer a mi abuela Anita, fuente de amor y cariño que con tan solo mirar su vivir me recuerda la frase “Haz las cosas ordinarias con un amor extraordinario”. Agradezco también a mi pareja Lucía, quien es la voz de aliento en los momentos donde se torna el camino oscuro, me impulsa a no abandonar sueños y crear nuevas metas.

Por otra parte, agradezco a la profesora Laura Escobar, por su guía, por otorgarnos un espacio en su atareada agenda, por compartir sus conocimientos, por tenernos paciencia y por resolver cada duda del proyecto con sabiduría. Agradecer a mi compañero Dilan que fue el puente de comunicación para entrar por mi parte a participar en el proyecto y por sus aportes.

Y finalmente agradecer al grupo de investigación OPALO y a la escuela de estudios industriales y empresariales que avaló el proyecto y permitió finiquitar el título de ingeniero industrial.

*Sebastián Elías Páez Becerra*

Quiero iniciar agradeciéndote, por haberme enseñado tanto y aunque ya no estás, me enseñaste a perseguir mis sueños. Quiero agradecer a mis padres por todo su apoyo porque parte de lo bueno en mi es gracias a ellos, a mis abuelos por brindarme su apoyo y me doy las gracias por no darme por vencido y la frase que mi madre una vez me dijo “Puedes no hacer lo que amas, pero ama lo que haces”.

Gracias inmensas a mi profesora Laura, por iluminar mi camino y darme la claridad para retomar mis metas, por poner su confianza en mí y, sobre todo, más que mi directora, la considero mi amiga, me enseñó que el trabajo duro tiene su recompensa. Agradecer a mi compañero Sebastián por mantener su buen humor en momentos de estrés.

Y finalmente al grupo de investigación OPALO, y a mi amada UIS por todos sus conocimientos brindados.

*Los sueños si se cumplen, el tiempo de Dios es perfecto y somos prueba de ello.*

*Dilan Jhoanny Mogollón Carreño*

**Tabla de contenido**

Introducción .....	11
1. Especificaciones del proyecto.....	13
1.1 Planteamiento del problema.....	13
1.2 Justificación del proyecto. ....	14
2. Objetivos .....	15
2.1 Objetivo general.....	15
2.2 Objetivos específicos .....	15
3. Revisión de literatura .....	16
3.1 Análisis Bibliométrico .....	16
3.2 Análisis de literatura. ....	23
3.2.1 Problema de formación de celdas .....	23
3.2.2 Problema generalizado de celdas cúbicas .....	26
3.2.3 Problemas de programación.....	27
3.2.4 Problema relacionado con el rol humano.....	29
3.2.5 Problema múltiple.....	30
3.3 Conclusiones de la revisión de literatura .....	32
4. Marco Teórico.....	35
4.1 Diseño de sistemas de producción .....	35
4.2 Metaheurísticas .....	35
4.2.1 Algoritmo genético .....	35
4.2.2 Optimización por Enjambre de Partículas (Particle Swarm Optimization-PSO) .	40
4.2.3 Optimización combinatoria.....	41
4.3 Sistemas de fabricación celular.....	41
4.4 Tecnología de grupos .....	41
4.5 Sistemas de manufactura.....	41
4.6 Asignación de operaciones y asignación de trabajadores .....	42
4.7 Planeación de la producción .....	42
4.8 Makespan .....	42

4.9	Complejidad computacional .....	43
5.	Modelo matemático .....	43
5.1	Descripción del modelo .....	43
5.2	Notación .....	44
5.2.1	Índices .....	44
5.2.2	Parámetros de entrada .....	44
5.2.3	Variables de decisión .....	45
5.2.4	Modelo matemático .....	45
6.	Calibración de parámetros .....	47
7.	Desarrollo de los algoritmos .....	50
7.1	Algoritmo Genético .....	51
7.1.1	Descripción del algoritmo .....	51
7.1.2	Funcionamiento del algoritmo .....	52
7.2	Algoritmo genético híbrido .....	58
7.2.1	Descripción del algoritmo .....	58
7.2.2	Funcionamiento del algoritmo .....	59
8.	Calibración de los algoritmos .....	63
8.1	Diseño $2^k$ para GA .....	64
8.2	Diseño $2^k$ GAPSO .....	65
8.2.1	Caso 1. Factores individuales: .....	67
8.2.2	Caso 2. Interacción de dos factores: .....	67
8.2.3	Caso 3. Interacción de tres factores: .....	68
8.2.4	Caso 4. Interacción de cuatro factores: .....	68
8.2.5	Decisión para GAPSO .....	68
9.	Comparación del desempeño de los algoritmos .....	68
10.	Conclusiones .....	73
11.	Recomendaciones .....	75
	Referencias bibliográficas .....	76

**Tabla de figuras**

Figura 1. Publicaciones por año (Últimos tres años) .....	18
Figura 2. Mapas de autoría e interrelación de las bases de datos. ....	19
Figura 3. Mapas de coocurrencia de las bases de datos. ....	20
Figura 4. Clasificación de modelos matemáticos en función de sus objetivos. ....	23
Figura 5. Problemas con más frecuencia en las publicaciones .....	32
Figura 6. ¿Qué busca solucionar el problema? .....	33
Figura 7. Métodos de solución más utilizados.....	34
Figura 8. Ecuación de velocidad PSO.....	40
Figura 9. Ecuación de actualización de Posición.....	41
Figura 10. Diagrama de flujo de GA .....	51
Figura 11. Estructura de Población inicial FTj .....	53
Figura 12. Diagrama de flujo para soluciones iniciales .....	54
Figura 13. Estructura de familia de variables $X_{jmwr}$ .....	55
Figura 14. Selección por torneo .....	56
Figura 15. Cruce de padres GA.....	56
Figura 16. Mutación por intercambio GA.....	57
Figura 17. Mutación por volteo de pin GA.....	57
Figura 18. Diagrama de flujo GAPSO .....	58
Figura 19. Mutación por intercambio .....	61
Figura 20. Mutación por volteo de pin.....	62
Figura 21. Cruce Individuo-Pbest GAPSO.....	62
Figura 22. Cruce hijo-Gbest GAPSO.....	63
Figura 23. Diseño 2k GA .....	64
Figura 24. ANOVA GA .....	64
Figura 25. Diseño 2k GAPSO.....	65
Figura 26. ANOVA GAPSO.....	66
Figura 27. Coeficientes GAPSO a partir de ANOVA .....	66
Figura 28. Prueba estadística T.....	70
Figura 29. Gráfico de medias prueba T .....	71

**Tablas**

Tabla 1. Cumplimiento de objetivos específicos .....	12
Tabla 2. Ecuaciones de Búsqueda.....	16
Tabla 3. Ecuación Combinada .....	17
Tabla 4. Número de resultados aplicando filtros. ....	18
Tabla 5. Autores más citados y con más publicaciones del campo de búsqueda .....	19
Tabla 6. Palabras clave del campo de búsqueda totalizado. ....	22
Tabla 7. Métodos de selección.....	36
Tabla 8. Tipos de operador de cruce.....	38
Tabla 9. Operadores de mutación .....	40
Tabla 10. Complejidad del problema.....	43
Tabla 11. Parámetros del modelo.....	47
Tabla 12. Parámetros calibrados .....	48
Tabla 13. Parámetros calibrados ajustados .....	49
Tabla 14. Parámetros fijos para los algoritmos.....	50
Tabla 15. Parámetros definidos para resultados .....	69
Tabla 16. Resumen parámetros para algoritmo .....	69
Tabla 17. Cantidad de variables involucradas .....	70
Tabla 18. Ejemplo de implementación de los algoritmos.....	72

### **Apéndices**

Los siguientes apéndices están adjuntos en la carpeta Apéndices, del enlace de drive:

Apéndice A. Revisión de literatura

Apéndice B. Resultados Matlab prueba grande

Apéndice C. Código Matlab PDF

Apéndice D. Código Matlab Editable

Apéndice E. Datos para diseños de experimento

Apéndice F. Diseño de experimento GA

Apéndice G. Diseño de experimento GAPSO

Apéndice H. Análisis de prueba T MINITAB

Apéndice I. Artículo Publicable

### Resumen

**Título del proyecto:** “Comparación del desempeño de dos algoritmos genéticos híbridos para el problema de formación de celdas de manufactura considerando el movimiento de los trabajadores (PFCMT)”.

**Autores:** Dilan Jhoanny Mogollón Carreño, Sebastián Elías Páez Becerra.

**Palabras clave:** Celdas de manufactura, Makespan, Algoritmo Genético GA, Optimización por Enjambre de Partículas PSO, Algoritmo Híbrido.

### Descripción:

En esta investigación se aborda el problema de formación de celdas de manufactura considerando el movimiento de trabajadores de forma intercelular e intracelular y recursos limitados, tratándose de una problemática expuesta por los autores Jufeng Wang, Chunfeng Liu & Kai Li (2019), la cual se centra en la reducción del tiempo de procesamiento (Makespan), considerando la asignación de tareas y formación de celdas de manufactura. Debido a que no cuenta con una solución exacta, se adaptan dos Metaheurísticas basadas en un Algoritmo evolutivo para la búsqueda de soluciones, en primer lugar, se trabaja el Algoritmo Genético (GA) sin ningún tipo de alteración, seguido se plantea un GA híbrido, uniendo GA con un Optimización por Enjambre de Partículas (PSO), para obtener datos de los dos algoritmos se programa el modelo y las metaheurísticas en el software Matlab con versión de 2022a. Mediante pruebas se realizan dos diseños de experimento factorial del tipo 2K para determinar qué factores inciden de manera significativa sobre cada algoritmo. Con estos factores definidos se corre el modelado, se analiza mediante una prueba T los resultados obtenidos para finalmente dar conclusiones respecto a los resultados y observaciones a lo largo de la investigación.

---

\* Trabajo de grado

\*\* Facultad de Ingeniería Físico-Mecánicas. Escuela de Estudios Industriales y Empresariales. Programa de Ingeniería Industrial. Director: M. Sc. Laura Yeraldín Escobar Rodríguez. Codirector: M. Sc. Edwin Alberto Garavito Hernández.

**Abstract**

**Title:** “Comparison of the performance of two hybrid genetic algorithms for the manufacturing cell formation problem considering the movement of workers (PFCMT)”.

**Authors:** Dilan Jhoanny Mogollón Carreño, Sebastián Elías Páez Becerra.

**Key Words:** Manufacturing cells, Makespan, GA Genetic Algorithm, PSO Particle Swarm Optimization, Hybrid Algorithm.

**Description:** This research addresses the problem of manufacturing cell formation considering the movement of workers intercellularly and intracellularly and limited resources, being a problem exposed by the authors Jufeng Wang, Chunfeng Liu & Kai Li (2019), which is focuses on the reduction of processing time (Makespan), considering the assignment of tasks and the formation of manufacturing cells. Because it does not have an exact solution, two Metaheuristics based on an evolutionary Algorithm are adapted to search for solutions, first, the Genetic Algorithm (GA) is worked without any type of alteration, followed by a hybrid GA, joining GA with a Particle Swarm Optimization (PSO), to obtain data from the two algorithms, the model and the metaheuristics are programmed in the Matlab software with version 2022a. Through tests, two 2K-type factorial experiment designs are carried out to determine which factors have a significant impact on each algorithm. With these defined factors, the modeling is run, the results obtained are analyzed by means of a T test to finally give conclusions regarding the results and observations throughout the investigation.

---

\* *Degree work*

\*\* *Faculty of Physical-Mechanical Engineering. School of Industrial and Business Studies. Industrial Engineering Program. Director: M. Sc. Laura Yeraldín Escobar Rodríguez. Co-director: M. Sc. Edwin Alberto Garavito Hernández.*

## Introducción

En las últimas décadas el desarrollo industrial ha desencadenado una cuarta revolución industrial, las expectativas altas de un cliente cada vez más informado y el incremento en la competencia global debido a la globalización, generan un entorno económico altamente competitivo y con extremas dificultades para la supervivencia empresarial. Para sobrevivir en este entorno incierto, las organizaciones deben gestionar estrategias eficaces y eficientes de planificación y programación de la producción (Salman Khalid, Azim, Abas, Rehman Babar, & Ahmad , 2021) Una de las estrategias más utilizadas es la implementación de sistemas de fabricación basadas en la tecnología de grupo, donde su objetivo principal es analizar un sistema complejo en subsistemas. La manufactura celular es una de las aplicaciones que más tiene relevancia en la tecnología de grupos, en esta aplicación se busca procesar grupos de piezas (familias de productos) con los mismos requisitos (dimensiones y operaciones) en grupos de máquinas (células o celdas). Además, esta composición otorga al fabricante ventajas como reducción en tiempos de preparación, inventarios en proceso y número total de máquinas; mejora en el flujo del proceso y programación de operaciones; y la ventaja más importante es el aumento en la flexibilidad del proceso de producción que busca satisfacer la demanda volátil del cliente (Hosseinabad & Zaman, 2020).

Teniendo en consideración la importancia de la manufactura celular, este documento se centra el problema presentado por Jufeng Wang, Chunfeng Liu & Kai Li (2019) el cual expone la programación de tareas y la formación de células integradas en un sistema de manufactura celular para entornos con recursos duales limitados considerando el movimiento de los trabajadores. Los autores plantean para su solución un algoritmo híbrido en el que unen un algoritmo heurístico basado en reglas de prioridad (PRBHA) y el algoritmo de recursividad directa revisado (RFRA) con el recocido simulado convencional (SA). Ahora bien, el presente trabajo se enfoca en buscar y diseñar dos métodos de solución para el problema anteriormente mencionado, para tal fin se pretende usar dos metaheurísticas basadas en algoritmo evolutivo, estas son Algoritmo Genético (GA) y Algoritmo Genético Híbrido unido con Optimización de Enjambre de Partículas Discreto (DPSO). Los métodos de solución son implementados en el software MATLAB y mediante análisis estadístico se presentan conclusiones respecto a los modelos y los resultados obtenidos por estos.

La finalidad de este documento es aportar conocimiento al área de diseño de sistemas productivos con la documentación y conclusión de resultados para que futuros trabajos se realicen en la temática de makespan y celdas de manufactura. A en la *Tabla I* se presentan los objetivos, además, de la localización del cumplimiento de estos.

**Tabla 1.***Cumplimiento de objetivos específicos*

<b>OBJETIVO ESPECÍFICO</b>	<b>CAPÍTULO</b>
Identificar heurísticas y metaheurísticas más utilizadas para el problema de formación de celdas de manufactura con entorno restringido de tareas y recursos duales mediante una revisión bibliográfica.	<b>3</b>
Diseñar dos métodos de solución basados en el algoritmo genético e implementarlos en el software Matlab.	<b>7</b>
Evaluar el desempeño de los algoritmos mediante un análisis estadístico de los resultados.	<b>8 y 9</b>
Elaborar un artículo de carácter publicable con el análisis, resultados y conclusiones del trabajo de investigación.	<b>Apéndice I</b>

---

## **1. Especificaciones del proyecto**

### **1.1 Planteamiento del problema.**

Los tiempos de finalización de tareas (makespan) son un aspecto esencial para verificar el desempeño dentro de los procesos de una organización que pretende posicionarse como una de las mejores en un ambiente empresarial altamente competitivo; sin embargo, para tal fin se deben considerar estrategias, herramientas y técnicas que junto con un horizonte de objetivos bien definidos impulsan a la culminación de sus tareas. Dentro de estas estrategias para disminuir el makespan se encuentran: la reconfiguración del sistema productivo en celdas de manufactura que consiste en una configuración de máquinas de trabajo por familia de productos o agrupación de tareas. Por otro lado se tiene la distribución de planta que resulta útil una configuración específica dependiendo del tipo de proyecto que se desee emplear, estos, son temas que han tomado más fuerza en los últimos años debido a su gran aporte al área de producción en las empresas; lo descrito con anterioridad, sumado con simulaciones que intentan ser lo más precisas en torno a la representación de la realidad hacen que se pueda tener una perspectiva un poco más certera al momento de intentar reducir el al máximo el tiempo de procesamiento, considerando movimientos de diferentes tipos y aspectos como número de trabajadores, maquinas, tamaño y numero de celdas, jornadas laborales y entre otros componentes son las vertientes que definen el tema tratado.

Dentro del contexto expresado con anterioridad se presenta un problema de formación de celdas en el cual intervienen múltiples factores restringidos que giran en torno al makespan. El problema es expuesto por los autores Jufeng Wang, Chunfeng Liu & Kai Li (2019) y se centra en reducir el tiempo de finalización de tareas que puedan existir dentro de un proceso productivo que considera movimiento de empleados dentro y fuera de las celdas, a esto se añaden parámetros y restricciones que buscan representar de manera exacta la realidad.

Considerando la dificultad y complejidad de la problemática se plantea el estudio de métodos de solución robustos y la implementación de herramientas computacionales como MATLAB, que mediante la codificación del problema otorguen configuraciones factibles entre tareas, celdas, maquinas, trabajadores y jornadas laborales. Para posteriormente, alcanzar la solución óptima del problema.

## **1.2 Justificación del proyecto.**

La modelización matemática es una herramienta fundamental para la comprensión y estudio de fenómenos que impactan la realidad, su principal papel como mediador entre una teoría y el mundo real deja como resultado modelos con diversos grados de complejidad (Cervantes Gómez, 2015). Ahora bien, el problema de formación de celdas de manufactura presentado por Jufeng Wang, Chunfeng Liu & Kai Li (2019) genera un modelo matemático con una complejidad valorada como NP-hard, por consiguiente, encontrar la solución óptima sin el uso de herramientas computarizadas se vuelve una tarea inalcanzable.

Este es el punto de partida del presente proyecto, el cual busca facilitar el desarrollo del modelo anteriormente expuesto, teniendo como propósito la exploración de nuevas alternativas que incidan de manera positiva al momento de encontrar una respuesta, esto es bajo parámetros como el número de variables, tiempos computacionales entre muchas otras consideraciones.

Es importante ver como la formación de celdas ayuda en la disminución del makespan ya que, al encontrarse una configuración adecuada, minimizará tiempos de recorrido no solo de producto en proceso y terminado sino también desplazamiento de mano de obra, generándose así una posible mejora, sin embargo, también trae beneficios en cuanto a calidad puesto que los centros serán enfocados a un proceso o producto en específico.

Los autores buscan implementar métodos de solución metaheurísticos que permitan llegar a la solución óptima. Luego, mediante un análisis estadístico de resultados y métricas se definen los parámetros de corrida del modelo para de esa forma determinar que método de solución obtiene el mejor rendimiento en cuanto a la calidad de la respuesta y establecer conclusiones. Con las conclusiones obtenidas se pretende aportar sobre el trabajo de Jufeng Wang, Chunfeng Liu & Kai Li (2019).

## 2. Objetivos

En el presente capítulo se detallan los objetivos definidos para el desarrollo del trabajo de grado.

### 2.1 Objetivo general

Comparar el desempeño de dos metaheurísticas basadas en algoritmo genético e híbridos para el modelo de optimización del sistema de fabricación celular con entorno restringido de tareas y recursos duales que considera el movimiento de los trabajadores.

### 2.2 Objetivos específicos

- ✓ Identificar heurísticas y metaheurísticas más utilizadas para el problema de formación de celdas de manufactura con entorno restringido de tareas y recursos duales mediante una revisión bibliográfica.
- ✓ Diseñar dos métodos de solución basados en el algoritmo genético e implementarlos en el software Matlab.
- ✓ Evaluar el desempeño de los algoritmos mediante un análisis estadístico de los resultados.
- ✓ Elaborar un artículo de carácter publicable con el análisis, resultados y conclusiones del trabajo de investigación.

### 3. Revisión de literatura

En el presente numeral se documenta el análisis bibliométrico seguido de su respectivo análisis preliminar de literatura y sus conclusiones.

#### 3.1 Análisis Bibliométrico

En la presente sección, se busca indagar y estar al tanto de investigaciones relacionadas con la temática Sistemas de Manufactura Celular. Además, se pretende identificar modelos matemáticos y métodos de solución (heurísticas y metaheurísticas) utilizados en el problema de formación de celdas de manufactura. Para ello, se formulan ecuaciones de búsqueda con las temáticas anteriormente mencionadas y se ingresan en las bases de datos disponibles en la biblioteca virtual de la Universidad Industrial de Santander. Para esta investigación se seleccionan las bases de datos Scopus y Web of Science debido a la cantidad y confiabilidad de la información que ofrecen. En la *Tabla 2* se observa la temática principal de cada ecuación, la ecuación formulada y el número de publicaciones encontradas en cada base de datos.

**Tabla 2.**

*Ecuaciones de Búsqueda*

BASE DE LA ECUACIÓN	ECUACIÓN	# PUBLICACIONES EN SCOPUS	# PUBLICACIONES EN WEB OF SCIENCE
Problema de celdas de manufactura	( <i>TITLE-ABS-KEY</i> ("cell* manufacturing system*") OR <i>TITLE-ABS-KEY</i> (" cell* formation problem"))	1.640	654
Modelo matemático	( <i>TITLE-ABS-KEY</i> ("math* model") OR <i>TITLE-ABS-KEY</i> ("math* programming"))	1.295.894	112.232
Método de solución	( <i>TITLE-ABS-KEY</i> ("Algorithm*") OR <i>TITLE-ABS-KEY</i> ("Heuristic*") OR <i>TITLE-ABS-KEY</i> ("Metaheuristic*"))	3.331.771	1.179.610

En primera instancia, se puede observar que la búsqueda da como fruto un número de publicaciones alto, dificultando la revisión de todos los resultados. Lo anterior se da considerando la búsqueda independiente. Por ende, se decide reducir el campo de búsqueda mediante la combinación de las tres ecuaciones, generando una menor cantidad de obtenidos y asociados directamente a las tres temáticas requeridas para esta investigación. Lo anterior se presenta en la *Tabla 3*.

**Tabla 3.***Ecuación Combinada*

ECUACIÓN COMBINADA	# RESULTADOS EN SCOPUS	# RESULTADOS EN WEB OF SCIENCE
( <i>TITLE-ABS-KEY</i> ("cell* manufacturing system*") OR <i>TITLE-ABS-KEY</i> (" cell* formation problem") AND <i>TITLE-ABS-KEY</i> ("math* model") OR <i>TITLE-ABS-KEY</i> ("math* programming") AND <i>TITLE-ABS-KEY</i> ("Algorithm*") OR <i>TITLE-ABS-KEY</i> ("Heuristic*") OR <i>TITLE-ABS-KEY</i> ("Metaheuristic*"))	295	180

En esta segunda búsqueda se destaca la diferencia en el número de publicaciones resultantes entre cada base de datos, obteniendo un 63,8% más de publicaciones en Scopus. Al realizar un análisis inicial de los títulos, autores y fecha de publicación, se encuentran algunas publicaciones que pertenecen a áreas de conocimiento que no se relacionan con la temática de investigación. En consecuencia, se decide emplear los siguientes filtros:

- **Limitado a tiempo:** En el caso del tiempo de publicación, se decide delimitar los resultados de búsqueda por igual en las bases de datos y solo seleccionar aquellos que su año de publicación entra en los años más recientes (2019,2020 y 2021).
- **Limitado a área temática:** En el caso de los temas tratados en cada publicación, para Scopus se limitan los resultados a las áreas temáticas de Engineering (Ingeniería) y Mathematics (matemáticas) y para Web of Science (ISI) se limitan los resultados a las áreas temáticas de Engineering Industrial (Ingeniería Industrial) y Operations Research Management Science (Ciencias de la gestión de la investigación de operaciones).

- **Limitado a tener un modelo matemático:** Se limitan los resultados a publicaciones que se desarrollan mediante la implementación de un modelo matemático para la solución de una problemática vinculada a los sistemas de manufactura celular.

En la *Tabla 4* se observa la cantidad de publicaciones resultantes luego de aplicar los filtros en cada base de datos.

**Tabla 4.**

*Número de resultados aplicando filtros.*

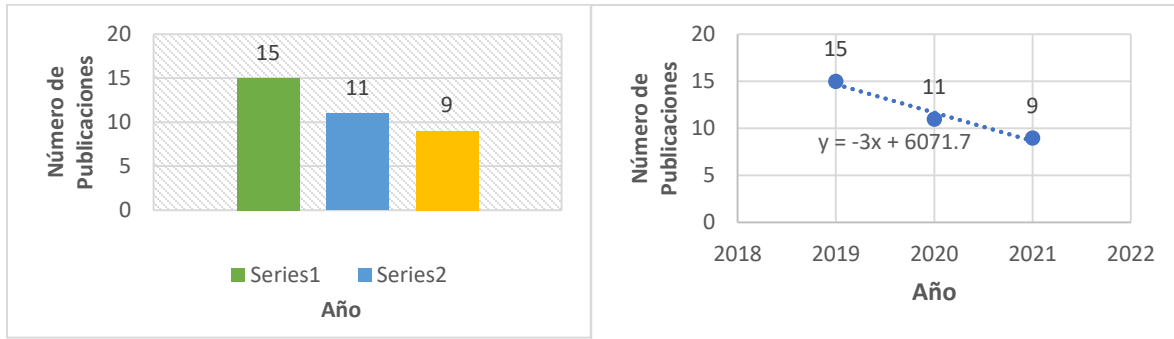
BASE DE DATOS	# RESULTADOS FILTRADOS
SCOPUS	24
WEB OF SCIENCE	11
TOTAL	35

Ahora bien, para continuar con la etapa de selección de documentos a analizar, se eliminan las publicaciones que se encuentren repetidas en las bases de datos para empezar a manejar un campo consolidado de documentos. En el *Apéndice A* se evidencia el número de publicaciones, su título, autores, fecha de publicación y palabras clave.

Luego de aplicar los filtros mencionados, se consolida un total de 35 documentos a analizar. En primera instancia se analiza la fecha de publicación, obteniendo como resultante la *Figura 1* que representa el número de publicaciones realizadas por año. Es evidente una tendencia decreciente de publicaciones y según el artículo de Estrada, Medino, Rebollo, Campos & Primo (2021), esto se debe a que miles de autores indiferente de su área de investigación, centran su interés específicamente en apoyar y abordar la crisis sanitaria, generando desatención hacia otras temáticas.

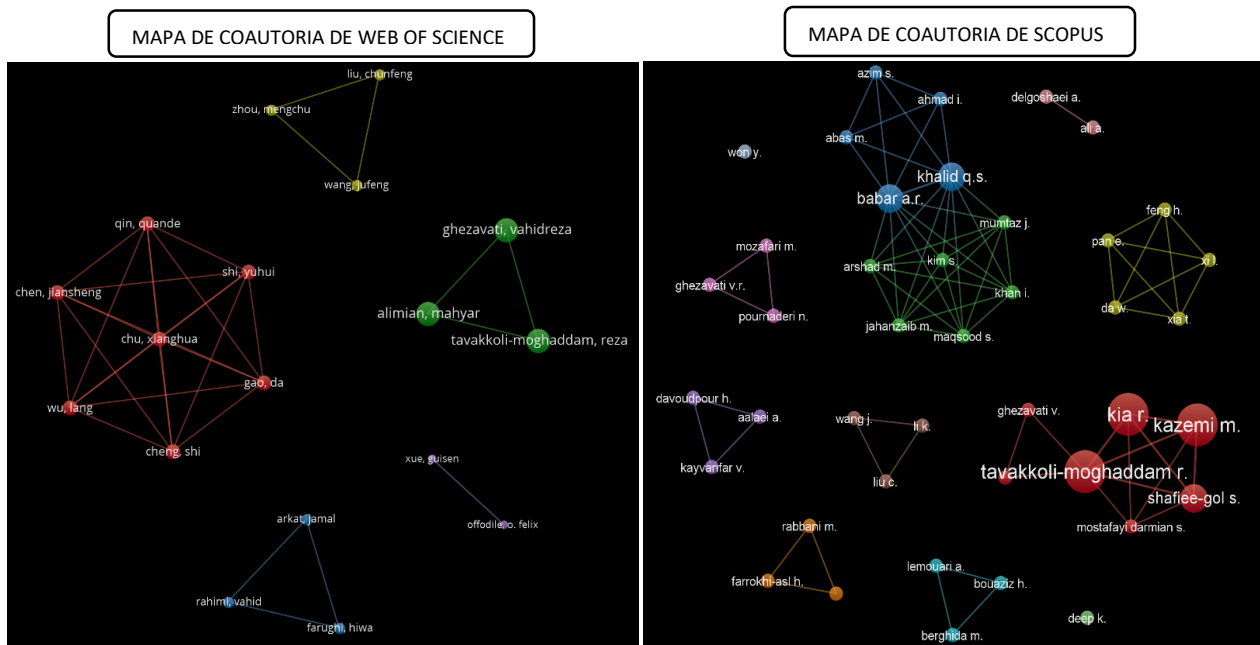
**Figura 1.**

*Publicaciones por año (Últimos tres años)*



Pese a tener un número reducido de publicaciones al transcurrir los años, se procede a analizar los autores encontrados en estas publicaciones. En la *Figura 2* se pueden observar los mapas de autoría e interrelación del campo de búsqueda totalizado (resultados de Scopus y Web of Science) realizados en el software VOSviewer.

**Figura 2.**  
Mapas de autoría e interrelación de las bases de datos.



De igual forma, se analizan los autores más citados y con más publicaciones en el campo de búsqueda consolidado (ver *Tabla 5*). Con estos resultados se puede buscar continuidad en las publicaciones de cada autor y distinguir si los documentos se relacionan y profundizan en una misma temática o por el contrario son temáticas independientes, modelos matemáticos distintos o métodos de solución diferentes.

**Tabla 5.**  
Autores más citados y con más publicaciones del campo de búsqueda.

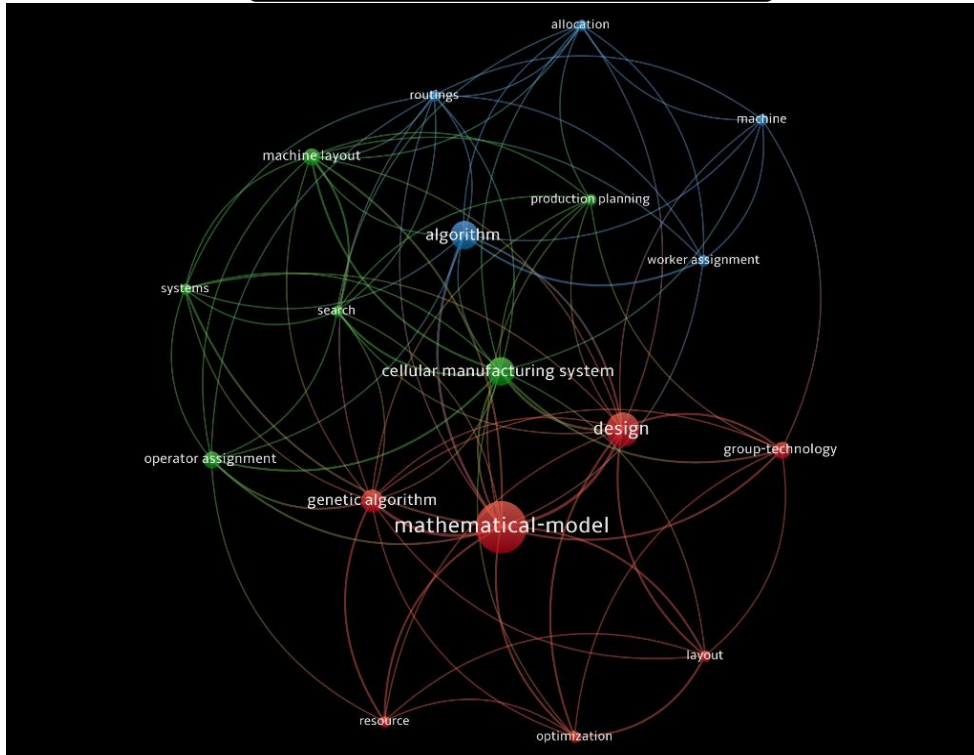
NOMBRE DEL AUTOR	# DE DOCUMENTOS PUBLICADOS	# DE CITACIONES
Tavakkoli-Moghaddam, R.	4	12
Khalid, QS.	2	13
Babar, AR.	2	13
Xi, L.	1	11
Xia, T.	1	11
Pan, E.	1	11
Feng, H.	1	11
Da, W.	1	11
Alimian, Mahyar	1	11
Ghezavati, Vahidreza	1	11

Por consiguiente, se realiza un análisis de palabras clave con dos finalidades, primordialmente se busca validar la precisión de las ecuaciones de búsqueda a través de la congruencia de resultados y como segundo objetivo, obtener una base de ideas e interrelaciones entre temas para el marco teórico que será presentado más adelante. En la *Figura 3* se pueden observar los mapas de coocurrencia de palabras clave del campo de búsqueda totalizado, este mapeo tiene en cuenta palabras clave del autor y del índice, esta figura se realizó en el software VOSviewer.

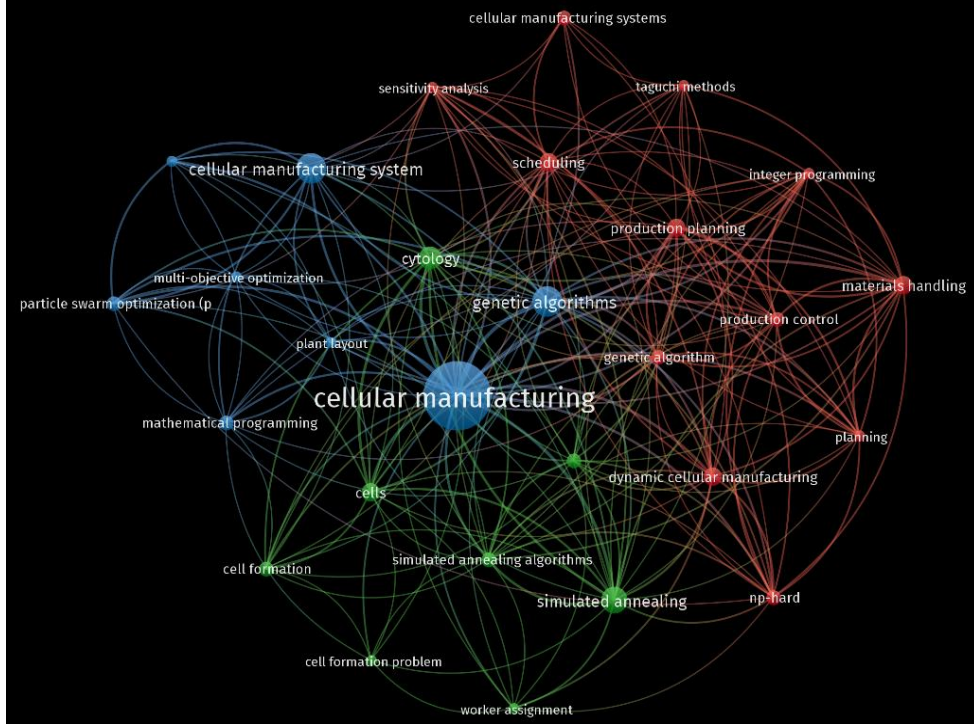
**Figura 3.**

*Mapas de coocurrencia de las bases de datos.*

MAPA DE COOCURRENCIA DE WEB OF SCIENCE



MAPA DE COOCURRENCIA DE SCOPUS



Como resultado se tiene la *Tabla 6*, en la que se presentan las palabras clave que se encuentran en cinco o más de documentos publicados.

**Tabla 6.**

*Palabras clave del campo de búsqueda totalizado.*

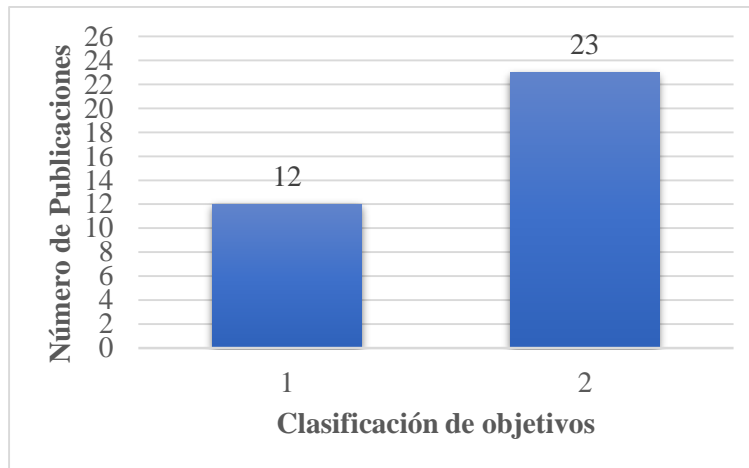
PALABRA CLAVE	# DE DOCUMENTOS PUBLICADOS
Cellular Manufacturing	18
Cellular Manufacturing System	13
Mathematical-model	9
Genetic Algorithms	8
Simulated annealing	7
Desing	6
Cytology	6
Algorithm	5
Dynamic Cellular Manufacturing System	5
Materials Handling	5
Production Planning	5
Cells	5
Scheduling	5

Con estos resultados se puede concluir que todas las palabras clave están relacionadas con las ecuaciones de búsqueda formuladas.

Ahora bien, se realiza un análisis del tipo de modelo matemático utilizado en cada documento, lo anterior con el fin de definir si el modelo es mono-objetivo o multi-objetivo. Los resultados se presentan en la *Figura 4*.

**Figura 4.**

*Clasificación de modelos matemáticos en función de sus objetivos.*



Se evidencia en la *Figura 4* una relación aproximada de 2 a 1 entre las publicaciones que tienen un modelo multi-objetivo y aquellas que tienen un modelo mono-objetivo. Esto se debe a la adaptabilidad que posee un modelo multi-objetivo con las problemáticas del mundo real. Ahora bien, el inconveniente de optimizar un modelo multi-objetivo es encontrar aquella solución que proporcione valores aceptables para todas las funciones objetivo, esto implica mayor complejidad en la programación y tiempos de cómputo más extensos (Ciatera Díaz, 2015).

### **3.2 Análisis de literatura.**

El concepto de sistemas de manufactura celular (CMS) se relaciona directamente y de forma estrecha con la tecnología de grupos, la cual es expuesta por primera vez en 1959 por el científico soviético Sergei Petrovich Mitrofanov. Para Mitrofanov, la tecnología de grupos es un proceso de fabricación basado en la clasificación de piezas con atributos y especificaciones similares dentro de grupos llamados familias que posteriormente se les realiza un proceso de operaciones con métodos definidos y programados (Cordovés García, Sanzano Tamayo, Lastre Aleaga, & Ávila Rondón, 2016)

Surgen entonces problemáticas relacionadas con el tema CMS, a los cuales los investigadores buscan modelar y solucionar. Para efectos del proyecto y basados en el campo de búsqueda totalizado se clasifican cinco problemáticas principales:

#### **3.2.1 Problema de formación de celdas**

Según el póster de Escobar, Garavito & Talero (2017), el problema de formación de celdas (CFP) es aquel que independiente de ciertos factores busca formar familias de productos de

acuerdo con similitud de procesos y características físicas para luego agrupar máquinas disponibles en celdas y finalmente incorporar las familias de productos en las celdas formadas. A continuación, se presentan todos los artículos presentes en el campo de búsqueda totalizado los cuales su temática a resolver es el CFP.

A principios de 2019 es publicado un artículo por Jufeng Wang, Chunfeng Liu & Kai Li (2019), donde se presenta un novedoso modelo centrado en un CMS con entorno restringido de prioridad de tareas y recursos duales (Máquina/Trabajador), como resultante se obtiene un modelo lineal entero mixto que busca minimizar el tiempo máximo de finalización de todas las tareas (makespan) y tiene presente en su modelado el movimiento de los trabajadores. Al ser un problema complejo, los autores implementan como método de solución un algoritmo de recocido simulado (SA) combinado con una heurística basada en reglas de prioridad (PRBHA) y el algoritmo de recursividad directa revisado (RFRA).

A mediados de 2019 Pournaderi, Ghezavati & Mozafari (2019), publican un artículo donde dentro de su modelo implementa el problema de disposición de instalaciones estáticas (SFLP) y el problema de disposición dinámica de instalaciones (DFLP). Se obtiene un modelo lineal en el cual se busca minimizar la suma de costos asociados con manejo de material y transporte, como método de solución se utilizan dos algoritmos metaheurísticos combinados, un algoritmo genético (GA) y un recocido simulado (SA).

A finales de 2019 se publican tres artículos bastantes distintivos, el primero es publicado por Raoofpanah, Ghezavati & Tavakkoli-Moghaddam (2019), su modelo se centra en la formación de celdas verdes las cuales se relacionan con los problemas medioambientales causados por los modos de producción y transporte. Como resultado se desarrolla un modelo no lineal entero mixto buscando minimizar el costo total compuesto por movimientos intracelulares, reformas de máquinas, fabricación de piezas, mantenimiento, pedidos pendientes y contaminación. Debido a su complejidad se utiliza un algoritmo de descomposición de Benders para su solución. En segundo lugar, los autores Shunmugasundaram, Anbumalar, Anand, Sivakumar & Nagarajan (2019) publican un modelo donde se plantea que la calidad de la solución depende de la matriz incidencia de la máquina de piezas inicial. Para ello, crean un algoritmo nuevo como método de solución que combina la técnica de agrupación basada en matrices y agrupación aglomerativa. Y finalmente, los autores Golmohammadi, Honarvar, Tian & Hosseini-Nasab (2019) publican un modelo donde a través de un diseño intracelular se puede diseñar la máquina y la celda con un proceso de

enrutamiento. Como resultante se tiene un modelo no lineal entero mixto en el que se busca minimizar el costo total de reubicación de piezas y maximizar la confiabilidad de las rutas de procesamiento intra e intercelulares. Como método de solución se implementan tres algoritmos y se concluye cual otorga el mejor resultado, estos algoritmos son: algoritmo genético (GA), algoritmo de Keshtel (KA) y algoritmo red deer (RDA).

A principios de 2020 se publican dos artículos, en primer lugar, está el autor Deep (2020) con su modelo de formación de celdas considerando el entorno dinámico al realizar producción con subcontratación. Se obtiene un modelo que permite un espacio de búsqueda cúbico de máquina-parte-trabajador y se soluciona con un algoritmo genético basado en recocido simulado (SAGA). En segundo lugar, se encuentra la publicación realizada por Mohtashami, Alinezhad & Niknamfar (2020) en la que exponen un modelo centrado en la flexibilidad en la asignación de máquinas. Como resultado se tiene un modelo que busca minimizar el costo del sistema debido a la cantidad de pérdida de producción a casa del desperdicio y por otra parte minimizar la variación de los costos difusos. Para su solución se implementan dos metaheurísticas, el primero es un algoritmo genético (GA) de clasificación no dominante-II y como segundo un algoritmo de optimización por enjambre de partículas (PSO).

A mediados del 2020 se publica el artículo de Alduaij & Hassan (2020), en el cual presenta un modelo centrado en el manejo de materiales mediante vehículos y bucles múltiples. Se obtiene un modelo lineal entero mixto que busca minimizar el costo total de producción además se toma como un modelo novedoso al integrar el diseño experimental con el modelado computacional y el análisis estadístico. Para su solución se implementa un algoritmo de distancia de la máquina dependiente del diseño.

A finales de 2020 se publican dos artículos, el primero es publicado por Ibrahim & Jarboui (2020), donde exponen un modelo altamente relacionado con la agrupación de máquinas y familias de piezas, se crea entonces un modelo p-mediana modificado que busca maximizar la suma de los coeficientes de disimilitud entre máquinas, de modo que una máquina debe asignarse a una sola celda. Como método de solución implementan un algoritmo de búsqueda de vecindad de variable general (VNS) y el algoritmo de estimación de distribución. También se realiza una publicación por parte de Lamba, Kumar, Mishra & Rajput (2020), en el cual describen un modelo que se centra en la distribución de la instalación celular dinámica, la eficiencia energética, la productividad de los recursos y la creación de valor sostenible. Como resultado se obtiene un modelo no lineal

entero mixto que busca minimizar el costo de manejo de materiales y costo de reorganización para ambos movimientos inter/intracelular. Además, el consumo total de energía eléctrica y el flujo de material agregado entre máquinas y para su solución se utiliza un algoritmo de recocido simulado (SA).

A finales del 2021 se publican tres artículos bastante distintivos entre ellos, en primer lugar, Mansour, Afefy & Taha (2021) publican y exponen un modelo centrado en la asignación de máquinas donde se considera la secuencia de operaciones, las rutas alternativas y el tiempo disponible de máquina. Además, presenta la asignación de trabajadores en función de su desempeño real. Obtiene como resultado un modelo no lineal que busca minimizar los costos como: costos de movimiento de material, costos de operación y costos de defectos teniendo en cuenta desechos y reprocesos, para su solución optan por crear un algoritmo heurístico. En segundo lugar, tenemos la publicación realizada por Shafiee-Gol, Kia, Kazemi, Tavakkoli-Moghaddam & Mostafayi Darmian (2021), donde entregan un modelo que además de tener en cuenta el problema de formación de celdas dinámicas, también incorpora el problema de planificación de la producción (PP) y el problema de ubicación y asignación (LA), esto se debe a que buscan la fabricación celular en múltiples plantas. Como resultado surge un modelo no lineal entero mixto que busca maximizar la utilidad teniendo en cuenta los ingresos obtenidos en ventas y muchos costos de operación del sistema. Por su complejidad se utilizan dos metaheurísticas como métodos de solución, el primero es un algoritmo optimizador de lobo gris (GWO) y segundo un algoritmo genético (GA). Y finalmente, los autores Lashgari, Kia & Jolai (2021), publican y exponen un modelo centrado en el entorno dinámico que incorpora el diseño de grupos y asignación de trabajadores. Obtienen un modelo que busca minimizar los tiempos de procesamiento y la incertidumbre encontrada en las demandas. Para su solución utilizan un algoritmo de recocido simulado (SA).

### ***3.2.2 Problema generalizado de celdas cúbicas***

Según el artículo de Bouaziz, Berghida & Lemouari (2020), el problema generalizado de celdas cúbicas (GCCFP) es una variante de la CFP donde cada pieza debe procesarse de acuerdo con un plan de proceso único. Por ende, debe optar por máquinas que solo procesen piezas si se encuentran en ruta demarcada por el plan y además especificar los trabajadores que operan en la máquina, creando una relación pieza-máquina-trabajador. A continuación, se presentan todos los

artículos presentes en el campo de búsqueda totalizado los cuales su temática a resolver es el GCCFP.

A inicios de 2019 los autores Rabbani, Farrokhi-Asl & Ravanbakhsh (2019), publican un artículo en el que exponen un modelo que considera la fiabilidad de la máquina y las posibles rutas en caso de una falla. Como resultante se obtiene un modelo lineal multiobjetivo que busca minimizar costos asociados al sistema de producción, optimizar mano de obra y minimiza la varianza de carga de trabajo entre diferentes celdas. Para su solución se utilizan dos metaheurísticas, el primero es un algoritmo genético de clasificación no dominada (NSGA) y como segundo se opta por un algoritmo de optimización de enjambre de partículas (PSO).

A mediados de 2019 es publicado un artículo por parte de Behnia, Mahdavia, Shirazi & Paydar (2019), en el que presentan un modelo bi-objetivo y bi-nivel donde se expone una matriz tridimensional (pieza, máquina y trabajador). Se obtiene un modelo lineal que busca minimizar el número total de vacíos (inactividad) y equilibrar cargas de trabajo al maximizar el interés de los trabajadores. Por su complejidad se implementan un algoritmo recocido simulado basado en población (NBI-PBSA) y un algoritmo de optimización por enjambre de partículas modificado (NBL-PS).

A mediados de 2020 los autores Bouaziz, Berghida & Lemouari (2020), presentan su modelo centrado en los movimientos entre celdas (variedad de rutas) y la heterogeneidad de las celdas al agrupar máquina, pieza y trabajador. Su resultado es un modelo lineal que busca minimizar costos de manejo de material y movimiento de trabajadores. Por otra parte, busca maximizar el índice de calidad considerando habilidades del trabajador. Para su solución se implementa un algoritmo de polinización de flores discretas (DFPA) que es una combinación de Optimización de Enjambre de Partículas Discretas (DPSO) y recocido simulado (SA), Además, implementa un algoritmo genético (GA).

### ***3.2.3 Problemas de programación***

En esta sección se presentan dos tipos de problemas de programación. Primordialmente se presenta el problema de programación de productos que es el más presente en las publicaciones. Según el libro de Nogueira River, Hernández Nariño, Oviedo-Rodríguez & Medina Nogueira (2014), define al problema de programación de productos como un problema de secuenciación y asignación, estas problemáticas se solucionan al asignar la cantidad correcta de tiempo y recursos

en el orden preciso. A continuación, se presentan todos los artículos presentes en el campo de búsqueda totalizado los cuales su temática a resolver el problema de programación de producción.

A mitad de 2019, es publicado un artículo por los autores Khalid, Arshad, Maqsood, Jahanzaib, Babar, Khan, Mumtaz & Kim (2019), donde presentan un modelo centrado en integrar la planificación de procesos y la programación de trabajos con técnicas evolutivas. Como resultado se obtiene un modelo no lineal que busca maximizar la utilización de la máquina y minimizar la cantidad de inventario de trabajo en proceso (WIP). Para su solución utilizan un algoritmo de optimización de enjambre de partículas híbrido con Nawaz, Enscore, Ham-NEH (NEPSO).

A principios de 2021, los autores Hong, Zeng & Gao (2021) publican y exponen un modelo que trata la asignación de cada trabajo a las celdas y controla el consumo de energía, el ancho de banda (intervalo de producción) y la distancia de manipulación. Se obtiene un modelo no lineal que busca minimizar el consumo de energía relacionado con configuración y procesamiento, la distancia total de manipulación y el tiempo máximo de finalización de todas las tareas (makespan). Debido a su complejidad se utiliza para la solución un algoritmo de evolución multiobjetivo mejorado basado en la descomposición con estrategia de búsqueda local (IMOEA /D-SL).

A mediados de 2021 se publican dos artículos, en primer lugar, los autores Shafiee-Gol, Kia, Tavakkoli-Moghaddam, Kazemi & Kamran (2021), entregan un modelo no lineal entero mixto que realiza una optimización combinatoria donde se integra la programación de piezas, la planificación de necesidades de material (MRP), planificación de producción (PP) y la planificación de transporte (TP). Para su solución se implementa un algoritmo de recocido simulado (SA). En segundo lugar, se realiza una publicación por parte de Khalid, Azim, Abas, Babar & Ahmad (2021), en el que presentan un modelo que busca programar productos agrícolas. Se obtiene un modelo lineal entero mixto que busca minimizar el trabajo en proceso (WIP) y maximizar la utilización promedio de la celda. Para su solución se implementa un algoritmo de optimización de enjambre de partículas modificadas (MPSO).

A finales de 2021, se realiza la publicación de dos artículos totalmente distintivo, primero los autores Salimpour, Pourvaziri & Azab (2021), entregan un modelo que se centra en organizar instalaciones en áreas desiguales y se pueden cambiar ubicaciones de las celdas de un periodo a otro. Como resultante surge un modelo lineal que busca minimizar el manejo de materiales totales y la disimilitud entre máquinas. Para su solución se emplea un algoritmo genético de clasificación no dominado modificado (MNSGA-II). En segundo lugar, se realiza la publicación de un modelo

novedoso por parte de Almasarwah & Suer (2021), en el que su modelo se centra en el problema de cuello de botella cambiante. Se obtiene un modelo lineal que busca minimizar el tiempo máximo de finalización de todas las tareas (makespan) y la disimilitud entre productos. Para su solución se implementa un método cíclico por lotes.

Ahora bien, como segunda problemática se presenta el problema de programación de celdas virtuales (VCMS), según el artículo de Zandieh (2019), el VCMS es una estrategia moderna para el diseño de instalaciones donde las máquinas están ubicadas en diferentes posiciones y las celdas virtuales son agrupaciones lógicas de máquinas, trabajos y trabajadores centradas en el control de la producción. Además, en este artículo se presenta un modelo lineal centrado en los tiempos de producción y la distancia total de transporte a través de la minimización de costos de producción, incluido el WIP, costos de mantenimiento de inventario, consumo de energía y movimientos de trabajadores. Para su solución se implementan tres algoritmos, donde se mide y se compara la calidad de las respuestas, como primera opción se tiene un algoritmo genético (GA), segundo un algoritmo basado en biogeografía (BBO) y finalmente un algoritmo híbrido entre BBO y GA.

### ***3.2.4 Problema relacionado con el rol humano***

Según el artículo de Abdullah, Abdul Rahman & Salleh (2019), se define al problema relacionado con el rol humano como una problemática de interés y conceptualización que genera un vacío en el campo de la investigación cuantitativa que se relaciona con los efectos que genera los componentes humanos en un CMS. A continuación, se presentan todos los artículos encontrados en el campo de búsqueda totalizado los cuales su temática a resolver es una problemática relacionada con el rol humano.

A finales de 2019 se realizan dos publicaciones relacionadas a la problemática mencionada anteriormente. En primer lugar, los autores Abdullah, Abdul Rahman & Salleh publican y exponen un modelo que aborda la necesidad de cuantificar el rol humano en función del trabajo que realizan. Se obtiene un modelo que busca optimizar el rendimiento al minimizar el makespan. El modelo planteado permitirá el análisis de datos a realizar sobre estructuras de actividad actuales o reconfiguraciones (IBID).

En segundo lugar, se publica un artículo por parte de Chu, Gao, Cheng, Wu, Chen, Shi & Qin (2019) en el cual se presenta un modelo que se centra el problema de asignación de trabajadores a múltiples celdas con entrenamiento cruzado y efectos de aprendizaje y olvido. Como

resultante se tiene un modelo no lineal entero mixto que busca minimizar los costos de formación y el desequilibrio de la carga de trabajo. Para su solución se implementa un método dinámico basado en la mutación de Cauchy.

### **3.2.5 Problema múltiple**

En esta sección se presentan todos los artículos que se centran en solucionar múltiples problemáticas en un CMS. A inicios de 2019 se realizan dos publicaciones, en primer lugar, los autores Feng, Xia, Da, Xi & Pan (2019) publican y exponen un modelo centrado en problemáticas como: determinación de número de celdas, formación de grupos de máquinas, selección de máquinas adecuadas para cada operación y generación de un cronograma de producción óptimo. Además, el modelo considera muchos atributos como máquinas duplicadas, rutas de proceso alternativas, partes reentrantes y número de celdas variables. Como resultante se obtiene un modelo no lineal de entero mixto que busca minimizar el makespan y para su solución se implementa un algoritmo genético mejorado (IGA). En segundo lugar, se publica un artículo por parte de Venkata Deepthi, Ramakotiah & Krishnaveni (2019), los cuales presentan un modelo centrado en problemas de programación y formación de celdas que evalúa los efectos de la producción en el ámbito económico y ecológico. Se obtiene un modelo no lineal entero mixto que busca minimizar el costo de capacitación de los trabajadores en función de disminuir el tiempo de procesamiento, también disminuye el makespan lo que a su vez disminuye la energía utilizada mejorando la huella de carbono. Debido a su complejidad se utiliza un algoritmo de llama de polilla (MFO) y una heurística para su solución.

A finales de 2019 se publica dos artículos, primero por parte de Ghassemi Tari & Ahadi (2019) exponen un modelo de programación y formación de celdas con asignación de estaciones de trabajo a las zonas cerradas no superpuestas. Además, centrado en la distribución de instalaciones con vehículos guiados automatizados (AGV) como dispositivo de transporte. Como resultante se tiene un modelo que busca minimizar los costos totales de manipulación de materiales. Como segundo los autores Aalaei, Kayvanfar & Davoudpour (2019), entregan un modelo centrado en las problemáticas de formación de celdas cúbicas para múltiples mercados y planificación de la producción para múltiples plantas. Cabe aclarar que el modelo también presenta interés sobre la determinación de producción en cada planta. Se obtiene un modelo que busca minimizar el costo total compuesto de costos de mantenimiento, subcontratación, manejo de

materiales, transporte, costos fijos de producción y salarios. Para su solución se implementa el método Taguchi y un algoritmo genético (GA).

A principios de 2020, se publica un artículo por parte de Xue & Offodile (2020) quienes exponen un novedoso modelo centrado en las problemáticas de formación de celdas dinámicas y planificación de la producción con reconfiguración de celdas, cantidades variables de demanda y máquinas de no sufren averías. Como resultado se tiene un modelo no lineal entero mixto que busca minimizar los costos relacionados con la máquina y el producto, tales como: mantenimiento, instalación, reubicación, manipulación del material entre celdas e intraceldas, mantenimiento del inventario, subcontratación y pedidos pendientes. Para su solución se implementa el método branch-and-bound.

A mediados de 2020 se realizan cuatro publicaciones, en primer lugar, los autores Rahimi, Arkat & Farughi (2020) presentan un modelo que busca solución a los problemas de programación y formación de celdas y se centra en atributos como: agrupamiento de máquinas, selección de rutas de procesamiento, secuenciación de operaciones y ubicación de celdas. Como resultante se tiene un modelo de enteros mixtos que busca minimizar el makespan y es solucionado con un algoritmo de optimización de la amortiguación de vibraciones (VDO). En segundo lugar, está la publicación de Wang, Liu & Zhou (2020), en la cual entregan un modelo centrado en máquinas multifuncionales con secuencias de operaciones cambiantes y trabajadores polivalentes que pueden aprender y olvidar. Se obtiene entonces un modelo no lineal entero mixto que busca minimizar el makespan y se soluciona con un algoritmo mejorado de búsqueda de bacterias (IBFA). En tercer lugar, se presenta el artículo de Alimian, Ghezavati & Tavakkoli-Moghaddam (2020), ellos exponen un modelo centrado en la implementación de la planificación de mantenimiento preventivo, la creación de familias y grupos de máquinas. Como resultado se tiene un modelo no lineal entero mixto que busca minimizar los costos totales y se soluciona implementado el paquete de descomposición de Bender en GAMS para evaluar las interacciones. En cuarto lugar, el artículo publicado por Mehdizadeh, Shamradifar & Akhavan Niaki (2020) en el que presentan un modelo centrado en el problema de programación y formación de celdas dinámicas con recursos limitados. Como resultante se obtiene un modelo lineal que busca minimizar los costos totales asociados a la planificación de la producción, construcción y formación de celdas. Para su solución se utiliza el método Taguchi para calibrar parámetros y

encontrar soluciones de mejor calidad. Además, utiliza un algoritmo genético (GA) y un algoritmo de optimización por enjambre de partículas (PSO).

A mediados de 2021, se publica un artículo por parte de Behnia, Shirazi, Mahdavi & Paydar (2021) en el cual exponen un modelo que posee una función líder que busca solucionar la formación de celdas, otorgar una decisión estratégica y lograr la innovación del producto. También, tiene una función seguidora que busca solucionar la programación de celdas donde se tiene en cuenta la asignación de recursos humanos. Se obtiene un modelo que busca minimizar los vacíos y elementos excepcionales en el CMS, además se desea optimizar la asignación de recursos humanos. Para su solución se implementan dos metaheurísticas mejoradas, en primer lugar, un algoritmo genético anidado en dos niveles (NBL-GA) y en segundo lugar un algoritmo de optimización de enjambre de partículas anidadas de dos niveles (NBL-PSO).

### **3.3 Conclusiones de la revisión de literatura**

En este numeral se presentan las conclusiones a las que se llega luego de realizar la revisión literaria de los documentos encontrados. En primer lugar, la *Figura 5* muestra los tipos de problema encontrados con mayor frecuencia en las publicaciones. Se concluye que el 22,85% de los artículos se centraron en dos problemáticas como lo son la programación y formación de celdas. Mientras que el 20% de las publicaciones solo se centraron en dar solución a la problemática de formación de celdas.

#### **Figura 5.**

*Problemas con más frecuencia en las publicaciones*

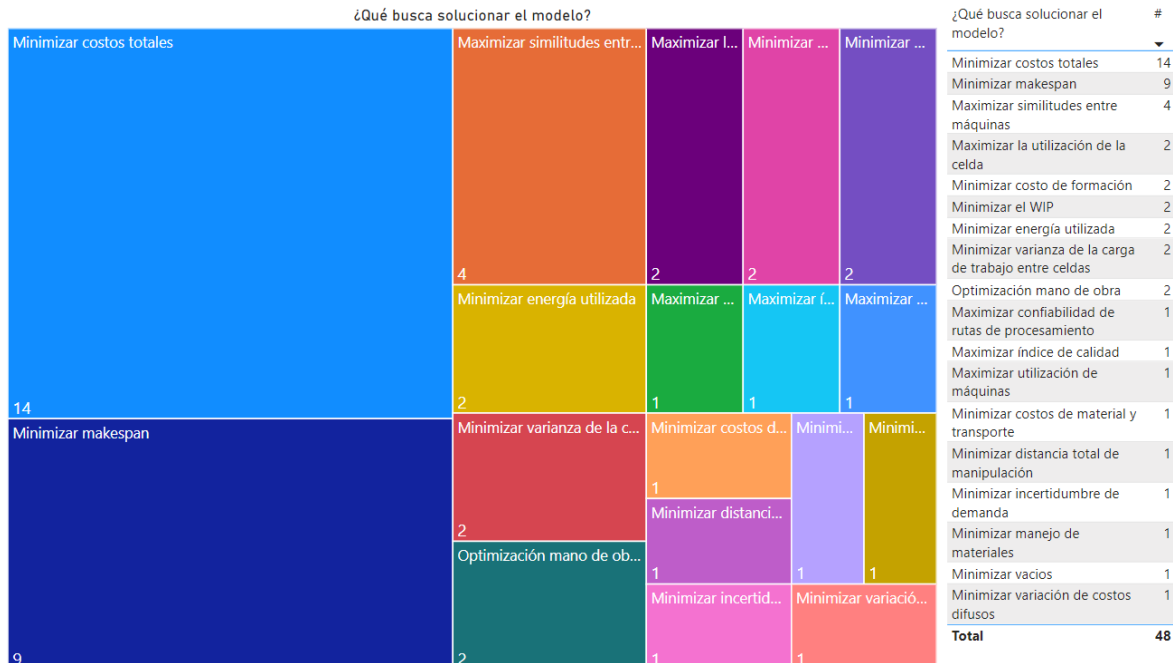
PROBLEMA DE CELDAS DE MANUFACTURA	Recuento de PROBLEMA DE CELDAS DE MANUFACTURA
Problema de programación y formación de celdas	8
Problema de formación de celdas	7
Problema de formación de celdas dinámicas	5
Problema de programación de productos	3
Problema de formación de celdas cúbicas	2
Problema de aplicación restringida y fragmentada del rol humano.	1
Problema de asignación de trabajadores	1
Problema de formación de celdas cúbicas generalizado	1
Problema de formación de celdas cúbicas para múltiples mercados y planificación de la producción para múltiples plantas	1
Problema de formación de celdas dinámicas y la planificación de la producción.	1
Problema de formación de celdas dinámicas.	1
Problema de programación de celdas virtuales	1
Problema de programación de producción en celdas dinámicas	1
Problema de programación de productos en celdas dinámicas.	1
Problema de programación de productos en celdas múltiples	1
<b>Total</b>	<b>35</b>

En segundo lugar, en la se puede evidenciar los tres criterios de optimización utilizados con mayor frecuencia en los modelos matemáticos, estos son: minimizar costos totales que se presenta en un 29,17% de los criterios de optimización encontrados, minimizar makespan que está presente en un 18,75% y maximizar similitudes entre máquinas que corresponde al 8,33%. También es de resaltar que estos porcentajes se calculan sobre un total mayor al número de documentos resultantes, esto se debe algunas publicaciones se centran en más de un criterio de optimización.

En la *Figura 6* se puede evidenciar los tres criterios de optimización utilizados con mayor frecuencia en los modelos matemáticos, estos son: minimizar costos totales que se presenta en un 29,17% de los criterios de optimización encontrados, minimizar makespan que está presente en un 18,75% y maximizar similitudes entre máquinas que corresponde al 8,33%. También es de resaltar que estos porcentajes se calculan sobre un total mayor al número de documentos resultantes, esto se debe algunas publicaciones se centran en más de un criterio de optimización.

### **Figura 6.**

*¿Qué busca solucionar el problema?*



Y finalmente, en la *Figura 7* se presentan los métodos de solución más utilizados en las investigaciones consultadas. Como resultado se tiene que el algoritmo genético (GA) está presente en un 12% de los metodos de solución encontrados, el algoritmo de recocido simulado (SA) está presente en un 8% y el algoritmo de optimización de enjambre de partículas (PSO) se encuentra en el 6%. Es necesario destacar que estos porcentajes al igual que los expuestos anteriormente se calculan sobre un total mayor al número de documentos resultantes y se debe a que algunos artículos utilizaban más de un método de solución.

**Figura 7.**  
*Métodos de solución más utilizados*

Métodos de Solución								MÉTODO DE SOLUCIÓN	#
6	Algoritmo Genético (GA)	Taguchi	Algori...	Algori...	Algori...	Algori...	Algori...	Algoritmo Genético (GA)	6
		2	1	1	1	1	1	Algoritmo de recocido simulado (SA)	4
		1						Algoritmo Optimización de Enjambre de Partículas (PSO)	3
4	Algoritmo de recocido simulad...	Algoritmo de opt...	Algoritmo Opti...	Algoritm...	Algoritm...	Algoritm...		Algoritmo heurístico	2
		1	1	1	1	1		Algoritmo de Búsqueda de vecindad de variable general (VNS)	2
		1	1	1	1	1		Algoritmo de bifurcación y enlace	1
		1	1	1	1	1		Algoritmo de descomposición de Benders	1
3	Algoritmo Optimización de Enj...	Algoritmo genéti...	Algoritmo opti...	Algoritm...	Algoritm...	Algoritm...		Algoritmo de distancia de la máquina dependiente del diseño	1
		1	1	1	1	1		Algoritmo de Estimación de Distribución	1
		1	1	1	1	1		Algoritmo de evolución multiobjetivo mejorado basado en la descomposición con estrategia de búsqueda local (IMOEa /D-SL)	1
2	Algoritmo heurístico	Algoritmo genéti...	Algoritmo Reco...	Método branch...	Méto...	Resuel...		Algoritmo de Keshtel (KA)	1
		1	1	1	1	1		Algoritmo de optimización de enjambre de partículas híbrido con Nawaz, Enscore, Ham-NEH (NEPSO)	1
								Algoritmo de optimización de la amortiguación de vibraciones (VDO)	1
								Algoritmo de polinización de flores discretas (DFPA)	1
								Algoritmo genético anidado en dos niveles(NBL-GA)	1
								Algoritmo genético basado en recocido simulado (SAGA)	1
								Algoritmo genético de calificación no dominada(NSGA)	1
								Algoritmo genético de clasificación no dominado modificado (MNSGA-II)	1
								Algoritmo genético híbrido	1
								Algoritmo genético(GA) de clasificación no dominante-II	1
								Algoritmo híbrido de BBO y GA	1
								Algoritmo mejorado de búsqueda de bacterias (IBFA)	1
								Algoritmo Optimización de Enjambre de Partículas anidadas de dos niveles (NBL-PSO)	1
								Algoritmo Optimización de Enjambre de Partículas Modificadas (MPSO)	1
								Algoritmo optimizador de lobo gris (GWO)	1
								<b>Total</b>	<b>50</b>

#### 4. Marco Teórico

El siguiente contenido contempla aquellos conceptos que se consideran son necesarios para el desarrollo del presente trabajo de investigación.

##### 4.1 Diseño de sistemas de producción

El diseño de sistemas de producción hace referencia al diseño de planta, es decir la forma en la que se dispondrá de los centros de trabajo, en el libro “Sistemas de producción automatizados” escrito por Barrientos & Gambao (2014) se exponen 4 tipos de diseño de sistemas que envuelve a nivel general las disposiciones que se pueden presentar, estas son:

- Pieza en posición fija.
- Agrupamiento por procesos.
- Agrupamiento por células.
- Agrupamiento en línea de producción.

##### 4.2 Metaheurísticas

A continuación, se enuncian los algoritmos que se trabajarán para la búsqueda de soluciones

###### 4.2.1 Algoritmo genético

El algoritmo genético fue discutido por primera vez por Holland en 1975, podría describirse esta metaheurística como la supervivencia del más apto. Este algoritmo es usado para

la optimización de problemas que manejan combinaciones que necesitan explorar el mayor número de soluciones posibles en un tiempo razonable (Singamsetty & Thenepalle, 2021).

El algoritmo se comporta de tal forma en que se selecciona de manera al azar soluciones que se denominan población o cromosomas dentro de los que se van almacenando datos, cada cromosoma es una solución y se le asigna un valor de aptitud (fitness) según su rendimiento con respecto a el objetivo fijado. De la población seleccionada se van seleccionando pares de cromosomas llamados padres, la forma de selección es según su valor de aptitud, estos dos cromosomas se cruzan dando origen a una nueva generación de cromosomas y estos dos suplantando a los anteriores (padres) siempre y cuando el valor de aptitud sea superior al de predecesor, seguido se mutan los cromosomas con el fin de mantener diversidad en la búsqueda, este proceso se continua ejecutando hasta reducir las posibilidades de cruce entre cromosomas, es entonces donde se busca la nueva población para empezar a iterar de manera constante hasta llegar al criterio de finalización de iteración (Singamsetty & Thenepalle, 2021).

#### 4.2.1.1 Función de aptitud o función fitness

Es uno de los pasos que dan sucesión a la siguiente generación, o a futuros cruces de cromosomas, cuanto más alto es el valor de aptitud se considera como un candidato óptimo para la generación siguiente (Singamsetty & Thenepalle, 2021). Usualmente se determina una ecuación que represente el objetivo buscado, en este caso la función fitness corresponde a la función objetivo, en el que, a menor makespan encontrado, mayor será su valor de aptitud.

#### 4.2.1.2 Operador de selección

La selección de cromosomas define el proceso que se lleva a cabo y la forma en la que se desempeñará su solución, en la *Tabla 7*, se presentan algunas opciones de operadores de selección más usadas:

**Tabla 7.**  
*Métodos de selección*

Método	Descripción
<i>Ruleta</i>	Es un operador que se comporta en forma de proporción, teniendo en consideración el valor de aptitud obtenido en la ecuación fitness, cuanto más alto es dicho valor, mayor será la proporción que se le asigna dentro del grupo seleccionado para el apareamiento. Este método no garantiza la

Continuación Tabla 7

<b>Método</b>	<b>Descripción</b>
	selección de los individuos aptos, pero si presentan una mayor oportunidad de ser seleccionados (Sivanandam & Deepa, 2008).
<b>Aleatorio</b>	Se elige un padre al azar, sin tener en cuenta su valor de aptitud, lo que hace que el método explore o considere candidatos poco aptos (IBID).
<b>Rango</b>	El rango se usa siempre y cuando los valores de aptitud difieran mucho, pues si esto ocurre usar la ruleta no es buena opción. Dado el caso el rango se encarga de clasificar la población y cada cromosoma recibe aptitud de dicha clasificación, aquel con peor aptitud tiene valor de 1 y el mayor tiene valor de N, resulta lento, pero evita la convergencia excesivamente rápida (IBID).
<b>Torneo</b>	Es una estrategia en la que se establece un torneo en el que los individuos compiten, aquel con mejor condición física es considerado el mejor (IBID).
<b>Boltzmann</b>	Simula el proceso de enfriamiento de metales que genera soluciones pequeñas en torno al objetivo trabajado, la temperatura inicia elevada indicando que la presión de selección es baja, a medida que la temperatura baja, la presión aumenta llevando al espacio de búsqueda con mejores candidatos (IBID).
<b>Estocástico</b>	Se asignan puntos en una línea, la distancia entre los puntos depende de su valor de aptitud (Similar a la ruleta), su manera de selección es similar a la rueda, con la diferencia que en este caso se tiene un sesgo cero y una dispersión mínima (IBID).

*Nota: información adaptada de Sivanandam & Deepa. 2008. "Introduction to Genetic Algorithms"*

#### 4.2.1.3 Operador de cruce

El operador de cruce es un aspecto fundamental dentro del proceso pues ayuda a diversificar la población estudiada y generando nuevos candidatos a partir de los más aptos, pues los cromosomas generados son creados a partir del par seleccionado con mayor valor de aptitud. Existen distintos operadores de cruce como los mostrados en la *Tabla 8*:

**Tabla 8.***Tipos de operador de cruce*

<b>Método</b>	<b>Descripción</b>
<b><i>Cruce de punto único</i></b>	Se seleccionan dos cromosomas los cuales se aparean y realizan un cruce en un punto aleatorio a lo largo de la cadena de cromosomas. El fin de este tipo de cruce es obtener los mejores hijos a partir de buenos padres (Sivanandam & Deepa, 2008).
<b><i>Cruce de dos puntos</i></b>	Funciona de la misma manera que el cruce de un punto, sin embargo, agregar un punto genera que se pueda interrumpir los componentes básicos del GA, pero, explora el espacio de búsqueda mucho más a fondo (IBID).
<b><i>Cruce de puntos múltiples (N puntos)</i></b>	Se puede examinar las soluciones de dos formas, en las que el número de puntos es par o puede ser impar, cuando el número de cruces es impar se asume un punto cruzado en la cadena al comienzo (IBID).
<b><i>Cruce uniforme</i></b>	Difiere del cruce de N puntos en el que, ya que se genera una máscara de cruce binaria en la que de ser asignado el 1 a la máscara se copiará el gen del primer padre, mientras que si es 0 se asignará una copia del segundo padre, las máscaras se generan aleatoriamente para cada par de padres (IBID).
<b><i>Cruce de tres padres</i></b>	Se seleccionan tres padres de manera aleatoria. Se comparan los bits (1 o 0) de los dos primeros padres y en caso de ser iguales el bit se toma para la descendencia, de no ser así se toma el bit del tercer padre para la descendencia (IBID).
<b><i>Cruce con sustituto reducido</i></b>	Busca producir siempre nuevos individuos siempre y cuando sea posible, para ello se restringe la ubicación del cruce.
<b><i>Cruce aleatorio</i></b>	Se relaciona con el cruce uniforme. Se determinan dos padres aleatoriamente, luego de la recombinación las variables en la cría se mantienen en orden, eliminando el sesgo (IBID).

Continuación Tabla 8

Método	Descripción
<b><i>PPX “Cruce conservativo de precedencia”</i></b>	Consiste en codificar los pares de padres, seguido se permutan las respuestas dependiendo de un conjunto que contiene los números 1 y 2, cuando se selecciona un padre, este es desechado y se pasa al segundo padre, continua el proceso eliminando al padre 2 y examinando cada par de cromosomas presentes en la región de búsqueda (IBID).
<b><i>Cruce ordenado</i></b>	Se cruzan dos puntos de manera ordenada en el que cada gen tiene tres partes, izquierda, medio y derecha, la descendencia del hijo 1 adquiere la parte derecha e izquierda y su parte central la determinan los genes del padre 1, en el orden que los valores se presentan en el padre 2, lo restante se aplica al hijo 2 mediante un proceso similar.
<b><i>PMX “Cruce parcial emparejado”</i></b>	Se eligen aleatoriamente dos puntos de cruce, mediante una codificación de permutación garantiza que todas las posiciones serán revisadas una vez en cada descendencia (IBID).
<b><i>Probabilidad de cruce</i></b>	La probabilidad de cruce es necesaria como un parámetro para definir la frecuencia de cruce, si no se presenta cruce los hijos con copias de sus padres. Esto busca que la nueva generación salga de los mejores cromosomas, pero es bueno optar por un par viejo dentro de la nueva generación (IBID).

*Nota: información adaptada de Sivanandam & Deepa. 2008. “Introduction to Genetic Algorithms”*

#### 4.2.1.4 Operador de mutación

El operador de cruce es indispensable si se desea explorar más opciones de posibles soluciones al problema, puesto que la función del operador de cruce es evitar que el algoritmo se estanque en un óptimo local (Singamsetty & Thenepalle, 2021).

En la *Tabla 9* se presentan algunos de los más usados:

**Tabla 9.**  
*Operadores de mutación*

Método	Descripción
<i>Voltear</i>	Se voltea un bit, lo que implica que, si se encontraba en 0, cambia a 1 o viceversa, el gen mutado es el principal y a raíz de este se genera el hijo (Sivanandam & Deepa, 2008).
<i>Intercambio</i>	Se eligen dos posiciones aleatorias en la cadena de solución, seguido, estos bits cambian de puesto y genera una nueva cadena (IBID).
<i>Inversión</i>	Se elige una posición aleatoria y los bits juntos a esta se cambian, generando una descendencia nueva (IBID).
<i>Probabilidad de mutación</i>	La probabilidad de mutación define con qué frecuencia se mutan los pares de cromosomas, si no se presenta mutación, entonces la descendencia se genera después del cruce. No se recomienda mutar muy seguido pues entonces el GA cambia a una búsqueda aleatoria (IBID).

*Nota: información adaptada de Sivanandam & Deepa. 2008. "Introduction to Genetic Algorithms"*

#### 4.2.2 Optimización por Enjambre de Partículas (Particle Swarm Optimization-PSO)

Fue propuesto por J. Kennedy y R. Eberhart en 1995. Este tipo de optimización explora el espacio de búsqueda mediante partículas, las cuales llevan una trayectoria y se distribuyen hasta llegar a una posición determinada por la trayectoria. Para modificar la posición de la partícula el algoritmo define un vector de velocidad, en cada posición las partículas emiten un error con el fin que el vector de velocidad cambie y generando una posición diferente, de esta forma cada partícula va encontrando una ruta de soluciones las cuales serán analizadas y se seleccionará la mejor entre ellas (Duarte, Pantrigo, & Gallego, 2007).

Para este algoritmo se describen dos ecuaciones expresadas en las *Figura 8* y *Figura 9* que indican cómo se comporta el algoritmo:

#### **Figura 8.**

*Ecuación de velocidad PSO*

$$v_i(t+1) = wv_i(t) + c_1r_1(p_i(t) - x_i(t)) + c_2r_2(p_g(t) - x_i(t)),$$

*Nota: Tomado de Yu, Wang, & Xi (2008), "Evolving artificial neural networks using an improved PSO and DPSO"*

**Figura 9.**

*Ecuación de actualización de Posición*

$$x_i(t+1) = x_i(t) + v_i(t+1).$$

Nota: Tomado de Yu, Wang, & Xi (2008), "Evolving artificial neural networks using an improved PSO and DPSO"

En la *Figura 8* el primer término es designado al componente de velocidad, es aquel que determina la velocidad con la cual la partícula se desplazará y aprenderá a lo largo del tiempo. El segundo término guarda y expresa la posición de cada partícula, en caso de encontrar soluciones mejores a las que exploraba y, por último, el tercer componente representa la mejor solución encontrada hasta ese fragmento de tiempo.

### **4.2.3 Optimización combinatoria**

Rama de la matemática aplicada, donde computacionalmente se buscan soluciones algorítmicas a criterios de optimización. Su valor está en la palabra combinatoria haciendo referencia a que únicamente existe un número finito de soluciones factibles en un espacio de soluciones de gran talla. Una aplicación acertada de la optimización combinatoria se centra en reducir el tamaño del espacio a buscar y explora de modo eficiente dicho espacio (José, 2019).

### **4.3 Sistemas de fabricación celular**

Conocidos como celdas o células de manufactura. Son grupos de máquinas o estaciones de trabajo, esto conforma centros de trabajos los cuales son planeados en una distribución de planta con el fin de optimizar la producción y flexibilizar el sistema, el conjunto de celdas o células dan sentido a la planta y a su objetivo productivo (Barrientos & Gambao, 2014).

### **4.4 Tecnología de grupos**

La tecnología de grupos es una configuración que va muy de la mano con lo que corresponde a diseño de sistemas de producción, podría considerarse como un paso paralelo en el diseño de plantas, el concepto de tecnología de grupos consiste, así como su nombre lo indica, en agrupar un taller dependiendo del proceso o máquinas con el fin que se encuentren lo más cercanas posible, generando así mayor flexibilidad en el sistema y volviéndose mucho más sólido, además de volver todo el proceso fácil de supervisar y difícil de interrumpir (Xiong, Lu, & Lu, 2021).

### **4.5 Sistemas de manufactura**

El concepto de sistema de manufactura es bastante amplio puesto que considera todo aquello referente a la fabricación de productos, sin embargo, es aplicado de manera puntual en

cada caso dentro de las diferentes entidades, un ejemplo claro de esto es el trabajo expuesto por Guo, y otros (2021) en el que se pretende optimizar un sistema de producción de artículos de gran tamaño y frágiles además de considerar tiempos de entrega en este caso se considera como sistema de manufactura a la producción de los productos tratados, pero, si se observa en otra investigación, como ejemplo, se puede encontrar el caso trabajado por Ma, Lee, Shi, & Li (2020) en el que se trata un proceso que busca la asignación de operarios a diferentes tareas para cumplir con el sistema de fabricación. Como se observa, un sistema de manufactura es todo aquello que envuelve un proceso de fabricación en una empresa de manera óptima y que se ajusta a la necesidad tratada por la empresa que lo lleva a cabo.

#### **4.6 Asignación de operaciones y asignación de trabajadores**

La asignación de operaciones es la forma en la que se definen tareas a la mano de obra o máquinas de un proceso productivo, para esto es necesario tener en consideración los recursos con los que se cuentan en un determinado periodo de tiempo para un horizonte de producción de finido, esto puede verse en el caso de estudio que enuncia “...Dado que el número de operadores (ya sea seis, siete u ocho) es mucho menor que el número de estaciones, un operador suele ser responsable de varias máquinas...” (IBID) es de esta forma en la que teniendo en cuenta los recursos tanto de máquinas disponibles como trabajadores presentes se pueden asignar máquinas a trabajadores y tareas a máquinas.

#### **4.7 Planeación de la producción**

La planeación de la producción se define como el proceso estructurado más importante en el área de producción destinada para el logro de metas y objetivos definidos a nivel empresarial dentro de un periodo de tiempo conocido, estos optimizan los procesos basándose en el principio de eficacia, eficiencia, racionalidad y trascendencia, provocando el cumplimiento de metas al menor costo (Zuluaga, 2014)

#### **4.8 Makespan**

Según autores como Salinas Hilburg , Zapater, Moya, & Ayala (2021) el makespan se define como el tiempo total de ejecución de un lote que será llevado a cabo en un centro de datos bajo el escenario que se plantea en el trabajo, expresado de otra forma, el makespan hace referencia al tiempo total desde que inicia un proceso hasta que termina, pasando por todas y cada una de las etapas que comprende el sistema que se lleva a cabo en la organización.

#### 4.9 Complejidad computacional

Los modelos matemáticos según su complejidad son clasificados en ciertos grupos dependiendo de la cantidad de recursos usados al momento de implementarse en una máquina computacional. La *Tabla 10* muestra el resumen de lo mencionado:

**Tabla 10.**  
*Complejidad del problema*

Clase	Descripción
<i>P</i>	Son problemas que pueden ser resueltos de manera exacta y se emplea un tiempo polinomial en su solución.
<i>NP</i>	Son problemas de gran complejidad en los cuales se involucra una gran cantidad de variables.
<i>NP-Complete</i>	No puede ser resuelto en tiempo polinomial y a su vez no existe una solución exacta para el problema, en lugar de esto se acerca a una solución.
<i>NP-Hard</i>	No se puede definir su complejidad mediante la maquinas por esto resulta mas complejo encontrar una solución, llevando a explorar espacios de soluciones con el fin de proporcionar una respuesta.

*Nota: información adaptada de Cruz Chávez, Moreno Bernal, & Perlata Abarca. 2014. "Aplicación de la teoría de la complejidad en optimización combinatoria"*

### 5. Modelo matemático

#### 5.1 Descripción del modelo

El presente trabajo se centra en el modelo matemático de programación lineal entera, propuesto por Jufeng Wang, Chunfeng Liu & Kai Li (2019), el cual busca minimizar el tiempo máximo de finalización de todas las tareas (makespan) con las que cuenta un sistema de manufactura celular con recursos limitados (máquinas y trabajadores). Para el desarrollo del modelo los autores establecen cuatro supuestos:

- Trabajador y máquina: el número de máquinas y trabajadores se conocen. El número de máquinas es mayor que el número de trabajadores.
- Tareas: Para cada tarea, al menos una máquina tiene capacidad de procesarla. Para cada máquina, al menos un trabajador tiene la capacidad de procesarla, las tareas no se pueden interrumpir lo que implica que cada tarea debe ser procesada por una

máquina y un solo trabajador. El tiempo de procesamiento de tareas depende de la tarea asignada y del trabajador. Existe relación de precedencia entre las tareas.

- Tamaño de celda: Cada celda tiene un tamaño máximo especificado y no se permite redundancia de máquinas en las celdas.
- Movimiento de trabajadores: los trabajadores pueden moverse entre diferentes maquinas, haciendo el movimiento intracelular e intercelular. El tiempo de movimiento se conoce de antemano.

Los supuestos son tomados y traducidos de “A hybrid simulated annealing for scheduling in dual-resource celular manufacturing system considering worker movement”. Las notaciones usadas en el modelo estudiado son presentadas a continuación.

## 5.2 Notación

### 5.2.1 Índices

$w$  = Índice para el trabajador.

$c$  = Índice para la celda.

$j$  = Índice para la tarea.

$t$  = Índice para el tiempo.

$r$  = Índice para intervalo de tiempo continuo.

### 5.2.2 Parámetros de entrada

$W$  = Número de trabajadores.

$J$  = Número de tareas.

$M$  = Número de máquinas.

$N$  = Mayor número de tareas que un trabajador puede procesar.

$B_u$  = Límite superior de tamaño de celda(medido en número de máquinas).

$C$  = Número de celdas, es el entero más pequeño no menor a  $\frac{M}{B_u}$ .

$Q_{jm} = 1$  si la tarea  $j$  se puede procesar en la máquina  $m$ , y 0 en caso contrario ( $j = 1, \dots, J$ ;  $m = 1, \dots, M$ ).

$M_j$  = Conjunto de máquinas que tienen la capacidad de procesar la tarea  $j$  ( $j = 1, \dots, J$ ).

$P_{jmw}$  = Tiempo de procesamiento de la tarea  $j$  en la máquina  $m$  por trabajador  $w$  ( $j = 1, \dots, J$ ;  $m \in \mathcal{M}_j$ ;  $w = 1, \dots, W$ ).

$Y_{mcm'c'}$  = Tiempo de movimiento del trabajador hacia la máquina  $m'$  en la celda  $c'$  desde la

máquina  $m$  en la celda  $c$  ( $m, m' = 1, \dots, M; c, c' = 1, \dots, C$ ).

$P_j$  = Conjunto de tareas predecesoras de la tarea  $j$  ( $j = 1, \dots, J$ ).

$L$  = Número suficientemente grande.

### 5.2.3 Variables de decisión

$Z_{mc}$  = 1 si la máquina  $m$  está ubicada en la celda  $c$ , y 0 en caso contrario ( $m = 1, \dots, M; c = 1, \dots, C$ ).

$C_m$  = Celda en la que se encuentra la máquina  $m$  ( $C_m = c$  si  $Z_{mc} = 1$ ).

$X_{jmwr}$  = 1 si la tarea  $j$  es procesada en la máquina  $m$  por el trabajador  $w$ , y la tarea  $j$  es la  $r$  –ésima procesada por el trabajador  $w$  (es decir, es procesada en la  $r$  –ésimo intervalo de tiempo continuo), y 0 en caso contrario ( $j = 1, \dots, J; m = 1, \dots, M; w = 1, \dots, W; r = 1, \dots, N$ ).

$FT_j$  = Tiempo de finalización de la tarea  $j$  ( $j = 1, \dots, J$ ).

### 5.2.4 Modelo matemático

$$\text{Min } C_{max} = \max\{FT_j | j = 1, \dots, J\} \quad (1)$$

Sujeto a (S.t)

$$\sum_{c=1}^C Z_{mc} = 1, \forall m \quad (2)$$

$$\sum_{m=1}^M Z_{mc} \leq 1, \forall c \quad (3)$$

$$X_{jmwr} \leq Q_{jm}, \forall j, m, w, r \quad (4)$$

$$\sum_{m=1}^M \sum_{w=1}^W \sum_{r=1}^N X_{jmwr} = 1, \forall j \quad (5)$$

$$\sum_{j=1}^N \sum_{m=1}^M X_{jmwr} \leq 1, \forall w, r \quad (6)$$

$$\sum_{i=1}^N \sum_{m=1}^M X_{imwr} - \sum_{j=1}^N \sum_{m=1}^M X_{j,m,w,r-1} \leq 0, \forall w, r = 2, \dots, N \quad (7)$$

$$FT_j - FT_i + L(2 - X_{jm'wr} - X_{i,m,w,r-1}) \geq P_{jmw} + Y_{m c_m m' c_m'}, \forall i \neq j, \forall m, w, \\ r = 2, \dots, N \quad (8)$$

$$FT_j - FT_i \geq \sum_{m=1}^M \sum_{w=1}^W \sum_{r=1}^R P_{jmw} X_{jmwr}, \forall i \in P_j, \forall j \quad (9)$$

$$FT_j \geq 0, Z_{mc}, X_{jmwr} \in \{0,1\}, \forall j, m, w, r \quad (10)$$

El modelo lineal propuesto por Jufeng Wang, Chunfeng Liu & Kai Li (2019), se enfoca en la minimización del máximo tiempo de procesamiento, es decir busca al máximo la reducción del tiempo de procesamiento, la función objetivo expresada en la *Ecuación 1*, se enfoca en ese único termino, denominado makespan. Sin embargo, para tal cometido se tiene en cuenta ciertas condiciones que deben cumplirse para encontrar la mejor respuesta.

En primer lugar, encontramos las *Ecuación 2* y *Ecuación 3* destinadas a la formación de celdas, así pues, la ecuación 2 se centra en que cada maquina debe asignarse únicamente en una celda, mientras que la ecuación 3 indica que el número de máquinas presentes en cada celda no puede superar el tamaño máximo destinada para esta. Luego tenemos el conjunto de ecuaciones comprendidas desde la 4 hasta la 9, estas se enfocan en la asignación de tareas, empezando con la *Ecuación 4* que indica que solo se puede asignar tareas en aquellas máquinas que se encuentran habilitadas para procesar dicha tarea, después tenemos la *Ecuación 5* encargada de hacer que cada

tarea  $j$  para cada máquina  $m$  de cada trabajador  $w$  sea procesada en el tiempo de ese trabajador, luego, se encuentra la *Ecuación 6* destinada a que cada trabajador opere máximo una máquina y realice como mucho una tarea dentro de su intervalo de tiempo, ahora, se tiene la *Ecuación 7* que es básicamente una ecuación de precedencia que indica que si un trabajador procesa una tarea en un tiempo continuo, el mismo trabajador debió procesar otra tarea en el tiempo continuo anterior. La *Ecuación 8* muestra la relación existente entre el tiempo de finalización de dos tareas que se procesaron en intervalos de tiempo consecutivos por un trabajador, visto de otra forma, que la diferencia del tiempo de finalización de dos tareas debe ser mínimamente el tiempo consumido para procesarlas junto con el desplazamiento de los trabajadores, mientras, la *Ecuación 9* expresa la relación que existe entre dos tareas consecutivas. Por último, se tiene la *Ecuación 10*, en la que la variable  $FT_j$  se trata de una variable positiva, mientras que las variables  $Z_{mc}$  y  $X_{jmwr}$  son variables binarias.

## 6. Calibración de parámetros

Para el problema PFCMT expuesto por los autores Jufeng Wang, Chunfeng Liu, & Kai Li (2019) se encuentran ya expresados los tamaños que adquieren los parámetros de entrada, en la *Tabla 11* se muestran cuáles son los intervalos de cada parámetro proporcionados por los autores, en dicha investigación se fijan y varían parámetros con el fin de obtener combinaciones y determinar cómo se ve afectado el modelo ante el número de variables.

**Tabla 11.**  
*Parámetros del modelo*

Parámetro	Intervalo
Tareas (J)	DU (150,300)
Máquinas (M)	DU (30,60)
Trabajadores (W)	DU (30,60)
Tamaño máximo de máquinas por celda (Bu)	DU (4,10)
Tiempos de procesamiento ( $P_{jmw}$ )	DU (5,40)
Número de máquinas que pueden procesar la operación ( $N_m = M_j$ )	DU (1,10)

Debido a que las pruebas realizadas por los autores se realizan en cuatro instancias, se realiza un análisis para determinar el crecimiento de cada intervalo, encontrándose así que, para

las tareas, existe un aumento de 50 tareas por prueba, lo que representa un crecimiento del 33.33% con relación a cada prueba, tomando como base un mínimo de tareas como 150, la *Ecuación 11* representa el cálculo obtenido:

$$\frac{\text{Incremento}}{\text{Tareas mínimas}} \times 100 = \frac{50}{150} \times 100 = 33.33\% \quad (11)$$

Este análisis se hace con la finalidad de disminuir el número de variables totales, bajo la consideración que se desea tener un tiempo computacional razonable ya que al intentar emular una de las pruebas mínimas establecidas por los autores del modelo se encontraron datos poco viables especialmente en tiempo de corrida, esta calibración surge luego que al correr una prueba de 100 individuos con parámetros de 150 tareas, 30 máquinas, 25 trabajadores y 6 tiempos obteniendo así un tiempo de corrida de 112009.464955 segundos equivalente a 31.11374027 horas (Apéndice B). De este modo se procede a calcular cada uno de los parámetros para determinar el incremento porcentual en cada uno de estos, los datos se muestran en la *Tabla 12*:

**Tabla 12.**  
*Parámetros calibrados*

Parámetro	Incremento	Tamaño
Tareas (J)	33.33%	49.99 $\approx$ 50
Máquinas (M)	33.33%	9.99 $\approx$ 10
Trabajadores (W)	33.33%	9.99 $\approx$ 9

Para el parámetro de trabajadores se opta por redondear al entero menor más cercano debido al supuesto 1 que enuncia que el número de máquinas debe ser superior al número de trabajadores. Para el parámetro de máquinas en celdas, se considera el mínimo trabajado en las pruebas, fijando un valor máximo de 4 máquinas por celda, para el tiempo de procesamiento se considera el intervalo más pequeño definido en  $U(5, 10)$ . Para el número de máquinas que pueden procesar la tarea se parte del supuesto 2 que indica que al menos una maquina tiene la capacidad de procesar cualquier tarea, estableciendo así un ideal en que se destina que todas las tareas pueden ser procesadas en todas las máquinas. El número grande expresado por el índice L se destina como en número de variables de  $X_{jmwr}$  más un 10% establecido a criterio de los estudiantes, teniendo así:

$$\text{Número de variables } X_{jmwr} = 50 * 10 * 9 * 6 = 27000$$

$$\text{Número grande } L = 27000 * 1.1 = 29700$$

Para el parámetro tiempo continuo  $r$ , no se tiene definido dentro de las pruebas, por tanto, para asignar un valor a este, se toma como base el supuesto 2 que indica que cada tarea puede ser procesada cualquier máquina y cada máquina puede ser operada por cualquier trabajador, en este orden de ideas, se asigna una operación aritmética sencilla en la que se divide el número de tareas sobre la cantidad de trabajadores, obteniendo así la *Ecuación 12*:

$$\frac{\text{Número de tareas}}{\text{Número de trabajadores}} = \frac{50}{9} = 5.556 \approx 6 \quad (12)$$

Por último, para definir el máximo de celdas los autores plantean la *Ecuación 13* que representa la relación entre el número de máquinas y máximo de máquinas permitidas por celda:

$$\frac{\text{Número de máquinas}}{\text{Máximo de máquinas por celda}} = \frac{10}{4} = 2.5 \approx 3 \quad (13)$$

La *Tabla 13* muestra el resumen de los parámetros fijados para el diseño de experimentos de cada algoritmo y de esa manera determinar cuál o cuáles de las combinaciones de coeficientes en los algoritmos presentan un mejor rendimiento:

**Tabla 13.**

*Parámetros calibrados ajustados*

Parámetro	Tamaño
Tareas (J)	50
Máquinas (M)	10
Trabajadores (W)	9
Tiempos (R)	6
Máquinas por celda (Bu)	4
Celdas (C)	3

*Continuación tabla 13*

Número grande (L)	29700
Tiempo de procesamiento ( $P_{jmw}$ )	U (5,10)

---

### 7. Desarrollo de los algoritmos

Las metaheurísticas planteadas se basan en Algoritmo Genético (GA) tratándose de un primer método en GA puro y un segundo método que combina GA con Optimización por enjambre de partículas (PSO), que forman un algoritmo híbrido denominado GAPSO.

Para definir los parámetros de cada algoritmo se toma como base aquellos valores encontrados en la revisión de literatura que trabajan GA y PSO, este resumen se expone en el Apéndice A. Sin embargo, para el número de individuos se encuentra en dicha revisión que se maneja un máximo de 300 individuos para el GA, mientras que para sus descendencias se tiene un máximo de 800 y un mínimo de 50 para este mismo algoritmo. Para el PSO se cuenta con un número de 300 iteraciones en su máximo nivel y 50 para el mínimo, por tanto, se decide que para comparar ante el mismo número de soluciones obtenidas por cada algoritmo y establecer cual presenta mejor rendimiento es necesario fijar la población en 300 individuos y el número de generaciones y cruces se establece en el intervalo de entre 50 y 200, de esta forma las generaciones realizadas por el GA tendrán sentido, ya que si se decide tomar 300 generaciones el método de selección para el cruce de padres no tiene razón de ser y no interesaría el método. Para los demás parámetros de cada algoritmo, se toman los máximos y mínimos encontrados en la revisión de literatura.

Con el fin de comparar el desempeño y establecer cuál de los dos es mejor en función de cual reduce más el makespan se definen parámetros fijos para cada algoritmo, como se presentan en la *Tabla 14*.

**Tabla 14.**  
*Parámetros fijos para los algoritmos*

<b>Método 1: GA</b>	<b>Método 2: GAPSO</b>
Individuos: 300	Individuos: 300
Número de cruces: 50, 200	Número de iteraciones: 50, 200
Probabilidad de mutación: 10%, 50%	Coefficiente de inercia: 0.75, 1.2
	Coefficiente de aprendizaje personal: 0.97, 2.5
	Coefficiente de aprendizaje social: 0.5, 2.5

---

---

El algoritmo genérico usualmente se diseña con una probabilidad de cruce, sin embargo, para este caso, se cambia la probabilidad de cruce por un numero de descendencia controlado, haciendo que se generen tantas decencias como iteraciones se tengan en GAPSO y de ese modo, poder comparar el rendimiento de los parámetros en el mismo ambiente de generaciones.

De esta forma se establecen dos niveles a examinar determinados como nivel alto y bajo, esto se explica más a detalle en el *capítulo 8*. Debido a que los algoritmos están basados en GA, es de esperarse que compartan secciones en común, tales como la generación de la población inicial, los métodos de cruce y la forma en la que muta la descendencia, sin embargo, aunque funcionan de la misma forma cabe aclarar que no generarán variables iguales.

### **7.1 Algoritmo Genético**

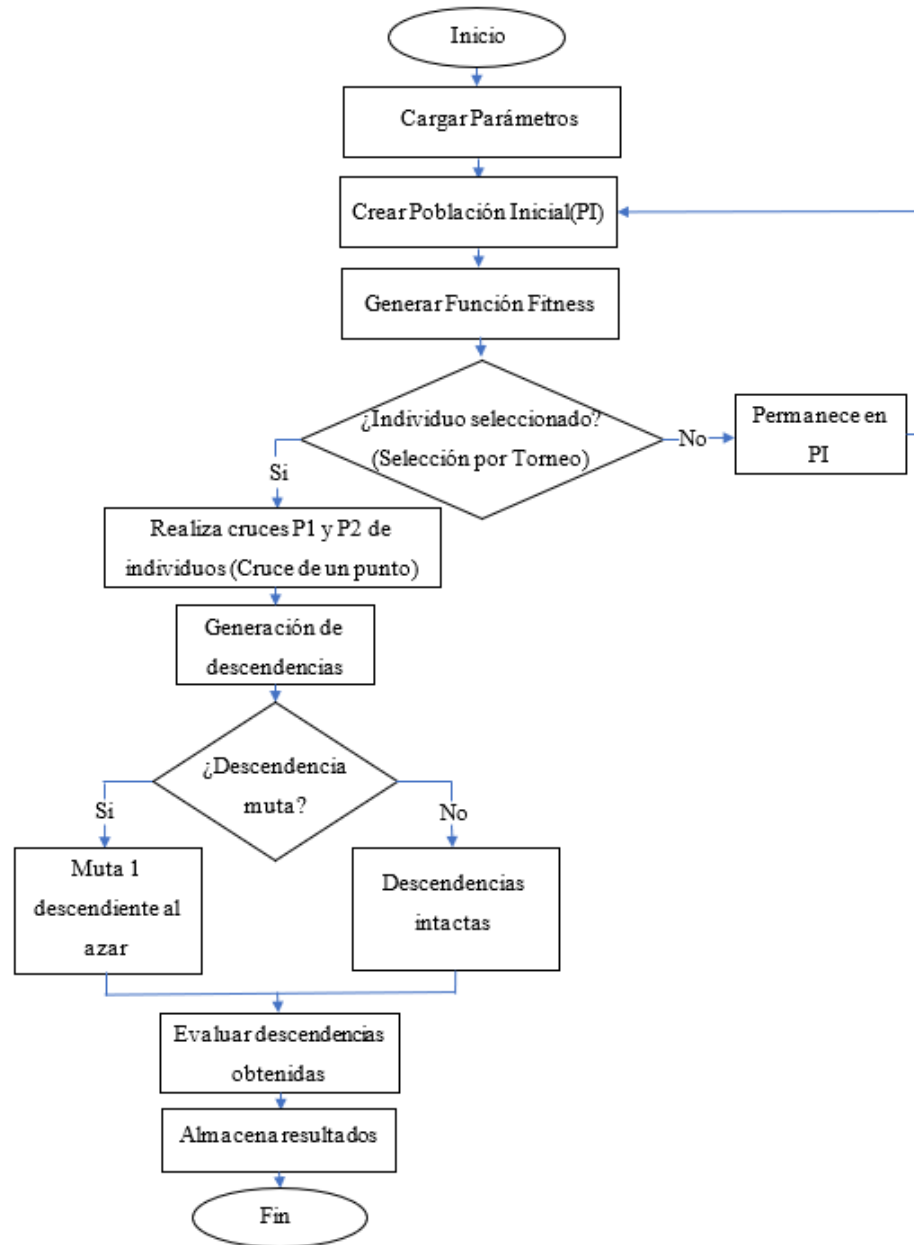
El primer método de solución adaptado al modelo matemático es el algoritmo genético (GA), ya que se trata de un algoritmo evolutivo tiene gran afinidad al momento de manejar variables binarias.

#### **7.1.1 Descripción del algoritmo**

Se inicia con un algoritmo genético canónico, el cual se adecúa al modelo para dar soluciones iniciales (población inicial) y a través de la reproducción y mutación se generen soluciones más adaptadas. La *Figura 10* muestra el cómo se va a implementar este algoritmo:

#### **Figura 10.**

*Diagrama de flujo de GA*



### 7.1.2 Funcionamiento del algoritmo

En esta sección se describe con más detalle la implementación del algoritmo genético:

- **Población inicial**

Este apartado es considerado la base del algoritmo, para ello se establecen entonces las variables del modelo que aportarán en la creación de esta población. Estas son: tiempo de finalización de las tareas ( $FT_j$ ) y la variable binaria de asignación de tareas, maquinas, trabajadores e intervalos de tiempo continuos ( $X_{jmwr}$ ). Las demás variables involucradas

en el modelo matemático no se verán involucradas dentro de las etapas del algoritmo genético, sino que se generan de manera diferente, véase Apéndice C en la sección 2. *Variable Zmc.*

- Se inicia con un valor ideal para  $FT_j$  compuesto por la suma entre los tiempos de procesamiento de cada tarea ( $P_{jmw}$ ) más tiempos de desplazamiento de los trabajadores ( $Y_{mcm'c'}$ ) y el tiempo de procesamiento de la tarea anterior ( $FT_{j-1}$ ). La *Ecuación 14* detalla lo explicado con anterioridad:

$$FT_j = \sum_{m=1}^M \sum_{w=1}^W P_{jmw} + \sum_{c=1}^{C,C'} \sum_{m=1}^{M,M'} Y_{mcm'c'} + FT_{j-1}, \forall j \tag{14}$$

El resultado de la ecuación es un vector de una fila por el número de tareas a procesar ( $VECTOR_{1 \times j}$ ) y este resultado se replica tantas veces como individuos se definieron cruzar, creándose una matriz de soluciones iniciales donde todos los participantes empiezan desde la misma solución. La *Figura 11* muestra a modo de ejemplo la matriz mencionada, sin embargo, no contiene los valores reales del modelo:

**Figura 11.**  
*Estructura de Población inicial FTj*

		FTj				
		j=1	j=2	...	j=J-1	j=J
Individuos	1	20	40	...	80	100
	2	20	40	...	80	100
	⋮	20	40	...	80	100
	n-1	20	40	...	80	100
	n	20	40	...	80	100

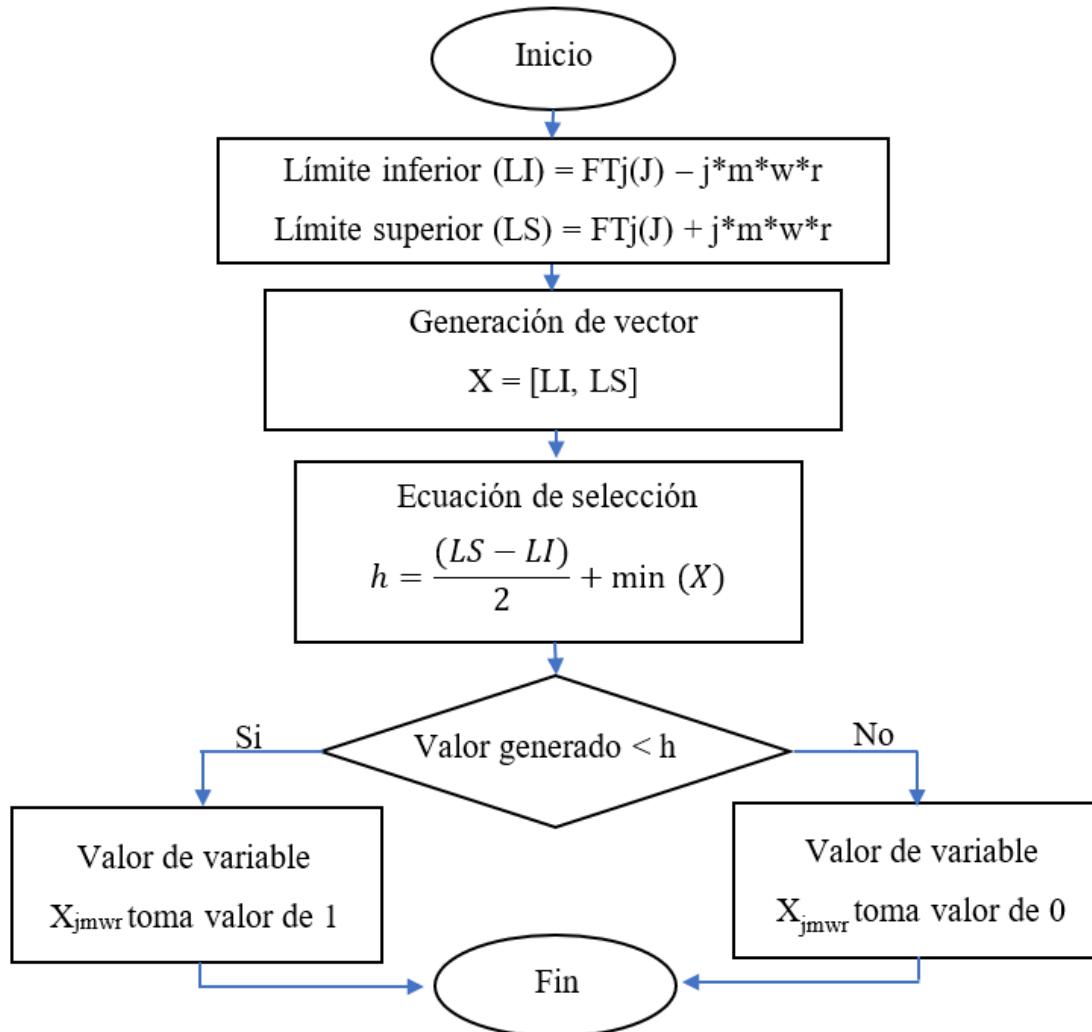
Los valores observados en la figura son acumulados debido a que se da cumplimiento al supuesto 4 que enuncia precedencia en las tareas, por ende, el tiempo de la última tarea representa el makespan.

- En segundo lugar, para generar diversificación en la población inicial, se utilizan los subíndices de la variable  $X_{jmw}$  y el makespan encontrado en la población inicial, esto con la finalidad de asignar dos límites,  $makespan \pm j * m * w * r$ , generando de manera aleatoria un vector de soluciones que cuenta con el doble de

variables de  $X_{jmwr}$ , luego, se implementa una condición para transformar las soluciones creadas de  $FT_j$  en un cifrado binario, esto utilizando el valor medio ( $h$ ) del intervalo de soluciones. En la *Figura 12* se muestra esta etapa de diversificación, con más detalle:

**Figura 12.**

*Diagrama de flujo para soluciones iniciales*



Es de aclarar que la que la variable  $X_{jmwr}$ , es una matriz de 4 dimensiones, sin embargo, por comodidad en la programación en el software, se genera de forma vectorial, replicándose tantas veces como individuos se tengan. La *Figura 13* muestra un ejemplo del resultado de la diversificación:

**Figura 13.**

*Estructura de familia de variables  $X_{jmwr}$*

		$X_{jmwr}$																							
		r=1								r=2															
		w=1				...				w=W				w=1				...				w=W			
		j=1		...		j=J		...		j=1		...		j=J		...		j=1		...		j=J		...	
		m=1	m=2	...	m=M-1	m=M	...	m=1	m=2	...	m=M-1	m=M	...	m=1	m=2	...	m=M-1	m=M	...	m=1	m=2	...	m=M-1	m=M	
Individuos	1	0	1	...	1	0	...	0	0	...	0	1	0	0	...	0	0	...	0	1	...	0	1	0	
	2	1	0	...	0	0	...	1	0	...	1	1	0	0	...	1	1	...	1	0	...	0	0	0	
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	n-1	0	1	...	1	1	...	1	1	...	0	0	1	1	...	0	0	...	1	1	...	1	1	0	
	n	1	0	...	1	0	...	0	0	...	1	1	0	1	...	1	0	...	0	1	...	1	1	0	

- **Generar función de adaptación (Fitness)**

Una vez creada la población inicial y teniendo la asignación de la variable  $X_{jmwr}$ , lo siguiente es identificar los individuos más capacitados para el modelo, para ello se establece la siguiente ecuación que representa la función fitness:

$$F_f = \text{Función objetivo} + \text{penalizaciones} \tag{15}$$

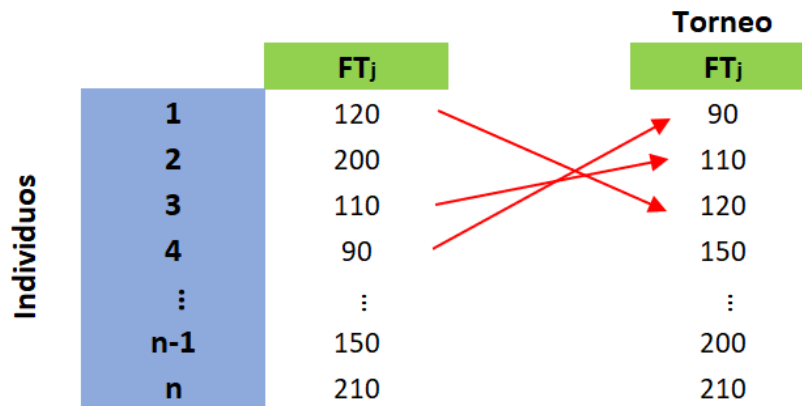
El objetivo de esta función es penalizar aquellos individuos que no cumplen con las restricciones planteadas en el modelo, esta penalización es un incremento porcentual en los tiempos de procesamiento de la tarea que se esté evaluando; en caso de no evaluarse una tarea específica la penalización por incumplimiento será distribuida de manera uniforme a todas las tareas, este proceso es realizado para toda la población. La penalización representa un porcentaje asignado por los estudiantes dependiendo de la restricción que se incumpla, dicho porcentaje es establecido bajo la consideración que existen restricciones más flexibles que otras, es así como se establece que para las restricciones 1 y 4, presentes en las *Ecuación 2* y *Ecuación 5* respectivamente se asignan un incremento del 2% por cada vez que se incumpla la restricción, el porcentaje es asignado debido a que se consideran como restricciones críticas al tener un punto de solución y no una región de soluciones, lo que indica que muy poco factible o es muy difícil de cumplir, por esta razón la penalización es pequeña, mientras, a las restricciones restantes se asigna un porcentaje del 16% a cada una, la penalización es superior ya que en estos casos si se cuentan con regiones de solución y se considera que no serán incumplidas tan fácilmente.

- **Selección**

Considerando que la función de adaptación se aplicó a toda la población, para el algoritmo genético se establece el método de selección por torneo, el cual extrae solo a los mejores

candidatos o los candidatos que mejor se adaptaron y los organiza de mejor al peor. El número de individuos presentes en la selección corresponde al número de cruces decididos a realizar, tomándose así para el algoritmo genético valores de 50 y 200 individuos para crear la matriz de selección. El método de torneo es definido puesto que se asume que los candidatos con menor makespan generarán las mejores descendencias. La *Figura 14* expresa el método de selección por torneo:

**Figura 14.**  
*Selección por torneo*



- **Cruce**

Esta sección expresa como se lleva a cabo el cruce, para este se define el método de cruce de un solo punto, en el que se selecciona un punto de corte al vector  $X_{jmwr}$  para realizar el cambio de información entre el par de padres del torneo, tal como se muestra en la *Figura 15*. El número de descendencias es igual al número de padres seleccionados, puesto que cada pareja de progenitores genera un total de 2 descendencias.

**Figura 15.**  
*Cruce de padres GA*



- **Mutación**

El proceso de mutación se realiza con el fin de alterar aún más el componente genético de cada descendiente, sin embargo, es necesario establecer un controlador que delimite la

cantidad de mutaciones, para esto se maneja la probabilidad de mutación como un parámetro del GA, para establecer si se realiza o no la mutación, se selecciona un porcentaje generado al azar y si ese porcentaje es mejor que la probabilidad de mutación, entonces se realizará dicha acción y seleccionando uno de los dos hijos al azar para mutar, en caso que no ocurra la condición no se hará nada. El proceso de mutación es realizado con cada par de hijos generados del cruce. Es destacable el hecho que se manejan dos tipos de mutación dentro del GA, bajo la consideración que la variable involucrada dentro de este algoritmo híbrido es de naturaleza binaria, los posibles resultados que se obtienen son dos:

-*Mutación por intercambio*: Es destinada en caso de que, al seleccionar dos pines de manera aleatoria, dichos pines sean diferentes, tal como se muestra en la *Figura 16*:

**Figura 16.**  
*Mutación por intercambio GA*

		Xjmwrr											
		r=1					r=2						
		w=1		...	w=W		w=1		...	w=W			
		j=1	...	j=j	...	j=1	...	j=j	...	j=1	...	j=j	
		m=1	m=2	...	m=M-1	m=M	...	m=1	m=2	...	m=M-1	m=M	
Individuos	1	0	1	...	1	0	...	0	0	...	0	1	0
	2	1	0	...	0	0	...	1	0	...	1	1	0
	...	...	...	...	...	...	...	...	...	...	...	...	...
	n-1	0	1	...	1	1	...	1	1	...	0	0	1
	n	1	0	...	1	0	...	0	0	...	1	1	0

-*Mutación por volteo de pin*: Se realiza para aquel par de pines seleccionados que compartan el mismo valor, en este caso se elige uno de estos al azar para ser mutado, en caso de que fuese 0 el valor, este es cambiado a 1, tal como se muestra en la *Figura 17*:

**Figura 17.**  
*Mutación por volteo de pin GA*

		Xjmw <sub>r</sub>											
		r=1											
		w=1					w=W						
		j=1	...	j=J	...	j=1	...	j=J	...	j=J			
		m=1	m=2	...	m=M-1	m=M	...	m=1	m=2	...	m=M-1	m=M	m=1
Individuos	1	0	1	...	1	0	...	0	0	...	0	1	0
	2	1	0	...	0	0	...	1	0	...	1	1	0
	...	...	...	...	...	...	...	...	...	...	...	...	...
	n-1	0	1	...	1	1	...	1	1	...	0	0	1
	n	1	0	...	1	0	...	0	0	...	1	1	0

- **Evaluar descendencias**

Por último, se realiza nuevamente la evaluación para determinar si se encontró una mejora dentro de las soluciones examinadas, esta solución es mejor con respecto a las soluciones iniciales.

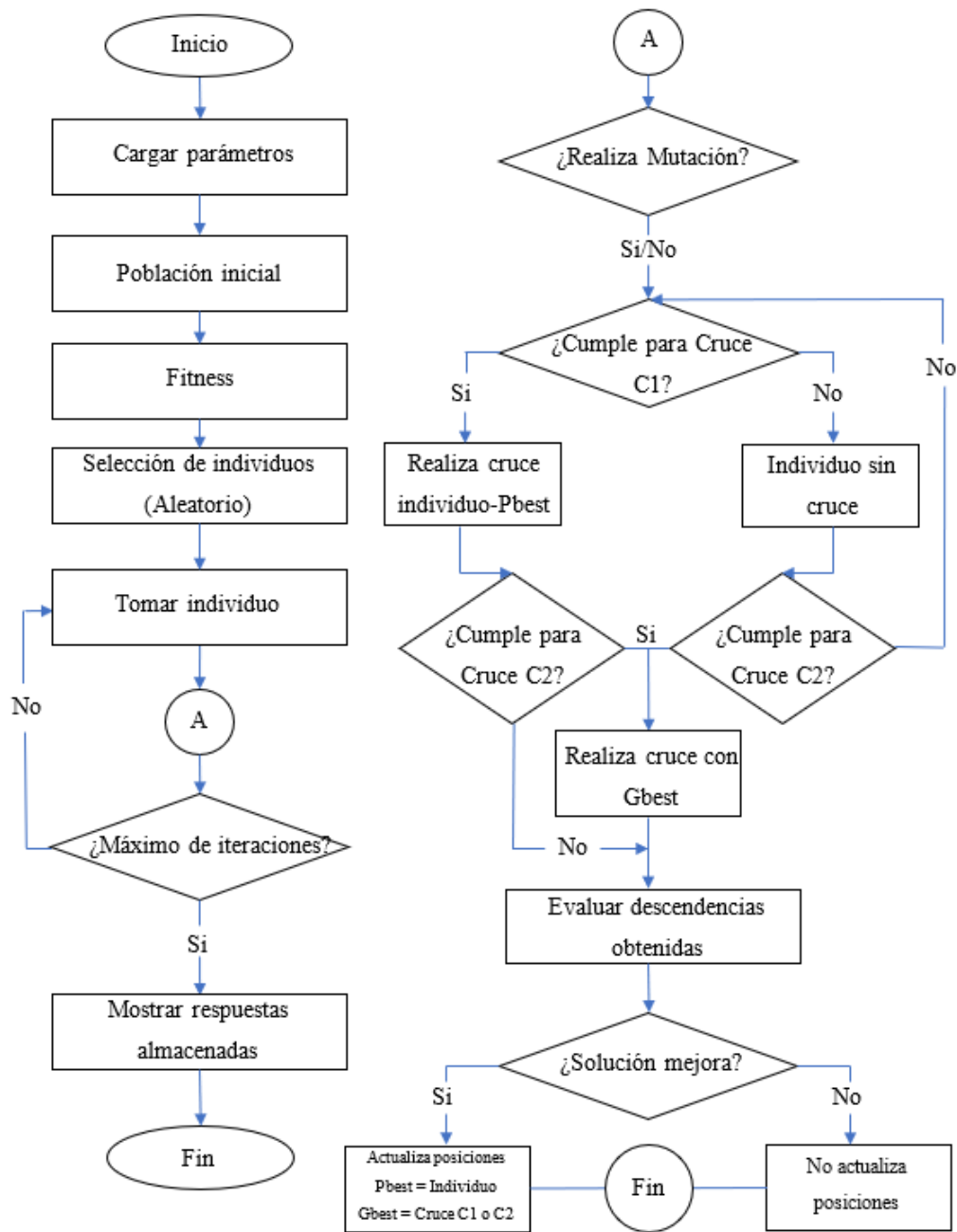
### 7.2 Algoritmo genético híbrido

Para el algoritmo genético híbrido se plantea como base GA y se adapta el algoritmo Optimización por Enjambre de Partículas (PSO) como un complemento con el fin de mejorar sustancialmente la calidad de respuestas.

#### 7.2.1 Descripción del algoritmo

Como se menciona anteriormente el algoritmo híbrido consiste en un GA al cual se le adapta PSO, sin embargo, para esta adaptación se es necesario tener claro el funcionamiento del algoritmo, para ello la *Figura 18* muestra cada paso de este algoritmo:

**Figura 18**  
*Diagrama de flujo GAPSO*



No

**7.2.2 Funcionamiento del algoritmo**

En esta sección se muestra los criterios para el funcionamiento del algoritmo híbrido.

- **Población inicial**

Como se mencionó antes, existen partes que comparten los dos algoritmos, para el algoritmo híbrido GAPS0, la población inicial, se genera exactamente igual a como se

describe en el *capítulo 7*, para el algoritmo genético, en la *sección 7.1.2* en el ítem *Generar función de adaptación (Fitness)* o se encuentra el origen de la referencia..

Como uno de los propósitos determinar que algoritmo se comporta mejor, la generación de la variable  $FT_j$  sin penalización tiene exactamente los mismos valores en los dos algoritmos.

- Función de adaptación (Fitness)

La función fitness es tomada y penalizada exactamente igual que como se expone en el *capítulo 7*, para el algoritmo genético, en la *sección 7.1.2* en el ítem *Generar función de adaptación (Fitness)*.

Al igual que en la sección inmediatamente anterior, como el propósito es comparar cual algoritmo es mejor las poblaciones iniciales son generadas exactamente igual, con las mismas variables para  $X_{jmwr}$  que se generan en GA.

- Selección

Considerando que la función de adaptación se aplicó a toda la población, para el algoritmo híbrido se establece el método de selección de manera aleatoria, es decidido así bajo la premisa que la posición del individuo más apto ya se conoce, la aleatoriedad de los cruces entre el mejor individuo y otro elegido al azar generará un espacio de búsqueda con una mayor diversidad. Se seleccionan tantos individuos como iteraciones se deseen realizar.

- Optimización por Enjambre de Partículas (PSO)

Luego de la selección, es cuando se incorpora el PSO, para el cual, se parte del punto que el PSO tiene tres componentes fundamentales, el primero es el componente de velocidad, el segundo, la información recopilada por cada individuo y por último la información de la mejor partícula. De esta forma se adapta cada uno de los componentes del PSO al GA.

#### ***Componente de velocidad***

El componente de velocidad es acompañado por un coeficiente de inercia, que determina la velocidad con la que aprende cada partícula, en este caso el componente de velocidad es considerado como la mutación de los individuos y el coeficiente de inercia, será el punto de partida para definir si se realiza o no dicha mutación. Para llevarse a cabo la mutación, se seleccionan dos números de manera aleatoria en el rango de variables de  $X_{jmwr}$ , una vez seleccionados, se involucra el coeficiente de inercia ( $nn$ ), el cual se toma como base para determinar dos números de manera aleatoria también, estos, se encuentran en un rango

entre cero y el coeficiente de inercia, para llevarse a cabo la mutación se debe cumplir que el requisito expuesto en la *Ecuación 15*

$$\# Aleatorio [0, nn] < \# Aleatorio [0, nn] \tag{15}$$

Al realizar de manera aleatoria la mutación se busca la diversificación de las descendencias, este proceso es realizado con cada uno de los individuos involucrados en la selección. Es destacable el hecho que se manejan dos tipos de mutación dentro del GAPSO, bajo la consideración que la variable involucrada dentro de este algoritmo híbrido es de naturaleza binaria, los posibles resultados que se obtienen son dos:

-*Mutación por intercambio*: Es destinada en caso de que, al seleccionar dos pines de manera aleatoria, dichos pines sean diferentes, tal como se muestra en la *Figura 19*.

**Figura 19.**  
*Mutación por intercambio*

		Xjmwrr											
		r=1											
		w=1					w=W						
		j=1		j=J			j=1		j=J			j=	
		m=1	m=2	...	m=M-1	m=M	...	m=1	m=2	...	m=M-1	m=M	m=1
Individuos	1	0	1	...	1	0	...	0	0	...	0	1	0
	2	1	0	...	0	0	...	1	0	...	1	1	0
	...	...	...	...	...	...	...	...	...	...	...	...	...
	n-1	0	1	...	1	1	...	1	1	...	0	0	1
	n	1	0	...	1	0	...	0	0	...	1	1	0

-*Mutación por volteo de pin*: Se realiza para aquel par de pines seleccionados que compartan el mismo valor, en este caso se elige uno de estos al azar para ser mutado, en caso de que fuese 0 el valor, este es cambiado a 1, tal como se muestra en la *Figura 20*.

**Figura 20.**  
Mutación por volteo de pin

Individuos	X <sub>jmw</sub> r											
	r=1											
	w=1					w=W						
	j=1	...	j=J	...	j=1	...	j=J	...	j=1	...	j=J	
m=1	m=2	...	m=M-1	m=M	...	m=1	m=2	...	m=M-1	m=M	m=1	
1	0	1	...	1	0	...	0	0	...	0	1	0
2	1	0	...	0	0	...	1	0	...	1	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...
n-1	0	1	...	1	1	...	1	1	...	0	0	1
n	1	0	...	1	0	...	0	0	...	1	1	0

**Componente de aprendizaje personal**

En esta sección se presenta el primer cruce del GA con componentes de PSO, se destina un coeficiente de aprendizaje personal (C1) y de manera análoga a la Ecuación 15 es este el filtro el que determinar si se realiza el cruce, dicho cruce se presenta entre uno de los individuos tomados de la selección y la partícula que guarda la mejor información para FTj (Pbest), para el cruce número 1 realizado, se considera a Pbest como el primer individuo de la selección. El método de cruce destinado es cruce de un punto, en donde se selecciona un numero aleatorio y este será el punto de corte del vector X<sub>jmw</sub>r para intercambiar la información entre los dos padres. La Figura 21 muestra la forma en que se realiza dicho cruce.

**Figura 21.**  
Cruce Individuo-Pbest GAPSO



**Componente de aprendizaje social**

Al igual que el cruce 1, en este segmento se realiza cruce de un punto, al igual que antes se presenta de manera análoga la Ecuación 15 pero se evalúa el coeficiente de aprendizaje social (C2) y que a diferencia del cruce anterior, en este se obtiene un máximo de 4 descendencias, esto se cumple si el primer cruce lleva a cabo, obteniendo así un cruce entre

la descendencia del cruce 1 y la partícula con mejor solución de toda la población (Gbest), en caso de no ser así, el cruce es realizado entre el individuo proveniente de la selección y Gbest. La *Figura 22* muestra el cruce realizado entre un hijo y Gbest.

**Figura 22.**

*Cruce hijo-Gbest GAPSO*



En caso de que no se realice ningún cruce se entra en un bucle que obliga que hasta no obtener descendencias no se sale de dicho bucle, esto es puesto que, si no se realiza un cruce, cabe la posibilidad que no se haya realizado una mutación tampoco y esto conlleva a analizar la misma respuesta ya que se trata del mismo individuo proveniente de la selección, por tanto, el bucle deja de realizarse hasta que con ese individuo se realice al menos un cruce.

- Evaluar partículas y actualizar posiciones

Una vez el proceso de cruce termina, las descendencias obtenidas son sometidas a una evaluación para establecer si se encontró una mejor solución, en caso de ser así, aquel individuo que realizó el cruce 1 pasa a ser el nuevo Pbest y la respuesta evaluada que presentó el mejor desempeño dentro de los hijos pasa a tomar el lugar de Gbest. Todo el procedimiento se repite tantas veces como iteraciones se establecido realizar.

Como último paso, se selecciona aquella respuesta que presenta el menor Makespan dentro de todas las iteraciones realizadas y es este resultado el que se puede observar en la ventana de salida de MATLAB.

En el Apéndice D se muestra la manera en la que se encuentran programados los dos algoritmos, al ejecutarse se puede observar el tiempo obtenido por cada algoritmo.

## 8. Calibración de los algoritmos

Los parámetros de cada algoritmo son mostrados en la *Tabla 14* para estos se pretende saber cuál o cuáles factores son los que inciden de manera significativa dentro de cada modelo, por lo que se plantean dos diseños de experimentos factorial de tipo  $2^k$ , uno para cada algoritmo.

### 8.1 Diseño $2^k$ para GA

Para el caso del GA se plantea un diseño tipo  $2^2$  en el software Minitab 2022, la *Figura 23* muestra la designación de factores, así como los niveles manejados para el desarrollo del experimento.

**Figura 23.**  
*Diseño  $2^k$  GA*

		Factores	
		Número de padres	Probabilidad de mutación (%)
Nivel	Alto (+)	200	50
	Bajo(-)	50	10

Como se mencionó con anterioridad estos son valores tomados de la revisión de literatura realizada en el capítulo *¡Error! No se encuentra el origen de la referencia.*. El diseño se lleva a cabo bajo los parámetros expuestos en la *Tabla 14* con 20 réplicas por cada combinación y estudio de factores, obteniéndose un total de 80 muestras. En el Apéndice E se encuentran los datos promedio de cada replica tomada. la *Figura 24* permite observar la tabla de análisis de varianza (ANOVA) así como la información de los coeficientes usados, esto a un nivel de confianza del 95%.

**Figura 24.**  
*ANOVA GA*

**Análisis de Varianza**

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	3	49331707	16443902	0,67	0,576
Lineal	2	43258092	21629046	0,88	0,421
Número de padres	1	6669701	6669701	0,27	0,605
Prob. Mutación	1	36588391	36588391	1,48	0,227
Interacciones de 2 términos	1	6073615	6073615	0,25	0,621
Número de padres*Prob. Mutación	1	6073615	6073615	0,25	0,621
Error	76	1877261678	24700812		
Total	79	1926593385			

**Coeficientes**

Término	Coef	EE del coef.	Valor T	Valor p	FIV
Constante	99753	556	179,52	0,000	
Número de padres					
50	-289	556	-0,52	0,605	1,00
Prob. Mutación					
10	-676	556	-1,22	0,227	1,00
Número de padres*Prob. Mutación					
50 10	276	556	0,50	0,621	1,00

Nota: Adaptado del software Minitab 21.2.

Como se puede observar, según ANOVA ningún factor de manera independiente o la combinación de ellos presenta efectos significativos sobre el modelo, lo que indica que para el GA es indiferente el número de padres cruzados y/o la probabilidad de mutación, sin embargo, se observa que para una probabilidad de mutación del 10% se muestra un efecto alto con relación a los demás factores, tal como se observa en la *Figura 24*.

El Apéndice F muestra el diseño de experimento realizado en Minitab.

**8.2 Diseño 2<sup>k</sup> GAPSO**

Para el caso del GAPSO se plantea un diseño tipo 2<sup>4</sup> en el software Minitab 2022, la *Figura 25* muestra la designación de factores, así como los niveles manejados para el desarrollo del experimento.

**Figura 25.**

*Diseño 2k GAPSO*

		Factores			
		Número de iteraciones	Coef. Inercia	Coef. Personal	Coef. Social
Nivel	Alto (+)	200	1,2	2,5	2,5
	Bajo(-)	50	0,75	0,97	0,5

Al igual que para el GA, en este caso, estos valores corresponden a los presentes en la revisión de literatura y es corrido bajo los mismos parámetros expuestos en la *Tabla 14*, sin embargo, para este segundo algoritmo se establecen 10 réplicas por cada combinación y estudio de factores, obteniéndose un total de 160 muestras. En el Apéndice E se encuentran los datos promedio obtenidos de cada réplica hecha. La *Figura 26* permite ver la tabla ANOVA realizada para el GAPSO a un nivel de confianza del 95%, mientras que la *Figura 27*, permite ver los coeficientes que mejor desempeño presentan.

**Figura 26.**  
ANOVA GAPSO

Análisis de Varianza

Fuente	GL	Fuente	Valor p
Modelo	15	Modelo	0,003
Lineal	4	Lineal	0,173
Coef. Inercia	1	Coef. Inercia	0,766
Coef. Aprendizaje personal	1	Coef. Aprendizaje personal	0,535
Coef. Aprendizaje social	1	Coef. Aprendizaje social	0,040
Iteraciones	1	Iteraciones	0,194
Interacciones de 2 términos	6	Interacciones de 2 términos	0,000
Coef. Inercia*Coef. Aprendizaje personal	1	Coef. Inercia*Coef. Aprendizaje personal	0,564
Coef. Inercia*Coef. Aprendizaje social	1	Coef. Inercia*Coef. Aprendizaje social	0,355
Coef. Inercia*Iteraciones	1	Coef. Inercia*Iteraciones	0,854
Coef. Aprendizaje personal*Coef. Aprendizaje social	1	Coef. Aprendizaje personal*Coef. Aprendizaje social	0,065
Coef. Aprendizaje personal*Iteraciones	1	Coef. Aprendizaje personal*Iteraciones	0,136
Coef. Aprendizaje social*Iteraciones	1	Coef. Aprendizaje social*Iteraciones	0,000
Interacciones de 3 términos	4	Interacciones de 3 términos	0,774
Coef. Inercia*Coef. Aprendizaje personal*Coef. Aprendizaje social	1	Coef. Inercia*Coef. Aprendizaje personal*Coef. Aprendizaje social	0,935
Coef. Inercia*Coef. Aprendizaje personal*Iteraciones	1	Coef. Inercia*Coef. Aprendizaje personal*Iteraciones	0,897
Coef. Inercia*Coef. Aprendizaje social*Iteraciones	1	Coef. Inercia*Coef. Aprendizaje social*Iteraciones	0,299
Coef. Aprendizaje personal*Coef. Aprendizaje social*Iteraciones	1	Coef. Aprendizaje personal*Coef. Aprendizaje social*Iteraciones	0,410
Interacciones de 4 términos	1	Interacciones de 4 términos	0,324
Coef. Inercia*Coef. Aprendizaje personal*Coef. Aprendizaje social*Iteraciones	1	Coef. Inercia*Coef. Aprendizaje personal*Coef. Aprendizaje social*Iteraciones	0,324
Error	144	Error	
Total	159	Total	

Nota: Adaptado del software Minitab 21.2.

**Figura 27.**  
Coeficientes GAPSO a partir de ANOVA

## Coeficientes

Término	Coef	Término	Valor p	FIV
Constante	100292	Constante	0,000	
Coef. Inercia 0,75	-121	Coef. Inercia 0,75	0,766	1,00
Coef. Aprendizaje personal 0,97	252	Coef. Aprendizaje personal 0,97	0,535	1,00
Coef. Aprendizaje social 0,5	-839	Coef. Aprendizaje social 0,5	0,040	1,00
Iteraciones 50	-529	Iteraciones 50	0,194	1,00
Coef. Inercia*Coef. Aprendizaje personal 0,75 0,97	235	Coef. Inercia*Coef. Aprendizaje personal 0,75 0,97	0,564	1,00
Coef. Inercia*Coef. Aprendizaje social 0,75 0,5	-376	Coef. Inercia*Coef. Aprendizaje social 0,75 0,5	0,355	1,00
Coef. Inercia*Iteraciones 0,75 50	75	Coef. Inercia*Iteraciones 0,75 50	0,854	1,00
Coef. Aprendizaje personal*Coef. Aprendizaje social 0,97 0,5	-755	Coef. Aprendizaje personal*Coef. Aprendizaje social 0,97 0,5	0,065	1,00
Coef. Aprendizaje personal*Iteraciones 0,97 50	-608	Coef. Aprendizaje personal*Iteraciones 0,97 50	0,136	1,00
Coef. Aprendizaje social*Iteraciones 0,5 50	-1828	Coef. Aprendizaje social*Iteraciones 0,5 50	0,000	1,00
Coef. Inercia*Coef. Aprendizaje personal*Coef. Aprendizaje social 0,75 0,97 0,5	-33	Coef. Inercia*Coef. Aprendizaje personal*Coef. Aprendizaje social 0,75 0,97 0,5	0,935	1,00
Coef. Inercia*Coef. Aprendizaje personal*Iteraciones 0,75 0,97 50	52	Coef. Inercia*Coef. Aprendizaje personal*Iteraciones 0,75 0,97 50	0,897	1,00
Coef. Inercia*Coef. Aprendizaje social*Iteraciones 0,75 0,5 50	-423	Coef. Inercia*Coef. Aprendizaje social*Iteraciones 0,75 0,5 50	0,299	1,00
Coef. Aprendizaje personal*Coef. Aprendizaje social*Iteraciones 0,97 0,5 50	335	Coef. Aprendizaje personal*Coef. Aprendizaje social*Iteraciones 0,97 0,5 50	0,410	1,00
Coef. Inercia*Coef. Aprendizaje personal*Coef. Aprendizaje social*Iteraciones 0,75 0,97 0,5 50	401	Coef. Inercia*Coef. Aprendizaje personal*Coef. Aprendizaje social*Iteraciones 0,75 0,97 0,5 50	0,324	1,00

Nota: Adaptado del software Minitab 21.2.

Como se puede observar, según la tabla ANOVA existen varios factores que presentan efecto significativo sobre el GAPSO, para ello se tiene en cuenta varios casos:

### 8.2.1 Caso 1. Factores individuales:

Dentro de los factores de manera individual se tiene que únicamente el coeficiente de aprendizaje social tiene efecto significativo sobre el GAPSO, sin embargo, es el nivel bajo aquel que incide, tomando así un valor de 0.5.

### 8.2.2 Caso 2. Interacción de dos factores:

Al combinar dos factores, se tiene que solo el coeficiente de aprendizaje social y en número de iteraciones tiene efecto significativo sobre el GAPSO, lo que refuerza que dicho coeficiente de aprendizaje social influye dentro del algoritmo. Aunque a un nivel de confianza del 95% no se acepta ninguna otra combinación de dos factores, es destacable la combinación entre el coeficiente de aprendizaje personal y el de aprendizaje social, esto puesto que se encuentra cerca de representar efecto significativo con un alfa de 5%, por tanto, es bueno tener en consideración la combinación de estos dos factores también. Para la primera combinación de factores mencionada, se define los

niveles bajos de nuevo, quedando para el coeficiente social un valor de 0.5 y el número de iteraciones como 50.

### **8.2.3 Caso 3. Interacción de tres factores:**

Al realizarse la combinación de tres factores se encuentra que ninguno de ellos tiene efecto significativo sobre el funcionamiento del algoritmo, no obstante, la combinación entre el coeficiente de inercia, coeficiente social y el número de iteraciones se muestra como el más relevante, ayudando encontrar un patrón en el que la combinación vista en dos factores ayuda a la calibración de otros factores.

### **8.2.4 Caso 4. Interacción de cuatro factores:**

En este caso se rechaza la interacción de los cuatro factores, sin embargo, aunque se rechaza debido a que el valor p mostrado en ANOVA por no ser superior a un nivel de confianza del 95% es interesante considerar esta como un posible punto de partida para calibrar todos los factores del GAPSO, bajo la consideración que involucra la interacción de los dos factores aceptados y además proporciona la mejor combinación del diseño factorial.

### **8.2.5 Decisión para GAPSO**

Es así como después de considerar todos los resultados obtenidos se opta por tomar el nivel bajo como los parámetros seleccionados para la ejecución de GAPSO en MATLAB. Tanto en GA como en GAPSO se considera el nivel bajo de la cantidad de padres a cruzar y número de iteraciones puesto que se pretende reducir el tiempo computacional. Al definir los padres y las iteraciones en el mismo número se examinan la misma cantidad de soluciones para cada algoritmo generando que para la evaluación del desempeño de estos se comparen sobre el mismo tamaño en su campo de búsqueda. En el Apéndice G se muestra el diseño realizado en el software Minitab.

## **9. Comparación del desempeño de los algoritmos**

Para el evaluar el desempeño de los algoritmos, los resultados son sometidos a un análisis de una prueba T para dos muestras, con el fin de determinar si existe diferencia entre las medias de los makespan obtenidos por cada algoritmo, sin embargo, para este análisis, los parámetros del modelo matemático son diferentes a los usados en el diseño de experimento, esto se hace bajo la aclaración que se desea ampliar la cantidad de variables involucradas, es por esto que se opta por manejar 66.66% siguiendo la analogía explicada para los resultados expuestos en la *Tabla 13*, no obstante, los parámetros de cada algoritmo se mantienen según lo analizado y obtenido en el

capítulo 8. La *Tabla 15* muestra el resumen de los parámetros establecidos para el modelo matemático.

**Tabla 15.**  
*Parámetros definidos para resultados*

Parámetro	Incremento	Tamaño
Tareas (J)	66.66%	99.99 $\approx$ 100
Máquinas (M)	66.66%	19.99 $\approx$ 20
Trabajadores (W)	66.66%	16.665 $\approx$ 17
Tiempo continuo (R)	-	6
Máquinas por celda (Bu)	-	4
Celdas (C)	-	5
Número Grande (L)	-	224400
Tiempo de procesamiento ( $P_{jmw}$ )	-	U (5, 10)

Bajo los parámetros expuestos en la *Tabla 15* junto con los niveles definidos en la *Tabla 16* para los algoritmos se realiza la prueba que proporciona los datos necesarios para someter los algoritmos al estudio de la prueba T.

**Tabla 16.**  
*Resumen parámetros para algoritmo*

Método 1: GA	Método 2: GAPSO
Individuos: 300	Individuos: 300
Número de cruces: 50	Número de iteraciones: 50
Probabilidad de mutación: 10%	Coeficiente de inercia: 0.75
	Coeficiente de aprendizaje personal: 0.97
	Coeficiente de aprendizaje social: 0.5

La *Tabla 17* muestra la cantidad de variables involucradas en la prueba bajo la cual se establecen las conclusiones.

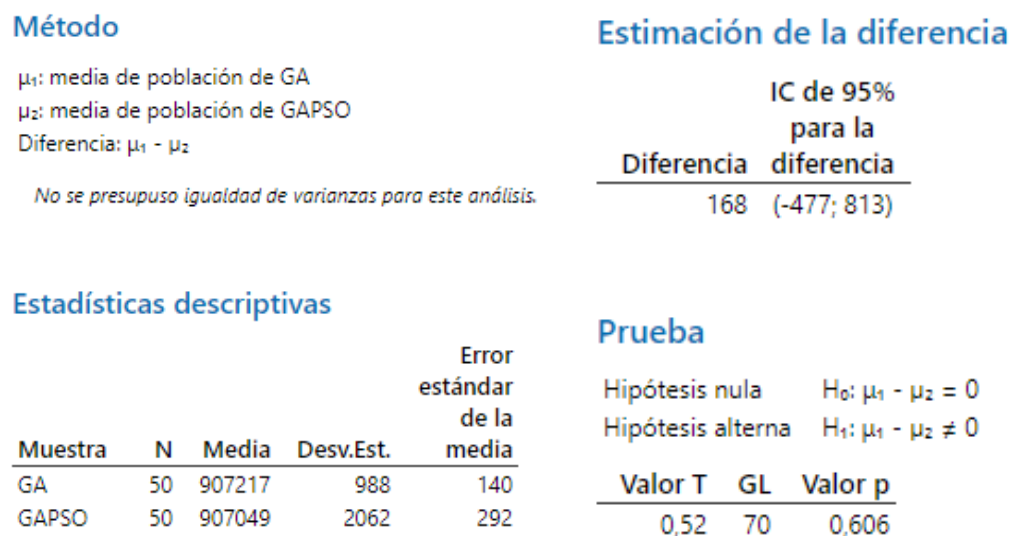
**Tabla 17.**  
*Cantidad de variables involucradas*

Variable	Cantidad de variables
Xjmwrr	204000
FTj	100
Zmc	100

Cabe aclarar que la cantidad de variables presentes hacen referencia para un solo individuo, bajo la consideración que se tiene una población de 300, la cantidad de variables en total para toda la población puede expresarse como la suma de las variables de la *Tabla 17* multiplicado por 300, dando un total para toda la población de 61.26 millones de variables.

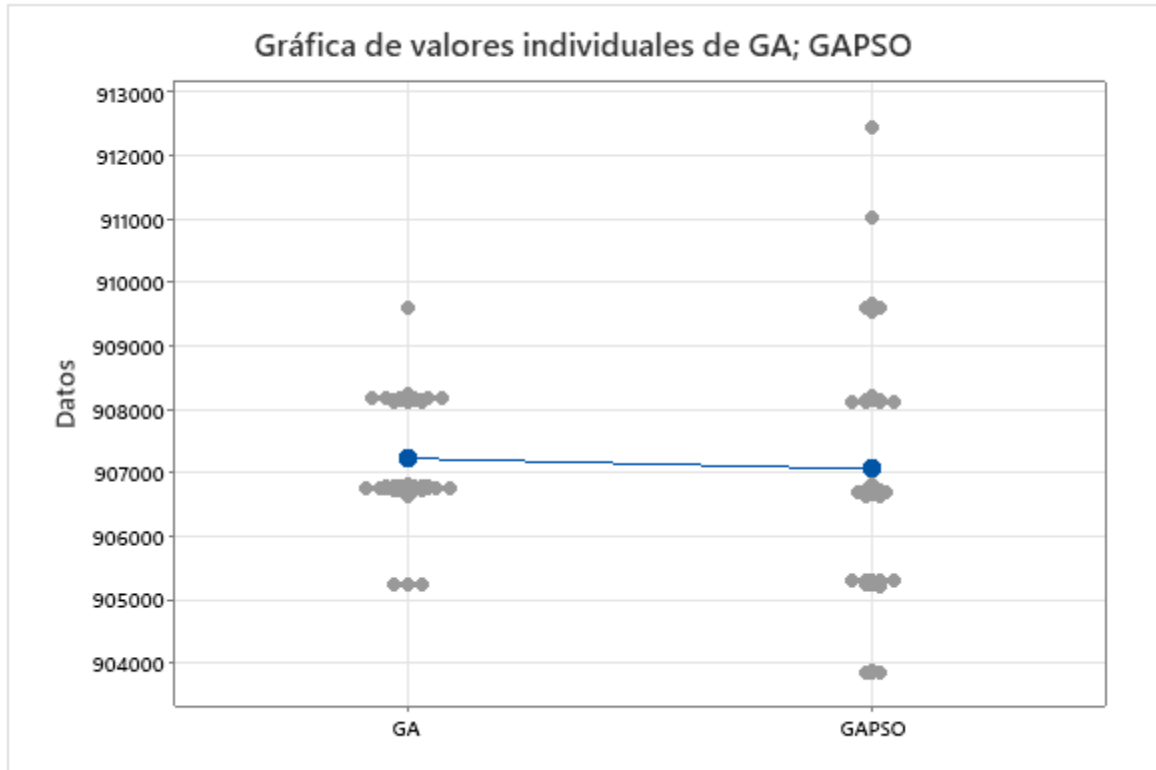
La *Figura 28* muestra los resultados obtenidos en la prueba T a partir de cual se realiza el análisis necesario para establecer conclusiones.

**Figura 28.**  
*Prueba estadística T*



Nota: Adaptado de software Minitab 21.2.

A un nivel de confianza del 95% no existe evidencia estadística para rechazar la hipótesis nula, por tanto, se concluye que no existe diferencia entre los makespan medios obtenidos por cada algoritmo. En la *Figura 29* se muestra que el espacio de búsqueda de GAPSO es superior al de GA, dando a entender que existe más diversidad de soluciones en el algoritmo híbrido. En el Apéndice H se puede observar la prueba T realizada a los resultados obtenidos.

**Figura 29.***Gráfico de medias prueba T*

Nota: Adaptado del software Minitab 21.2.

Partiendo de la información proporcionada en el *Figura 28* es de notar que la prueba T realizada no presenta diferencia entre los valores medios de cada algoritmo, sin embargo, en la *Figura 29* se puede notar que el espacio de búsqueda para soluciones del GAPSO es superior. No obstante, es necesario considerar el valor mínimo de cada grupo de soluciones puesto que se busca la minimización del tiempo máximo de procesamiento de tareas, es bajo esta consideración que se define el mínimo makespan obtenido por cada algoritmo. Los resultados generados en el software MATLAB 2022a son obtenidos al ser corridos en un ordenador con sistema operativo Windows 11 de procesador Intel Core™ i5-10300H con 8 GB de RAM, esto para los parámetros definidos en la *Tabla 15*.

La *Tabla 18* muestra tres valores obtenidos al correr el modelo matemático junto con las metaheurísticas en el software MATLAB. En primer lugar, se encuentra el mínimo tiempo

obtenido de aquella matriz de soluciones iniciales. Luego se tiene el makespan obtenido por el algoritmo genético y por último se muestra la solución proporcionada por el algoritmo genético híbrido.

**Tabla 18.**  
*Ejemplo de implementación de los algoritmos*

	Solución inicial	Algoritmo genético	Algoritmo híbrido
<b>Makespan</b>	905.238	905.224	903.839
<b>Unidades de tiempo reducidas</b>	NA	14	1399
<b>Tiempo de corrida</b>	NA	4.681,8 seg	6.066,1 seg
<b>% de tiempo con relación al total</b>	NA	43,540268%	56,414118%

Nota: Adaptado de software MATLAB 2022a

Como se observa en la *Tabla 18* el makespan es menor en el algoritmo híbrido, presentando así un mejor resultado comparado con el GA, sin embargo, es de notar que el GA cuenta con un menor tiempo computacional en relación con el tiempo total transcurrido, siendo inferior al tiempo empleado por GAPS0. En total el tiempo de corrida, desde la generación de variables hasta la muestra de los resultados es de 10.752,80 segundos aproximadamente, el equivalente a 2.98 horas lo cual se considera un tiempo razonable considerando la cantidad de variables involucradas.

El algoritmo híbrido siempre presenta una mejor solución al momento de disminuir el makespan, aspecto que no se observa en GA, pues dentro de las pruebas realizadas, en ocasiones el makespan era superior al resultado sin aplicar algoritmo.

$$\frac{\text{Mejora de GA}}{\text{Mejora de GAPS0}} * 100 = \frac{14}{1399} * 100 = 98.99\% \quad (16)$$

La *Ecuación 16* es el porcentaje de mejora para los datos expuestos en la *Tabla 18*.

Bajo lo expuesto anteriormente se puede considerar superior el algoritmo genético híbrido GAPS0 puesto que, aunque presenta un tiempo de computacional mayor, su solución siempre es menor y en el mayor número de casos es superior a la del GA. En ese orden de ideas el GAPS0 presenta un mejor rendimiento en comparación con GA por las razones expuestas

anteriormente, siendo superior en un 98.99% con respecto al GA, tal como se observa en la *Ecuación 16*.

## 10. Conclusiones

Resumiendo lo planteado y empezando por el análisis bibliométrico se puede concluir que la temática de manufactura celular es un tema de investigación vigente. Además, se puede observar el impacto de la crisis sanitaria (COVID-19) sobre todas las áreas de investigación, teniendo presente que el número de publicaciones relacionadas con el tema de investigación por año presentan una tendencia promedio decreciente del 23,64%.

Por otra parte, la revisión de literatura deja en evidencia que el criterio de optimización (minimización del makespan), las problemáticas a solucionar (programación y formación de celdas) y el método de solución implementado en el modelo matemático en el cual se centra el presente trabajo, son de gran interés para los investigadores y se encuentran con mayor frecuencia en los artículos estudiados.

Del modelo matemático se puede concluir que presenta una complejidad NP-hard y no es posible llegar a una solución exacta, por ende, se deben implementar metaheurísticas que se adapten y enfoquen en alcanzar el óptimo.

En cuanto a los algoritmos implementados en el software MATLAB 2022a se llega a mejores soluciones factibles de makespan comparado con las soluciones iniciales sin ninguna metaheurística. Comparando ambos algoritmos es posible concluir que si se evalúa mediante el cumplimiento del criterio de optimización el algoritmo con mejores resultados es el algoritmo genético híbrido GAPSO, mientras que si evaluamos el tiempo promedio de computación aquel algoritmo con menor tiempo para llegar a una solución factible es el algoritmo genético (GA), estas dos conclusiones están relacionadas con el espacio de búsqueda del GAPSO, pues este es mayor generando mejores soluciones pero ejecutando por más tiempo el algoritmo. Ahora bien, hablando de rendimiento y análisis individual obtenemos que la cantidad de tiempo reducida promedio del GA respecto al GAPSO es del 98,9% logrando una reducción significativa del makespan.

Por otra parte, el diseño de experimentos aplicado para cada algoritmo permite concluir en primera instancia que en el GA ningún factor de manera independiente o combinada presenta efectos significativos sobre el modelo matemático, esto puede deberse a que sin importar la

cantidad de cruces los resultados obtenidos no presentan un cambio drástico dentro de la optimización. Mientras que para el GAPSO si existen factores individuales y combinación entre estos que presentan efectos significativos sobre el algoritmo híbrido, lo que indica que cuanto más flexible o grande es el coeficiente social y coeficiente personal generan muchas más descendencias, perimiendo así que el espacio de búsqueda sea más amplio.

En segunda instancia el diseño  $2^4$  para el algoritmo GAPSO permite concluir que individualmente el coeficiente de aprendizaje social en su nivel bajo presenta un efecto significativo sobre el modelo, esto se deduce bajo la aclaración que, aunque el número de iteraciones es significativo, es indiferente si se realizan 50 o 200, puesto que dentro de las pruebas realizadas no se encuentra una diferencia significativa con relación a estos dos niveles.

La prueba T nos permite ver que no existe diferencia dentro del promedio de valores de cada algoritmo, por tanto, se toma el valor mínimo de cada paquete de soluciones proporcionadas por los algoritmos para definir cual es mejor. En caso de que las medias de cada algoritmo fuesen diferentes se puede definir con una mayor precisión cuál de los dos algoritmos es superior, sin embargo, se observa que, aunque estadísticamente las medias son iguales la media del GAPSO presenta una tendencia a ser el valor menor.

## 11. Recomendaciones

### *Para futuras investigaciones se recomienda:*

Implementar otros métodos de solución canónicos e híbridos al modelo matemático estudiado con la finalidad de establecer comparativas de mayor alcance.

Adaptar al modelo nuevas consideraciones como: adaptar flujos de procesos, eliminar supuesto de precedencia de tareas, implementar tiempos de espera dentro de los procesos o la capacidad de tener inventario en proceso.

Generar mejoras sobre el algoritmo genético con nuevos operadores o probando distintos métodos de selección, cruce y mutación para esclarecer que combinación se adapta mejor al modelo.

Construir nuevo algoritmo genético con la generación de otra población inicial afectada por otras variables de solución o implementar una función fitness nueva y diferente que permita afectar de manera más significativa los individuos aptos y evitar la convergencia hacia óptimos locales.

Buscar otras formas para generar soluciones iniciales, también se recomienda, intentar adaptar la forma en la que los autores del modelo generan soluciones iniciales al GA.

**Referencias bibliográficas**

- Aalaei , A., Kayvanfar , V., & Davoudpour , H. (2019). Integrating multi-dynamic virtual cellular manufacturing systems into multi-market allocation and production planning. *SAGE journals*. Obtenido de <https://bibliotecavirtual.uis.edu.co:3422/doi/10.1177/0954405417731465>
- Abdullah, R., Abdul Rahman, N., & Salleh, R. (2019). A systematic approach to model human system in cellular manufacturing. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*. Obtenido de [https://www.jstage.jst.go.jp/article/jamdsm/13/1/13\\_2019jamdsm0001/\\_pdf/-char/en](https://www.jstage.jst.go.jp/article/jamdsm/13/1/13_2019jamdsm0001/_pdf/-char/en)
- Alduaij , A., & Hassan, N. (2020). Adopting a circular open-field layout in designing flexible manufacturing systems. *International Journal of Computer Integrated Manufacturing*. Obtenido de <https://bibliotecavirtual.uis.edu.co:3681/doi/epub/10.1080/0951192X.2020.1775300?needAccess=true>
- Alimiana, M., Ghezavati, V., & Tavakkoli-Moghaddam, R. (2020). New integration of preventive maintenance and production planning with cell formation and group scheduling for dynamic cellular manufacturing systems. *Journal of Manufacturing Systems*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S0278612520301011?via%3Dihub>
- Almasarwah, N., & Süer, G. (2021). Consideration of processing time dissimilarity in batch-cyclic scheduling of flowshop cells. *International Journal of Production Research*. Obtenido de <https://bibliotecavirtual.uis.edu.co:3681/doi/epub/10.1080/00207543.2020.1818863?needAccess=true>
- Barrientos, A., & Gambao, E. (2014). *Sistemas de producción automatizados*. Madrid, España: Dextra. Obtenido de [https://uids-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=TN\\_cdi\\_proquest\\_ebookcentral\\_EBC3229680&context=PC&vid=UIDS&lang=es\\_CO&search\\_scope=uids\\_cdi&adaptor=primo\\_central\\_multiple\\_fe&tab=uids\\_tab&query=any,contains,sistemas%20de%20prod](https://uids-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=TN_cdi_proquest_ebookcentral_EBC3229680&context=PC&vid=UIDS&lang=es_CO&search_scope=uids_cdi&adaptor=primo_central_multiple_fe&tab=uids_tab&query=any,contains,sistemas%20de%20prod)
- Behnia, B., Shirazi, B., Mahdavi, I., & Mahdi Paydar, M. (2021). Nested Bi-level metaheuristic algorithms for cellular manufacturing systems considering workers' interest. *RAIRO-Oper*. Obtenido de <https://www.rairo-ro.org/articles/ro/pdf/2021/01/ro180083.pdf>

- Behniaa, Mahdavia, Shirazi, & Paydar. (2019). A bi-objective bi-level mathematical model for cellular manufacturing system applying evolutionary algorithms. *Scientia Iranica*. Obtenido de [http://scientiairanica.sharif.edu/article\\_20563\\_deca3a456f1c7e8ce2e115e5ba7c30f4.pdf](http://scientiairanica.sharif.edu/article_20563_deca3a456f1c7e8ce2e115e5ba7c30f4.pdf)
- Bouaziz, H., Berghida, M., & Lemouari, A. (2020). Solving the generalized cubic cell formation problem using discrete flower pollination algorithm. *Expert Systems With Applications*. Obtenido de <https://pdf.sciencedirectassets.com/271506/1-s2.0-S0957417420X00042/1-s2.0-S0957417420301706/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEBgaCXVzLWVhc3QtMSJGMEQCICKP%2BZ8iKI2gTszls wVOb1RB1wpKxk01GpsP4z5Ck2C6AiBcLo9c2%2FksY%2BGYO9W2XXFQ7UFZH Z6l2%2FENsJvJ>
- Cervantes Gómez, L. (2015). Modelización matemática principios y aplicaciones. Obtenido de <https://www.fcfm.buap.mx/assets/docs/publicaciones/Modeliza.pdf>
- Chu, X., Gao, D., Cheng, S., Wu, L., Chen, J., Shi, Y., & Qin, Q. (2019). Worker assignment with learning-forgetting effect in cellular manufacturing system using adaptive memetic differential search algorithm. *Computers & Industrial Engineering*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S036083521930419X?via%3Dihub>
- Ciatera Díaz, D. (Septiembre de 2015). MODELO DE OPTIMIZACIÓN MULTIOBJETIVO PARA EVALUACION DE EFICIENCIA EN UNA EMPRESA DE SERVICIOS ELÉCTRICOS. Obtenido de <http://repositoriodigital.ucsc.cl/bitstream/handle/25022009/133/David%20Ciatera%20D%20C3%ADaz.pdf?sequence=1&isAllowed=y>
- Cordovés García , A., Sanzano Tamayo, J., Lastre Aleaga, A., & Ávila Rondón, R. (18 de Julio de 2016). Procedimiento para la fabricación de elementos de máquinas mediante tecnología de grupo en la pequeña y mediana empresa. *INGENIARE*. Obtenido de <https://www.redalyc.org/pdf/772/77252418008.pdf>
- Cruz Chávez, M. A., Moreno Bernal, P., & Perlata Abarca, J. (2014). Aplicación de la teoría de la complejidad.
- Deep, K. (2020). Machine cell formation for dynamic part population considering part operation trade-off and worker assignment using simulated annealing-based genetic algorithm.

- INDERSCIENCE*. Obtenido de <https://www.inderscienceonline.com/doi/abs/10.1504/EJIE.2020.105720>
- Duarte, A., Pantrigo, J., & Gallego, M. (2007). *Metaheurísticas*. Madrid. Obtenido de <https://bibliotecavirtual.uis.edu.co:3806/visor/48608>
- Escobar, L., Garavito, E., & Talero, L. (2017). Estudio del problema de formación de celdas de manufactura flexibles para el período comprendido entre 2007-2017 mediante un análisis bibliométrico. *ResearchGate*. Obtenido de [https://www.researchgate.net/publication/321255928\\_Estudio\\_del\\_problema\\_de\\_formacion\\_de\\_celdas\\_de\\_manufactura\\_flexibles\\_para\\_el\\_periodo\\_comprendido\\_entre\\_2007-2017\\_mediante\\_un\\_analisis\\_Bibliometrico](https://www.researchgate.net/publication/321255928_Estudio_del_problema_de_formacion_de_celdas_de_manufactura_flexibles_para_el_periodo_comprendido_entre_2007-2017_mediante_un_analisis_Bibliometrico)
- Estrada Lorenzo, J., Medino Muñoz, J., Rebollo Rodríguez, M., Campos Asensio, C., & Primo Peña, E. (2021). CONSECUENCIAS EN LA PRODUCCIÓN CIENTÍFICA DE LA COVID-19. *Rev Esp Salud Pública*, 5. Obtenido de [https://www.mschs.gob.es/biblioPublic/publicaciones/recursos\\_propios/resp/revista\\_cdrom/VOL95/C\\_ESPECIALES/RS95C\\_202105072.pdf](https://www.mschs.gob.es/biblioPublic/publicaciones/recursos_propios/resp/revista_cdrom/VOL95/C_ESPECIALES/RS95C_202105072.pdf)
- Feng, H., Xia, T., Da, W., Xi, L., & Pan, E. (2019). Concurrent design of cell formation and scheduling with consideration of duplicate machines and alternative process routings. *Springer*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2142/content/pdf/10.1007/s10845-016-1245-7.pdf>
- Ghassemi Tari, F., & Ahadi, K. (2019). Cellular layout design using Tabu search, a case study. *RAIRO-Oper*. Obtenido de <https://www.rairo-ro.org/articles/ro/abs/2019/05/ro170319/ro170319.html>
- Golmohammadi, A.-M., Honarvar, M., Tian, G., & Hosseini-Nasab, H. (2019). A new mathematical model for integration of cell formation with machine layout and cell layout by considering alternative process routing reliability: A novel hybrid metaheuristic. *International Journal of Industrial Engineering & Production Research*. Obtenido de <http://ijiepr.iust.ac.ir/article-1-911-en.pdf>
- Hong, Z., Zeng, Z., & Gao, L. (2021). Energy-efficiency scheduling of multi-cell manufacturing system considering total handling distance and eligibility constraints. *Computers & Industrial Engineering*. Obtenido de <https://pdf.sciencedirectassets.com/271420/1-s2.0-S0360835220X00131/1-s2.0-S0360835220306689/main.pdf?X-Amz-Security->

Token=IQoJb3JpZ2luX2VjEC8aCXVzLWVhc3QtMSJHMEUCIHRvtBNGbIV%2FO9h  
YmP%2FPjbxvVYOa4Hc5eZpZdYBL6wgmAiEA5S1BzLzEiTk9HxrYJx3QdjaDSduaF  
geWwYPQKwja

- Hosseinabad, E., & Zaman, M. (2020). A Brief Review on Cellular Manufacturing and Group Technology. *RsearchGate*. Obtenido de [https://www.researchgate.net/profile/Emad-Rabiei-Hosseinabad/publication/338544448\\_A\\_Brief\\_Review\\_on\\_Cellular\\_Manufacturing\\_and\\_Group\\_Technology/links/5e223fd492851cafc38c76b2/A-Brief-Review-on-Cellular-Manufacturing-and-Group-Technology.pdf](https://www.researchgate.net/profile/Emad-Rabiei-Hosseinabad/publication/338544448_A_Brief_Review_on_Cellular_Manufacturing_and_Group_Technology/links/5e223fd492851cafc38c76b2/A-Brief-Review-on-Cellular-Manufacturing-and-Group-Technology.pdf)
- Ibrahim, S., & Jarboui, B. (2020). A General Variable Neighborhood Search approach based on a p-median model for cellular manufacturing problems. *Springer*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2142/content/pdf/10.1007/s11590-020-01662-4.pdf>
- J. P. (2019). Modelado mediante Optimización Combinatoria. Obtenido de <http://personales.upv.es/jpgarcia/LinkedDocuments/MCOIOptimizacionCombinatoria.pdf>
- Jufeng Wang, Chunfeng Liu , & Kai Li. (2019). A hybrid simulated annealing for scheduling in dual-resource cellular manufacturing system considering worker movement. *Automatika*. Obtenido de <https://bibliotecavirtual.uis.edu.co:3681/doi/full/10.1080/00051144.2019.1603264>
- Khalid, Q., Arshad, M., Maqsood, S., Jahanzaib, M., Babar, A., Khan, I., . . . Kim, S. (2019). Hybrid particle swarm algorithm for products' scheduling problem in cellular manufacturing system. *Symmetry*. Obtenido de <https://www.mdpi.com/2073-8994/11/6/729/htm>
- Khalid, Q., Azim, S., Abas, M., Babar, A., & Ahmad, I. (2021). Modified particle swarm algorithm for scheduling agricultural products. *Engineering Science and Technology an International Journal*. Obtenido de <https://pdf.sciencedirectassets.com/305941/1-s2.0-S2215098621X00031/1-s2.0-S2215098620342804/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEC4aCXVzLWVhc3QtMSJIMEYCIQD5HyNJm5T5fABSsDidTRJsbpiFP0C20WEpNN9dyULc6QIhAN3VbGensSpeRtTfJuSBviWPHDf1r6eUPaBksEUktkiv>

- Lamba, K., Kumar, R., Mishra, S., & Rajput, S. (2020). Sustainable dynamic cellular facility layout: a solution approach using simulated annealing-based meta-heuristic. *Springer*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2142/content/pdf/10.1007/s10479-019-03340-w.pdf>
- Lashgari, M., Kia, R., & Jolai, F. (2021). Robust optimisation to design a dynamic cellular manufacturing system integrating group layout and workers' assignment. Obtenido de <https://bibliotecavirtual.uis.edu.co:4066/wos/woscc/full-record/WOS:000655131300002>
- Mansour, H., Afefy, I., & Taha, S. (2021). Heuristic-based approach to solve layout design and workers' assignment problem in the cellular manufacturing system. *International Journal of Management Science and Engineering Management*. Obtenido de <https://bibliotecavirtual.uis.edu.co:3681/doi/epub/10.1080/17509653.2021.1986682?needAccess=true>
- Mehdizadeh, E., Shamuradfar, M., & Akhavan Niaki, S. (2020). An integrated mathematical programming model for a dynamic cellular manufacturing system with limited resources. *INDERSCIENCE*. Obtenido de <https://www.inderscienceonline.com/doi/pdf/10.1504/IJSOM.2020.109437>
- Mohtashami, A., Alinezhad, A., & Niknamfar, A. (2020). A fuzzy multi-objective model for a cellular manufacturing system with layout designing in a dynamic condition. *INDERSCIENCE*. Obtenido de <https://www.inderscienceonline.com/doi/abs/10.1504/IJISE.2020.106086>
- Nogueira River, D., Hernández Nariño, A., Oviedo-Rodríguez, M., & Medina Nogueira, D. (2014). Programación de la producción. *ResearchGate*. Obtenido de <file:///C:/Users/sebas/Downloads/LibrookProgramaciondelaproduccion.pdf>
- Pournaderi, N., Ghezavati, V., & Mozafari, M. (2019). Developing a mathematical model for the dynamic facility layout problem considering material handling system and optimizing it using cloud theory-based simulated annealing algorithm. *SN Applied Sciences*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2142/article/10.1007%2Fs42452-019-0865-x>
- Rabbani, M., Farrokhi-Asl, H., & Ravanbakhsh, M. (2019). Dynamic cellular manufacturing system considering machine failure and workload balance. *Springer*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2142/content/pdf/10.1007/s40092-018-0261-y.pdf>

- Rahimi, V., Arkat, J., & Farughi, H. (2020). A vibration damping optimization algorithm for the integrated problem of cell formation, cellular scheduling, and intercellular layout. *Computers & Industrial Engineering*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S036083522030173X?via%3Dihub>
- Raoofpanah, H., Ghezavati, V., & Tavakkoli-Moghaddam. (2019). Solving a new robust green cellular manufacturing problem with environmental issues under uncertainty using Benders decomposition. *Engineering Optimization*. Obtenido de <https://bibliotecavirtual.uis.edu.co:3681/doi/full/10.1080/0305215X.2018.1517258>
- Salimpour, S., Pourvaziri, H., & Azab, A. (2021). Semi-robust layout design for cellular manufacturing in a dynamic environment. *Computers and Operations Research*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S0305054821001428?via%3Dihub>
- Salman Khalid, Q., Azim, S., Abas, M., Rehman Babar, A., & Ahmad, I. (24 de Enero de 2021). Modified particle swarm algorithm for scheduling agricultural products. *ScienceDirect*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S2215098620342804?via%3Dihub>
- Shafiee-Gol, S., Kia, R., Kazemi, M., Tavakkoli-Moghaddam, R., & Darmian, S. (2021). A mathematical model to design dynamic cellular manufacturing systems in multiple plants with production planning and location–allocation decisions. *Springer*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2142/content/pdf/10.1007/s00500-020-05417-2.pdf>
- Shafiee-Gol, S., Kia, R., Tavakkoli-Moghaddam, R., Kazemi, M., & Kamran, M. (2021). Integration of parts scheduling, MRP, production planning and generalized fixed-charge transportation planning in the design of a dynamic cellular manufacturing system. *RAIRO Operations Research*. Obtenido de <https://www.rairo-ro.org/articles/ro/pdf/2021/01/ro170409.pdf>
- Shunmugasundaram, Anbumalar, Anand, Sivakumary, & Nagarajan. (2019). Design of cellular manufacturing system for power press industry to reduce total travelling time by hybrid

- algorithm. *INDERSCIENCE*. Obtenido de <https://www.inderscienceonline.com/doi/pdf/10.1504/IJSOM.2019.103056>
- Singamsetty, P., & Thenepalle, J. K. (2021). *An efficient genetic algorithm for solving open multiple travelling salesman problem with load balancing constraint*. *Decision Science Letters*.
- Sivanandam, S., & Deepa, S. (2008). *Introduction to Genetic Algorithms*. New York: Springer. Obtenido de <https://bibliotecavirtual.uis.edu.co:2142/content/pdf/10.1007%2F978-3-540-73190-0.pdf>
- Venkata Deepthi, Ramakotaiah, & Krishnaveni. (2019). Research on performance of multi-skilled workers for sustainable production planning in seru production systems. *International Journal of Innovative Technology and Exploring Engineering*. Obtenido de <https://www.ijitee.org/wp-content/uploads/papers/v8i10/J91720881019.pdf>
- Wang, J., Liu, C., & Li, K. (2019). A hybrid simulated annealing for scheduling in dual-resource cellular manufacturing system considering worker movement. *Automatika*. Obtenido de <https://bibliotecavirtual.uis.edu.co:3681/doi/pdf/10.1080/00051144.2019.1603264?needAccess=true>
- Wang, J., Liu, C., & Zhou, M. (2020). Improved Bacterial Foraging Algorithm for Cell Formation and Product Scheduling Considering Learning and Forgetting Factors in Cellular Manufacturing Systems. *IEEE SYSTEMS JOURNAL*. Obtenido de <https://bibliotecavirtual.uis.edu.co:2272/document/8984281>
- Xiong, X., Lu, G., & Lu, D. (Diciembre de 2021). Research on Children's Customized Furniture Design Based on Group Technology. *MDPI*. Obtenido de <https://www.mdpi.com/2076-3417/11/23/11371/htm>
- Xue, G., & Offodile, F. (2020). Integrated optimization of dynamic cell formation and hierarchical production planning problems. *Computers & Industrial Engineering*. Obtenido de <https://pdf.sciencedirectassets.com/271420/1-s2.0-S0360835219X00121/1-s2.0-S0360835219306242/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEcaCXVzLWVhc3QtMSJGMEQCIGMOdVbvLWqcn0zIF4at30%2Fr70JTHLLJoRbTpdSLLnTuAiA3h5M5NF1c6yS4dI6kRWnuJSjRgtWyo%2FAk6EMYuuLA>

- Yu, J., Wang, S., & Xi, L. (2008). *Evolving artificial neural networks using an improved PSO and DPSO*. Obtenido de Science Direct: <https://www-sciencedirect-com.bibliotecavirtual.uis.edu.co/science/article/pii/S0925231207003591?via%3Dihub>
- Zandieh, M. (2019). Scheduling of Virtual Cellular Manufacturing Systems: A Biogeography-Based Optimization Algorithm. *Applied Artificial Intelligence*. Obtenido de <https://bibliotecavirtual.uis.edu.co:3681/doi/epub/10.1080/08839514.2019.1577021?needAccess=true>
- Zuluaga, C. A. (2014). *Planeación de la Producción*. Medellín: Universidad: EAFIT. Obtenido de <https://bibliotecavirtual.uis.edu.co:3806/a/67714>