



# INSTITUTO POLITÉCNICO NACIONAL

---

UNIDAD PROFESIONAL INTERDISCIPLINARIA  
DE INGENIERÍA Y CIENCIAS SOCIALES Y  
ADMINISTRATIVAS

SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

## ESTUDIO SOBRE LA TECNOLOGÍA DE GRUPOS Y SU INTEGRACIÓN EN LA MANUFACTURA INTEGRADA POR COMPUTADORA

T E S I S

QUE PARA OBTENER EL GRADO DE  
MAESTRO EN CIENCIAS  
EN INGENIERÍA INDUSTRIAL  
PRESENTA :

MARIANA CONTRERAS MÉNDEZ

DIRECTORES:

M. EN C. MARIO AGUILAR FERNÁNDEZ

DR. JUVENAL MENDOZA VALENCIA



MÉXICO, D.F.

2011

# Índice

---

<b>LISTA DE FIGURAS .....</b>	<b>3</b>
<b>LISTA DE TABLAS .....</b>	<b>3</b>
<b>LISTA DE GRÁFICAS .....</b>	<b>3</b>
<b>LISTA DE ECUACIONES.....</b>	<b>3</b>
<b>RESUMEN .....</b>	<b>4</b>
<b>ABSTRACT.....</b>	<b>5</b>
<b>PREFACIO.....</b>	<b>6</b>
<b>CAPÍTULO 1 INTRODUCCIÓN.....</b>	<b>9</b>
<b>1.1. CONTEXTO</b>	<b>9</b>
1.1.1. SISTEMAS INTEGRADOS DE MANUFACTURA .....	9
1.1.2. CONCEPTO BÁSICO DE TECNOLOGÍA DE GRUPOS.....	9
1.1.3. EL ROL DE LA TECNOLOGÍA DE GRUPOS EN LA MANUFACTURA CELULAR Y EN LA INTEGRACIÓN CAD/CAM.....	10
1.1.4. BENEFICIOS DE LA TG.....	11
<b>1.2. ANÁLISIS DE LA LITERATURA Y APORTACIONES PREVIAS</b>	<b>13</b>
1.2.1. HISTORIA Y ANTECEDENTES DE LA TECNOLOGÍA DE GRUPOS .....	13
Inspección Visual .....	14
Sistemas de Clasificación y Codificación.....	14
Análisis del Flujo de Producción .....	15
1.2.2. FORMULACIÓN GENERAL AL PROBLEMA DE TECNOLOGÍA DE GRUPOS.....	17
Método de los coeficientes similares .....	19
Principales vertientes actuales en la clasificación de los métodos de solución.....	19
1.2.3. MÉTODOS INSPIRADOS EN LA NATURALEZA.....	20
Redes Neuronales.....	21
Algoritmo hormiga.....	22
Algoritmo Genético .....	23
Recocido Simulado (Simulated Annealing).....	24
Búsqueda Tabú .....	25
Modelos difusos .....	26
<b>1.3. NECESIDADES DE MÁS INVESTIGACIÓN, JUSTIFICACIÓN Y OBJETIVO</b>	<b>28</b>

**CAPÍTULO 2 MÉTODO.....30**

**2.1. DESCRIPCIÓN DE METODOLOGÍAS DE ALGORITMOS GENÉTICOS APLICADAS A PROBLEMAS DE TECNOLOGÍA DE GRUPOS 30**

2.1.1.	ALGORITMO EVOLUTIVO PARA LA FORMACIÓN DE CELDAS DE MANUFACTURA (GONZALVES, RESENDE).....	31
2.1.2.	ALGORITMO GENÉTICO MEJORADO PARA SOLUCIONAR PROBLEMAS DE FORMACIÓN DE CELDAS (ENGGA) .....	35
2.1.3.	ALGORITMO GENÉTICO PARA EL DISEÑO DE CELDAS DE MANUFACTURA (MURUGAN, SELLADURAI) .....	39
2.2.	EVALUACIÓN DE LAS METODOLOGÍAS .....	41
2.3.	ÁREAS DE OPORTUNIDAD DERIVADAS DE LA EVALUACIÓN .....	44

**CAPÍTULO 3 RESULTADOS.....45**

**3.1. IMPULSORES 45**

3.2.	DESCRIPCIÓN AGREGADA DE LA PROPUESTA .....	45
3.3.	DESCRIPCIÓN DESAGREGADA DEL ALGORITMO GENÉTICO POR FACTORES DE DECISIÓN.....	48
3.4.	VERIFICACIÓN DEL MODELO.....	62

**CAPÍTULO 4 DISCUSIÓN .....66**

4.1.	MECANISMOS INTERNOS .....	66
4.2.	ANÁLISIS EXTERNO .....	68
4.3.	APORTACIONES.....	72
4.4.	FUTURAS INVESTIGACIONES DERIVADAS DEL ESTUDIO .....	73

**REFERENCIAS .....77**

# Lista de figuras

---

Figura 1 Matriz Inicial .....	16
Figura 2 Matriz Solución .....	16
Figura 3 Matriz que representa el problema de Formación de celdas de manufactura .....	18
Figura 4 Representación de las celdas de manufactura por bloques diagonalizados en la matriz de incidencia .....	18
Figura 1 Matriz de Boctor (1991) $7 \times 11$ .....	31
Figura 1 Pseudocódigo de un Algoritmo Genético Simple .....	47
Figura 2 Estructura General del Algoritmo Genético por Factores de Decisión (AGFD). .....	48
Figura 3 Representación Gráfica del Método de la Ruleta .....	51
Figura 4 Esquema de Selección y Reproducción.....	52
Figura 5 Matriz Inicial, Chan y Milner (1982) $10 \times 15$ .....	62
Figura 6 Matriz solución, Chan y Milner (1982) $10 \times 15$ .....	63

# Lista de tablas

---

Tabla 1 Análisis comparativo de las metodologías.....	43
Tabla 1 Resultados de Eficiencias y Tiempos obtenidos mediante el AGFD .....	65
Tabla 1 Comparación de resultados obtenidos por el algoritmo propio y dos enfoque más .....	69
Tabla 2 Valores utilizados en los Algoritmos Genéticos .....	70

# Lista de gráficas

---

Gráfico 1 Comparación de los resultados entre las metodologías estudiadas y el AGFD, elaboración propia con base en .....	70
---	----

# Lista de ecuaciones

---

Ecuación 1 Función de Pertenencia para un modelo difuso (21) .....	27
--	----

# Resumen

---

La idea fundamental de la Tecnología de Grupos desde el punto de vista de la manufactura es crear celdas de producción en donde cada celda se especialice en producir una familia de piezas, esto se logra agrupando las piezas en base a la cantidad de operaciones y de máquinas similares que tienen en común. El gran problema para cambiar de un diseño de planta tradicional a uno en el que se aplique la Tecnología de Grupos consiste en solucionar el problema de agrupar las piezas en familias.

En este trabajo se construye un algoritmo genético que soluciona el problema de formación de celdas de manufactura. Se propone dentro del algoritmo un esquema de asignación de piezas a las celdas de máquinas por medio de factores de decisión que cuantifican la cantidad posible de eficiencia aportada por cada pieza para así determinar su asignación por medio de la mayor eficiencia posible alcanzada. Otras de las características que hacen diferente este algoritmo a los demás es que presenta los mejores rasgos que fueron tomados de otras metodologías como codificación cromosómica y la eliminación de heurísticos que se tenían que ejecutar por cada individuo. Se seleccionaron 28 problemas de la literatura y fueron resueltos por medio de esta metodología y sus resultados comparados con enfoques afines.

# Abstract

---

The main idea of Group Technology within manufacturing point of view is the creation of production cells in which each cells specialize itself in producing just one family of parts. This is achieved through clustering parts and machines based on the quantity of similar operations.

The main problem of switching between a traditional layout to a Group Technology-based layout is solve the problem of grouping parts into families.

This research builds up a genetic algorithm to solve the cell formation problem. It proposes a part allocation scheme that assign parts to the machines cells through decision factors that quantifies each part possible efficiency contribution. Another feature that distinguishes this proposed algorithm from others is that it gathers the best characteristics from other methodologies such as chromosomic codification and the removal of local heuristics through the decision factor contained in the algorithm. 28 problems were selected from the literature and solved through the proposed algorithm and its results are reported and compared with similar methods.

# Prefacio

---

La idea de trabajar con algoritmos genéticos surgió por un libro (2) que obtuve en el cual desarrollaban el tema de una manera muy sencilla y bastante alejada del concepto que tenía de que un algoritmo es algo que casi nadie puede idear y que son muy complicados en sus pasos y en su resolución. Programé, con los pocos conocimientos que obtuve de un curso o dos de introducción a la lógica de programación, uno de los ejemplos del libro y me di cuenta que en realidad son muy pocas las necesidades de conocimientos en algún lenguaje de programación que se necesitan y que a cambio se pueden solucionar problemas complejos aplicados a cualquier campo de la ciencia. Es por ello que me animé a generar un algoritmo genético propio que le diera solución al problema de formación de celdas.

La estructura del presente trabajo está basada en los lineamientos científicos internacionales establecidos en la norma ISO 7144-86(3). Asimismo el contenido se fundamenta en los lineamientos descritos en publicaciones internacionales (4, 5), los cuales asientan los convenios retóricos en los que se encuentra inmersa la investigación científica. Tanto la función como el contenido fueron concebidos con tal de respetar las formas del lenguaje y su estructura, puntos imperativos en la escritura del discurso científico.

No obstante, también se tomó en cuenta los lineamientos que marca esta institución para elaborar una tesis de grado asentados en el reglamento de Estudios de Posgrado del IPN (6), los cuales marcan ciertas directrices que regulan de forma general y esbozando ciertas mociones para regular todos los trabajos de posgrados desarrollados bajo esta institución

En este trabajo de investigación se reconoce generosamente la deuda con los investigadores que han precedido en el actual tema de tesis por lo que se documenta cuidadosamente cada fuente, de manera que las contribuciones anteriores reciban su crédito apropiado

La metodología que se desarrolló en este trabajo fue una propuesta propia y también tomó aspectos y características de otras metodologías que se describen en el capítulo 2. El capítulo 3 desarrolla detalladamente toda la metodología, por ello, cabe aclarar que en ella los puntos que estén referenciados a otros autores son los aspectos de esta propuesta que se basaron en ellos y los puntos que no cuenten con una referencia, conforma el trabajo propio.

Para sustentar la investigación científica se informa al lector que las fuentes de citas o alusiones tomadas de otros trabajos se encuentran referenciadas por medio del modelo de referencias bibliográficas Vancouver,

El modelo Vancouver inició en 1978, con un pequeño grupo de directores de revistas médicas generales, que se reunieron informalmente en Vancouver (Columbia Británica) para establecer las pautas relativas al formato de los manuscritos enviados a sus revistas. Este grupo llegó a ser conocido como el Grupo de Vancouver. Sus requisitos de uniformidad para los manuscritos, incluidos los formatos para las referencias bibliográficas desarrollados por la Biblioteca Nacional de Medicina de los Estados Unidos, fueron publicados por vez primera en 1979. El Grupo de Vancouver creció y evolucionó para convertirse en el Comité Internacional de Directores de Revistas Médicas (CIDRM), que se reúne anualmente y que, poco a poco, ha ido ampliando los temas estudiados.

Todas las referencias bibliográficas se manejaron con la ayuda del software Endnote X1 y la edición del texto se realizó en Word 2007 parte de la suite de software en Microsoft Office 2007. Todos los títulos principales y subtítulos se encuentran en negrillas, la tipografía en cursiva denota términos en otros idiomas diferentes al español, excepto en los subtítulos en cuarto y quinto nivel que también se encuentran en cursiva.

La estrategia de búsqueda consistió en rastrear artículos en bases de datos académicas a las cuales tenemos acceso, entre las principales se encuentran EBSCO, ELSEVIER, Springer, Scielo y IEEE. En el caso de las revistas usadas las fuentes fueron variadas pero se aprecia que hubo una prevalencia en tres revistas con reconocimiento internacional como son: International Journal of Production Research, Production Planning and Control y Computers & Industrial Engineering.

Por su contribución, los autores cuyo trabajo destaco más en esta investigación fueron Gonçalves, Tunnukij y Murugan; su trabajo ha sido fundamental para ayudarme a desarrollar un algoritmo genético con las características que yo deseaba y para ubicarme dentro del campo de la inteligencia artificial.

En el capítulo 1 se construye un panorama acerca de la Tecnología de Grupos, su ubicación dentro de la manufactura y sistemas de producción, así como sus beneficios y características principales y se plantea el problema general de la Tecnología de Grupos. Esta proposición

constituye el parteaguas de todos los métodos que se han desarrollado hasta la fecha para solucionarlo y principalmente es la razón de ser de la presente investigación.

El capítulo 2 es una exhaustiva revisión de 3 propuestas que me parecieron atacan el problema desde la misma perspectiva que yo quiero abordar. Se determinan sus diferencias y similitudes para poder vislumbrar las posibles áreas de oportunidad que desarrollé en el capítulo 3. En este capítulo se encuentra mi propuesta detallada a profundidad y con todas las condiciones y requisitos para poder replicarla bajo cualquier otro ambiente.

El capítulo 4 contiene las conclusiones a las que se llegaron al evaluar los resultados obtenidos por esta metodología. También se compararon con los de otros 2 enfoques y se pudieron deducir cuestiones interesantes acerca del funcionamiento general de los algoritmos genéticos.

Finalmente quiero agradecer al Instituto Politécnico Nacional en especial a la Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas por su apoyo en la obtención de la Beca Institucional durante el semestre Julio-Diciembre y al CONACyT porque también pude obtener un apoyo económico bajo su programa de becas de posgrado.

# Capítulo 1 Introducción

---

En este capítulo se muestra un panorama general de la tendencia actual en métodos para resolver problemas de Tecnología de Grupos. Asimismo se retratan los surgimientos de la Tecnología de Grupos dentro del campo de la manufactura y su evolución hacia la aplicación en esta área de los métodos actuales.

## 1.1. Contexto

Dentro del diseño de las configuraciones de los medios de producción de las empresas también llamados sistemas de manufactura o proceso de producción, se encuentra una configuración denominada distribución celular o por grupos. El enfoque que se utiliza para ayudar a la organización a distribuirse celularmente es denominada Tecnología de Grupos. Comúnmente estas células se organizan en sistemas complejos que agrupan diferentes aspectos de la producción y son llamados Sistemas Integrados de Manufactura (7).

### 1.1.1. Sistemas Integrados de Manufactura

Un sistema de manufactura se puede definir como un conjunto integrado de recursos humanos y de equipo, cuya función es desarrollar uno o más procesos y/o operaciones de ensamble en materia prima, una pieza o un conjunto de piezas. El equipo de producción puede incluir máquinas y su herramental, dispositivos de manejo de materiales y posicionamiento del trabajo, y sistemas de cómputo (7).

Uno de los principales objetivos al implementar un Sistema de Manufactura Integrado por Computadora es mejorar la calidad, la flexibilidad e incrementar la productividad. Uno de los enfoques que se utilizan para alcanzar estos objetivos es organizar u agrupar las máquinas de producción dentro de celdas de manufacturas usando los conceptos de Tecnología de Grupos (8).

### 1.1.2. Concepto básico de Tecnología de Grupos

La tecnología de grupos es una filosofía de manufactura en la cual se identifican las piezas similares para tomar ventaja de sus características comunes de diseño y fabricación. Las piezas similares se agrupan en familias de piezas, asimismo se alcanza la eficiencia al acomodar el equipo de producción en celdas, en dónde cada celda se especializa en producir

una familia de piezas, esta asociación de máquina con este objetivo en común se llama Manufactura Celular (9).

### **1.1.3. El rol de la Tecnología de Grupos en la Manufactura Celular y en la Integración CAD/CAM**

El diseño de un Sistema de Manufactura Celular consiste en tres etapas principales:

1. Estudio de factibilidad Técnica y Económica
2. Diseño de un Sistema de Manufactura
3. Implementación del Sistema

Se puede diseñar un Sistema de Manufactura Celular aplicando los conceptos de Tecnología de Grupos y de Justo a Tiempo(10). Uno de los primeros problemas a resolver en la etapa del diseño del sistema es la formación de Células pieza-máquina. De acuerdo a las similitudes de las características de diseño o los requerimientos del proceso, las piezas se pueden agrupar en familias y las máquinas en celdas. A partir de este punto teóricamente las familias de piezas pueden ser procesadas en su correspondiente celda de maquinado(10).

Las tendencias actuales en manufactura también han contribuido a la adopción de la Tecnología de Grupos, estas tendencias se describen a continuación (11):

- Una rápida proliferación de diferentes variedades en los productos, lo cual repercute en tamaños de lote más pequeños.
- La creciente demanda para cumplir con tolerancias más pequeñas, creando la necesidad de contar con medios más económicos para trabajar con gran exactitud.
- La creciente necesidad de poder trabajar con un mayor número de materiales, reforzando la demanda de medios de producción más económicos.
- La disminución en la cantidad de desecho debido a una parte del costo total del producto es absorbido en mayor medida por la creciente eficiencia en el trabajo.
- La presión de todos los factores arriba expuestos para incrementar la comunicación a todo lo largo de las funciones de producción con el objetivo de minimizar los costos de producción y maximizar las tasas de producción.

Según las propias características de la Tecnología de Grupos, se vuelve evidente que es un elemento importante en los sistemas CAD/CAM. Un aspecto esencial de la integración del CAD con el CAM es la información que utiliza el departamento de Ingeniería, Manufactura y

los demás departamentos de una empresa. La Tecnología de Grupos provee un medio para estructurar y guardar la información, tales como los atributos de diseño y manufactura, procesos, y las capacidades de fabricación, las cuales son propicias para el análisis por computadora. La integración de las distintos tipos de información relacionada con el inventario de producción sería virtualmente imposible sin la Tecnología de Grupos, consecuentemente la Tecnología de Grupos es un elemento importante en la integración CAD/CAM (11).

#### **1.1.4. Beneficios de la TG**

Un beneficio importante de TG es que se mejora la calidad del producto. En primer lugar, los operadores en un célula producen un conjunto similar de piezas, por lo que se vuelven más hábiles en la manufactura de tales piezas. Como resultado de su experiencia, cometerán menos errores. La TG ofrece lotes más pequeños en las células de producción, por lo que cuando se cometen errores en un lote completo, se producirán menos piezas con defectos. Debido a la proximidad de las estaciones de trabajo en una célula de producción, los operadores están generalmente familiarizados con todas las operaciones en la celda; lo cual les permite darse cuenta de errores en etapas previas del proceso (11).

Muchos factores contribuyen a disminuir el costo unitario de producción: las celdas con TG requieren menores tiempos de preparación y disminuye el manejo de materiales en comparación con el trabajo tradicional por lote en el piso de producción. Aunado a esto, la menor cantidad de trabajo en proceso requiere menor financiamiento y reduce el inventario, lo cual libera capital de trabajo. La mejora en la calidad del producto también libera capital, cuando se detectan errores en las primeras etapas del proceso, habrá menor cantidad de unidades a desechar o re-trabajar. Finalmente todas las características de la TG ayudan a abatir costos indirectos al reducir los tiempos de diseño de nuevos productos y al mejorar el proceso de estimación de costos (11).

La capacidad de producción se ve incrementada debido principalmente a la disminución de los tiempos de preparación de máquinas. Durante el tiempo de preparación nada está siendo producido, así que cuando se alcanza una reducción en esta actividad, se puede utilizar el ahorro logrado para la producción adicional de piezas. Por ejemplo si el porcentaje promedio de tiempo de utilización de una máquina dedicado a la preparación de la misma disminuye de 10% al 5%, el 5% de tiempo ahorrado puede usarse para producir unidades adicionales (11).

Esta capacidad disponible se puede usar de diversas maneras. Sin ella, las empresas que operan a toda su capacidad se verían en la necesidad de rechazar órdenes adicionales de clientes actuales o limitar sus ventas a nuevos clientes, lo cual les llevaría a perder posibles ventas o a perder a sus mismos clientes. Las empresas que satisfacen demandas estacionales podrían ser capaces de incrementar su producción en los puntos más altos de demanda a través de disminuir el tiempo de preparación de máquina, en lugar de incrementar su capacidad por medio de maquinaria extra que permanecería ociosa durante el resto del año. Estando en esta posición las empresas pueden cumplir con sus objetivos de ventas a un costo total menor, por lo tanto incrementarían su ganancia total.

La TG también ayuda a mejorar los niveles de servicio al cliente, lo que conduce a posicionarse con una nueva o renovada ventaja competitiva. Los siguientes son 5 beneficios relacionados a los diferentes aspectos del servicio al cliente(11).

1. Disminución del tiempo de entrega. Cuando se forman celda de TG, los tiempos de preparación entre los diferentes tipos de piezas o trabajos se acortan en comparación con el trabajo típico en el taller o por lotes de trabajo.
2. Lotes de trabajo más pequeños lleva a tener menor cantidad de trabajo en proceso lo cual reduce la cantidad de material en espera a ser procesado por cada máquina.
3. Menores colas de espera del material conducen a la reducción del tiempo total del proceso, disminuyendo así los tiempos de entrega.
4. La reducción de los tiempos de entrega puede ayudar a las empresas a ser más responsable con sus clientes de diversas formas. Cuando los clientes tratan de minimizar sus niveles de inventario, la capacidad de poder cumplir con menores tiempos entrega puede ser la diferencia entre mantener a un cliente o perderlo. Cuando las condiciones del mercado cambian rápidamente, los clientes pueden demandar envíos de producto más frecuentes, cambios significativos en las especificaciones del producto, o exigencias similares que puedan alterar el proceso de manufactura. La reducción en los tiempos de entrega hace posible acomodar los cambios a petición del cliente en el calendario de entregas. Estos cambios a última hora se pueden lograr sin que el productor tenga la necesidad de apoyarse en un inventario mayor de materia prima o de piezas (incurriendo en mayores costos), debido a que se reduce el ciclo total de pedido (el tiempo entre que se recibe una orden y esta orden es enviada).

5. Promesas de entrega más confiables. Una parte importante de la responsabilidad con el cliente es el cumplimiento de sus expectativas. A través de tiempos de entrega más cortos, las celdas con TG ayudan a la empresa a estimar las fechas de entrega al mismo tiempo que puede cumplir con la fecha de entrega prometida. Cuando se reducen y se cumplen los tiempos de entrega se reduce la incertidumbre inherente a cualquier proceso, lo que se traduce en un aumento en la confiabilidad de los clientes hacia la empresa.

## **1.2. Análisis de la Literatura y Aportaciones Previas**

En esta sección se pretende mostrar las aportaciones con las que otros investigadores han contribuido a la expansión en este campo del conocimiento. Lo que se plantea en este apartado es un breve recorrido histórico sobre los métodos que se han empleado para resolver el problema de formulación de celdas de trabajo y la forma en cómo han ido cambiando estas técnicas hasta llegar a enfoques que combinan métodos algorítmicos de búsqueda y optimización.

### **1.2.1. Historia y Antecedentes de la Tecnología de Grupos**

El uso de la Tecnología de Grupos dentro de la Manufactura fue mencionado inicialmente en 1925 por R.F. Flanders, quien describió un concepto de manufactura similar al concepto de Tecnología de Grupos. Flanders sugirió el uso de la estandarización del producto, dividir en departamentos en base al producto en lugar del proceso, minimización del transporte y control visual del trabajo mismo en lugar de registros de control en lugares remotos(8).

Su concepto fue luego aplicado en Jones and Lamson Machine Company. J.C.Kerr en su informe de investigación presentado en 1938 al Instituto de Ingenieros de Producción, habla sobre un diseño de planta basado en reunir a las máquinas. A. P. Sokolovsky en la URSS sugiere el desarrollo de piezas con formas y características similares para producirlas en un proceso estandarizado. También explica su idea por medio de un método de clasificación basado en procesos (8).

Durante las últimas cinco décadas se han realizados serios intentos para mejorar la eficiencia de las operaciones de manufactura por lotes. Como resultado de esto, se han escrito muchos artículos al respecto, de diversos autores y enfocándose y distintos temas (12). Los escritores más notables sobre este tema son Burbidge (9), Ivanov (13), Mitrofanov (14) y Petrov (15).

El avance representativo de la TG se dio con la introducción de esta filosofía en diferentes talleres de Europa en la década de los sesenta (1960's). En los años setenta, algunas de plantas de Asia y de Estados Unidos también establecieron células de manufactura(12).

La aplicación específica de la TG en este estudio es la manufactura celular. La mayoría de las aplicaciones de manufactura celular involucran agrupar los diferentes componentes en familias de piezas usando como criterio de clasificación las operaciones de producción similares entre ellas; posteriormente se dividen las máquinas en grupos correspondientes a cada familia. Cada grupo contiene todas las máquinas necesarias para llevar a cabo el conjunto de operaciones que la familia de piezas requiere(12). Para lograrlo se disponen de diversos métodos, entre ellos existen algunos de los antiguos y más difundidos son los siguientes (9):

- Inspección visual.
- Sistemas de Clasificación y Codificación.
- Análisis del Flujo de Producción.

### **Inspección Visual**

En este sistema se van clasificando las piezas a partir del examen de los planos y según sus procesos de fabricación en clases, subclases, grupos, subgrupos, etc. Pueden utilizarse las dimensiones necesarias hasta la formación de familias con el grado de semejanza requerido(16).

La búsqueda por Inspección Visual no es usada frecuentemente en las aplicaciones de Tecnología de Grupos, debido a que puede llevar a resultados inconsistentes ya que rara vez dos personas identificarán a un grupo de piezas dentro de la misma familia. Hay muchas razones para esto, tales como que cada persona tiene una apreciación diferente de las capacidades de proceso en una fábrica y reconoce de forma diferente los atributos significativos de una pieza dada. Por lo tanto este no es un método factible ya que carece de exactitud y sofisticación (16).

### **Sistemas de Clasificación y Codificación**

El objetivo de la codificación y clasificación para la aplicación de la Tecnología de Grupos es explotar las características similares en el diseño de piezas. El sistema consiste en dos actividades. Primero a cada parte del subcomponente se le asigna un código basado en un sistema de clasificación que describe los diversos atributos de una pieza. Después se clasifican

las piezas en familias basándose en atributos similares, tales como forma geométrica, tipo de material, requerimientos de maquinaria o de tolerancias. Idealmente el sistema de codificación debería de incluir tanto los atributos de diseño como los de manufactura de la pieza(11)

Cuando todas las piezas han sido codificadas, el sistema de codificación puede ser muy útil de diferentes formas. Por ejemplo, si un proveedor ofrece un descuento sustancial en algún determinado material. El departamento de compras puede buscar en la base de datos todas las piezas que utilicen ese tipo de material y evaluar la programación de la entrega del producto para anticipar las necesidades en de manufactura. Los ingenieros de diseño también pueden hacer uso de esta base de datos. Se puede usar el código de nueva una pieza para buscar en la base de datos piezas similares, las cuales pueden eliminar la duplicación innecesaria de diseños. Aún si se necesita un nuevo diseño, sería más fácil modificar uno existente, que empezar desde cero (11).

A pesar de que el sistema de codificación clasificación basada en Tecnología de Grupos puede brindar muchos beneficios, su implementación conlleva costos y una significativa inversión de tiempo. Las empresas que se pueden beneficiar más con este sistema son aquellas que manejan una gran cantidad de piezas y componentes, o que frecuentemente necesitan diseñar piezas.

Al usar un sistema de codificación y clasificación las empresas pueden estimar con mayor precisión los costos para nuevos productos. La mejora en las estimaciones ayuda a reducir la incertidumbre en las decisiones sobre cuanto cargar a un producto o en cuanto ofrecer un trabajo. Este sistema hace posible la asignación de precios competitivos que también aseguren la generación de una ganancia (11).

### **Análisis del Flujo de Producción**

El primer paso para planear un sistema de manufactura es la selección de familias de piezas y grupos de máquinas, (formación de células) (17).Entre los diferentes enfoques usados en la selección de las familias y grupos se encuentra el **Análisis de Flujo de Producción** (Production Flow Analysis, (PFA)) desarrollado por Burbidge en 1975(9).

Este método, permite formar simultáneamente las familias de piezas y los grupos de máquinas en que deben ser mecanizadas estas familias. Las suposiciones subyacentes en el

enfoque del PFA es que: 1) la mayoría de las piezas y máquinas en una empresa ya están claramente definidas en familias y grupos; 2) estos grupos y familias se pueden identificar claramente a través de un análisis del ruteo de piezas y de una lista de equipos. Este método no requiere un sistema de codificación especial y es relativamente sencillo de implementar(17).

Se han desarrollado múltiples heurísticos para el proceso de formación de células. El principal objetivo de estos enfoques es la combinación de las filas y columnas de la matriz de incidencia pieza-máquina para formar una matriz resultante con los valores de incidencia formando una diagonal de bloques (17). No obstante, en algunos casos no todos los componentes de una familia de piezas pueden ser procesados dentro de la misma célula de trabajo (18).

		Parts								
Machines		A	B	C	D	E	F	G	H	I
1		1			1				1	
2						1				1
3				1		1				1
4			1				1			
5		1							1	
6				1						1
7			1				1	1		

Figura 1 Matriz Inicial (9).

		Parts								
Machines		C	E	I	A	D	H	F	G	B
3		1	1	1						
2			1	1						
6		1		1						
1					1	1	1			
5					1		1			
7								1	1	1
4								1		1

Figura 2 Matriz Solución (9).

A los métodos vinculados con el PFA se les asocia numerosos algoritmos; los de identificación de agrupamientos se basan en una ordenación matricial hasta obtener bloques diagonales que identifiquen las asociaciones pieza-máquina; habitualmente surgen problemas por las soluciones encontradas que dan origen a piezas cuello de botella que requieren de máquinas

que no es posible agrupar en una sola celda; si bien han surgido algoritmos que resuelven esta cuestión, se convierten en problemas NP(19).

### 1.2.2. Formulación general al problema de Tecnología de Grupos

La idea fundamental de la Tecnología de Grupos desde el punto de vista de la manufactura es la decomposición del sistema de manufactura en subsistemas, al agrupar las piezas en familias de piezas y las máquinas en celdas de maquinado, basándose en las similitudes de las características de maquinado de las diferentes piezas. La agrupación de piezas se basa en la cantidad de operaciones y de máquinas similares que tienen en común, estas máquinas se agrupan en una celda de trabajo y de esta forma se obtiene un subsistema o célula. Es por esto que el gran problema para cambiar de un diseño de planta tradicional a uno en el que se aplique la Tecnología de Grupos consiste en solucionar el problema de agrupar las piezas en familias(20).

La agrupación ideal para un sistema de manufactura dado será tal que las máquinas y las piezas puedan ser distribuidas en células mutuamente excluyentes de tamaños adecuados sin que existe flujo intercelular de piezas entre las diferentes celdas de maquinado(20).

La mayoría de los métodos para la solución del problema de Tecnología de Grupos ocupan las hojas de proceso de piezas para formar las celdas pieza-máquina. Estas son también útiles para construir la matriz de incidencia inicial para todas las piezas y todas las máquinas. En esta matriz se enumeran en columnas todas las piezas y las máquinas se enumeran en filas formando una retícula de coincidencia entre piezas y máquinas. Posteriormente se le asigna el valor unitario a todas las máquinas que la pieza tenga que visitar para ser completamente producida(10).

$$a_{ij} = \begin{cases} 1 & \text{si la pieza } j \text{ es procesada en la máquina } i \\ 0 & \text{de otra forma} \end{cases}$$

		Parts								
		A	B	C	D	E	F	G	H	I
Machines	1	1			1				1	
	2					1				1
	3			1		1				1
	4		1				1			
	5	1							1	
	6			1						1
	7		1				1	1		

Figura 3 Matriz que representa el problema de Formación de celdas de manufactura (9).

La agrupación de piezas y máquinas en familias es equivalente a reacomodar las filas y columnas de la matriz de incidencia de tal manera que la mayor cantidad posible de valores diferentes a cero queden agrupados en bloques diagonalizados(10).

		Parts									
		C	E	I	A	D	H	F	G	B	
Machines	3	1	1	1							
	2			1	1						
	6	1		1							
	1				1	1	1				
	5				1		1				
	7								1	1	1
	4								1		1

Figura 4 Representación de las celdas de manufactura por bloques diagonalizados en la matriz de incidencia (9).

Las pocas entidades que se encuentran fuera de la diagonal formada por bloques representan a las operaciones que se efectuarán fuera de sus celdas asignadas. Estas entidades se llaman Elementos Excepcionales. La máquina correspondiente a estos elementos es conocida como una máquina Cuello de Botella. Los valores nulos (ceros) dentro de los bloques diagonalizados se llaman vacíos(10).

En todos los algoritmos y formulaciones de Tecnología de Grupos, por las características intrínsecas de esta filosofía, se desalienta el movimiento intercelular de materiales(20).

La mayoría de los métodos y algoritmos actuales de solución al problema de Tecnología de Grupos utilizan esta formulación como el punto de partida para la solución de los algoritmos.

## **Método de los coeficientes similares**

El Método de los coeficientes similares (Similarity Coefficient Method (SCM)) es otra metodología que consiste en la agregación progresiva de máquinas en celdas basándose en sus coeficientes de similitud. McAuley propuso esta metodología por primera vez en 1972, y posteriormente fue mejorada por Waghodekar y Sahu en 1984 y por Seifodidini y Wolfe en 1986(17).

Este método involucra un proceso jerárquico de agrupamiento de máquinas de acuerdo a "coeficientes de similitud" que posteriormente se utilizarán en el cómputo de la solución. El coeficiente de similitud se usa para describir que tan "iguales" son dos máquinas en términos del número de piezas que cada máquina procesa. Este coeficiente posteriormente se utiliza para producir un diagrama de árbol para el juicio final sobre cómo formar los grupos de máquinas(17).

Dentro de los enfoques que utilizan el SCM se encuentran el método de Agrupamiento por Enlace sencillo (Single Linkage Clustering (SCM)) y el método de Agrupamiento por Enlace Promedio (Average Linkage Clustering (ALC)). Mientras que el SCM es un método simple y requiere mínimos esfuerzos computacionales, éste puede generar celdas de maquinado en el que un gran número de las máquinas incluidas pueden ser relativamente "desiguales". El AOL soluciona este problema, pero requiere de grandes recursos de cómputo; y aún, a pesar de la técnica la asignación de las máquinas a las celdas de trabajo se realiza únicamente en base a los coeficientes de similitud(17).

## **Principales vertientes actuales en la clasificación de los métodos de solución**

Los grupos pieza-máquina necesitan ser identificados de tal manera que las piezas con tamaño, características de diseño o requerimientos de función o de proceso similares sean todas producidas completamente en una célula que comparta recursos comunes como por ejemplo máquinas, herramientas y trabajo(21).

El problema de asociación de piezas y máquina en celdas independientes fue en un inicio analizado por Burbidge (9). Él sugería un método llamado Análisis del Flujo de Producción para identificar los diferentes grupos pieza-máquina. Debido a su enorme complejidad, este problema ha recibido gran atención y de esta forma se han desarrollado en la literatura sobre TG muchos otros procedimientos(21).

Una desventaja común en estos procedimientos es que la introducción de nuevas piezas en el sistema requiere que se vuelva a correr el algoritmo de solución con el nuevo conjunto de datos. Otra desventaja es que estos procedimientos también han asumido que una pieza dada puede pertenecer sólo a una familia de piezas (celda de trabajo)(21).

Tomando en cuenta los recientes desarrollos sobre los procedimientos para la solución del problema de asociación pieza-máquina, éstos se pueden clasificar basándose en el tipo de cómputo requerido: cómputo blando (soft-computing) contra cómputo duro (hard-computing)(21).

Las características más deseadas cuando se usa un tipo de cómputo duro son la precisión, la certidumbre y el rigor. En contraste, el punto de partida en el cómputo blando es la tesis acerca de que la precisión y la certidumbre no siempre son posibles, y por consiguiente la computación, el razonamiento y la toma de decisiones deben de explotar estas características cada vez que les sea posible, en otras palabras, se basa en la tolerancia por la imprecisión y en la incertidumbre. Esta idea de explotar la tolerancia por la imprecisión y la incertidumbre coincide con la habilidad humana para entender, aprender, reconocer y clasificar patrones de datos. En este momento los principales componentes de los procedimientos que utilizan cómputo blando son las Redes Neuronales, Modelos con Lógica Difusa, Algoritmos Genéticos y Razonamiento Probabilístico. Los enfoques basados en el cómputo duro tradicional incluyen Métodos de Agrupamiento basado en Similitudes y enfoques Heurísticos de programación matemática(21).

### **1.2.3. Métodos inspirados en la naturaleza**

Glover y Greenberg enfatizan el paralelismo entre la Inteligencia Artificial y las técnicas de búsqueda. Son herramientas apropiadas especialmente para los problemas de búsqueda combinatoria. Simulated annealing está inspirado en la metalurgia, redes neuronales artificiales, algoritmos genéticos, colonias artificiales de hormigas, estrategias de forrajeo en la biología, búsqueda tabú en la sociología, etc. Estas técnicas son usadas exitosamente para solucionar problemas en donde se buscan las mejores secuencias de ciertos objetos(22).

Ellos comparten muchas características entre sí y al mismo tiempo mantienen sus peculiaridades. Las soluciones apropiadas a los problemas como la secuenciación, planeación, corte y empaque, balanceo de línea de producción y agrupación se logran representándolas por medio de cadenas de símbolos(22).

El proceso de búsqueda aleatoria o parcialmente aleatoria genera una serie de soluciones alternativas desde el principio, al cambiar la secuencia de la variable de entrada o al generar un híbrido con ésta. Estos procesos generalmente buscan en la vecindad o en los puntos anteriores y ocasionalmente "saltan" a puntos distantes para evitar una solución final prematura. También pueden permitir soluciones de menor calidad por un lapso de tiempo mientras esperan encontrar soluciones mejores. Cuentan con una naturaleza distribuida y aleatoria; y la característica de mantener cierta información. Por último, estos procesos de búsqueda alcanzan las soluciones óptimas o cercanas a la óptima en un tiempo razonable(22).

Las técnicas de Inteligencia Artificial necesitan de un mecanismo para evaluar el rendimiento de cualquier solución factible. Si una solución posteriormente generada es mejor a la anterior, entonces se acepta. Las características de estos métodos es la habilidad para aceptar soluciones alternativas que sean peores que la actual. No obstante este resultado está condicionado y su permanencia es temporal. Se espera alcanzar una mejora continua a largo plazo. Por otro lado estos procedimientos no garantizan encontrar la mejor solución y depende de parámetros para generar y mantener soluciones factibles, lo cual es molesto y requiere de un balance entre mantener y destruir la solución actual(22).

## **Redes Neuronales**

Las Redes Neuronales Artificiales (Artificial Neural Network) son sistemas distribuidos de procesamiento de información que simulan el proceso biológico de aprendizaje. Estas básicamente consisten en diferentes componentes como la unidad de proceso (Processing Unit, PU), conexiones, función de propagación, función de activación/transferencia y regla de aprendizaje. Las unidades de proceso están densamente interconectadas a través de ligas o conexiones. Las PU toman uno o más valores de entrada, los combinan en un único valor usando la regla de propagación, luego lo transforman en un valor de salida a través de la función de activación/transferencia(21).

Se pueden construir redes complejas al interconectar un mayor número de Unidades de Proceso. La red más simple está formada por un grupo de PU's colocados en una sola capa. Las redes multicapa se forman al situar varias capas sencillas en cascada(21).

Una red neuronal aprende por medio de un conjunto de patrones de entrenamiento generalizando las características dentro de estos patrones. Después de una adecuada actividad de generalización, la red almacena esas características internamente en su

arquitectura. Después del entrenamiento, la red neuronal debe ser capaz de reconocer y clasificar aquellos patrones de entrada que nunca haya visto antes. Este aprendizaje se lleva a cabo principalmente a través del reajuste de los pesos de interconexión usando ciertos procesos de aprendizaje, por ejemplo, la función de aprendizaje delta, la función Hebbiana y la función de aprendizaje competitivo(21).

El aprendizaje supervisado requiere del acoplamiento de cada valor de entrada con un valor objetivo, el cual representa el valor de salida deseado y un "maestro", que provee la información concerniente al error. En el aprendizaje no supervisado, el conjunto de valores de entrenamiento consiste solamente en vectores de entrada. Los procedimientos del aprendizaje no supervisado construyen modelos internos que capturan las regularidades en sus valores de entrada sin recibir ninguna información adicional(21).

Se han desarrollado diferentes tipos de modelos de redes neuronales en la literatura concerniente a este tipo de redes. Estos modelos se caracterizan por su propiedades, como estructura de la red (topología), cómo computa y qué computa la red (propiedad computacional) y cómo la red aprende a procesar (propiedad de aprendizaje o entrenamiento).

El número de capas, el número de neuronas en cada capa y los pesos que se le asignan a las conexiones se pueden seleccionar de tal forma para que brinden el mejor ajuste al conjunto de datos.

El gran número de unidades de proceso interconectadas proveen a la red neuronal la habilidad para aprender patrones los complejos de la información dada y para generalizar la información aprendida. Su propia naturaleza tolerante a fallas, le otorga a la red la habilidad de trabajar con información incompleta. Su capacidad para otorgar resultados en tiempo real hacen de los modelos de redes neuronales una gran herramienta de apoyo para crear un ambiente inteligente de manufactura en la vida real y para solucionar el problema de agrupamiento pieza-máquina(21).

### **Algoritmo hormiga**

El sistema hormiga consiste en agentes cooperativos (hormigas artificiales) y un conjunto de reglas para determinar la generación, actualización y uso de la información local y global con el objetivo de encontrar buenas soluciones. Las colonias de hormigas exhiben una conducta

muy interesante. Inclusive con las simples capacidades de una sola hormiga, la conducta de toda la colonia está altamente estructurada (22).

Hormigas semiciegas realizan individualmente movimientos aleatorios en su vida diaria dejando un rastro de feromonas. Estos rastros afectan a las otras hormigas predisponiendo la aleatoriedad de sus decisiones para elegir un camino. El rastro de feromonas eventualmente se evapora, pero, mientras pasa el tiempo algunos de ellos se intensifican por el ir y venir de numerosas hormigas entre los puntos de vital importancia. Como resultado algunas de estas trayectoras se vuelven evidentes. Este fenómeno se simula con un sistema artificial de hormigas para develar las mejores soluciones(22).

### **Algoritmo Genético**

Los algoritmos genéticos fueron formalmente introducidos por John Holland en 1975 y se han aplicado en diversos campos como las matemáticas, la ingeniería, la biología y las ciencias sociales. El algoritmo genético es un algoritmo de búsqueda basado en la mecánica de la selección natural y la genética natural. Estos combinan el concepto de sobrevivencia del más adaptado con el intercambio estructurado de información algo aleatoria para formar algoritmos de búsqueda que sean robustos. El concepto de Algoritmo Genético imita el proceso de evolución que ocurre en la biología natural (23).

Este algoritmo empieza generando una población inicial de posibles soluciones, a las cuales se les refiere como individuos o cromosomas. La alberca genética de una población dada contiene una solución potencial, o una mejor solución a un problema adaptativo. Esta solución no está “activa” porque la combinación genética en la cual recae está dividida entre varios sujetos. Sólo la asociación de diferentes genomas puede llevar a la solución. Los cromosomas evolucionan a través de iteraciones sucesivas llamadas generaciones. Durante cada generación, los cromosomas evolucionan utilizando una medida de adecuación(23).

Para crear la siguiente generación, se utilizan operadores de cruce y mutación para formar nuevos cromosomas (descendencia). La nueva generación se forma al seleccionar los cromosomas más aptos. Los cromosomas con mejor adecuación a las condiciones imperantes tienen mayor probabilidad de ser seleccionados. Después de varias generaciones, el algoritmo alcanza el criterio de paro y converge al mejor cromosoma, el cual representa la mejor solución encontrada al problema(23).

## Recocido Simulado (Simulated Annealing)

El método del recocido simulado es un método de búsqueda local aleatorizada que ha sido utilizado para derivar soluciones cercanas al óptimo para problemas con una alta complejidad computacional. Este método fue originalmente desarrollado por como un modelo de simulación para un proceso físico de recocido y de allí se le nombró como algoritmo de Recocido Simulado(24).

El método del recocido se utiliza en la industria para obtener materiales más resistentes, o más cristalinos, en general, para mejorar las cualidades de un material (25).

El proceso consiste en “derretir” el material (calentarlo a muy alta temperatura). En esa situación, los átomos adquieren una distribución “azarosa” dentro de la estructura del material y la energía del sistema es máxima. Luego se hace descender la temperatura muy lentamente por etapas, dejando que en cada una de esas etapas los átomos queden en equilibrio (es decir, que los átomos alcancen una configuración óptima para esa temperatura). Al final del proceso, los átomos forman una estructura cristalina altamente regular, el material alcanza así una máxima resistencia y la energía del sistema es mínima (25).

Experimentalmente se comprueba que si la temperatura se hace descender bruscamente o no se espera suficiente tiempo en cada etapa, al final la estructura del material no es la optima(25).

Esencialmente este método difiere de los demás métodos de búsqueda local ya que permita aceptar una solución inferior en la vecindad general de la solución actual con una probabilidad positiva. De esta forma el algoritmo facilita la búsqueda de una superficie de respuesta que puede tener múltiples óptimos locales(24). En general para implementar un procedimiento basado en el algoritmo del recocido simulado para cualquier problema complejo de optimización se tiene que tomar muchas decisiones, estas son:

- ¿Qué es una solución?
- ¿Cuál es el costo de una solución?
- ¿Cómo determinar una solución inicial?
- ¿Cuál es la vecindad de una solución?

Adicionalmente el tiempo computacional requerido para el heurístico basado en el recocido simulado puede ser controlado por un número dado de iteraciones (T) y por el número de

búsquedas locales llevadas a cabo en cada iteración (S). Un parámetro final que necesita ser especificado al implementar este heurístico, es la probabilidad de aceptar una solución inferior en cada iteración. Típicamente el valor de esta probabilidad debe acercarse a 0 mientras se aproxima el límite de iteraciones(24).

### **Búsqueda Tabú**

La búsqueda tabú (TS) es un procedimiento heurístico de memoria adaptativa propuesto por Glover en 1986 para la búsqueda de los óptimos globales en problemas de optimización mono-objetivo(26).

TS explora el espacio de soluciones a través de repetidos movimientos desde una solución a la mejor de sus vecinas tratando de evitar los óptimos locales. Para un problema mono-objetivo, TS realiza una búsqueda por entornos en la cual se desplaza en cada iteración a la mejor solución no tabú del vecindario de la solución actual. Los principales atributos de cada solución visitada son almacenados en una lista tabú por un determinado número de iteraciones para evitar que estas soluciones sean revisitadas, es decir, para evitar ciclos en la búsqueda por entornos. Así, un elemento del vecindario de la solución actual es declarado tabú (es decir, es prohibido) si alguno de sus atributos está en la lista tabú. En general, un método basado en búsqueda tabú requiere de los siguientes elementos (26):

1. Solución inicial. La búsqueda debe comenzar desde una solución inicial que podría ser cualquier solución admisible que satisfaga las restricciones del problema. Una buena solución inicial podría acelerar la búsqueda con el consiguiente ahorro de tiempo. Dicha solución puede ser generada aleatoriamente o utilizando funciones ávidas o greedy functions (funciones que incorporan información adicional del problema utilizadas como estrategias para generar puntos de mejor calidad).
2. Movimiento. Un movimiento es un procedimiento aleatorio o determinístico por el que se genera una solución admisible a partir de la solución inicial. Usualmente, este procedimiento es sencillo para el caso de problemas combinatorios, pero mucho más complejo para el caso de problemas de optimización continuos.
3. Vecindad. Dada una solución S, la vecindad  $N(S)$  es el conjunto de todas las soluciones admisibles que pueden ser generadas por la ejecución de un movimiento sobre la solución actual S. Este conjunto suele ser numerables para problemas combinatorios y, en aquellos

casos en los que  $N(S)$  sea grande, se suele operar con un subconjunto de más creativo para definir  $N(S)$ .

4. Lista tabú. Es un mecanismo de memoria adaptativa que trata de evitar que la búsqueda entre en un ciclo o quede atrapada en un óptimo local. Una vez que un movimiento, que genera una nueva solución, es aceptado, su movimiento inverso se añade a la lista tabú y permanece en ésta un número determinado de iteraciones. Si el tamaño de la lista tabú es pequeño, entonces la búsqueda se intensifica en una determinada área del espacio, mientras que si el tamaño de la lista es grande se enfatiza la búsqueda en diferentes regiones del espacio de soluciones.

5. Criterio de parada. En general, la búsqueda termina después de un número determinado de iteraciones, después de un tiempo de computación predefinido o cuando se alcanza un número dado de iteraciones sin mejorar la mejor solución(26).

No obstante, dentro de un TS se puede encontrar una gran variedad de estrategias destinadas a mejorar la búsqueda como, por ejemplo, criterios de aspiración, que admite movimientos tabú si se satisface el criterio de aspiración (por ejemplo, se mejora el valor de la función objetivo de la mejor solución encontrada hasta ese momento) con idea de cruzar las barreras impuestas en las restricciones tratando de encontrar otras zonas factibles más prometedoras; fases de intensificación, que permite concentrar la búsqueda en aquellas zonas más prometedoras; fase de diversificación, que permite desplazarse hacia zonas no exploradas, etc.

## **Modelos difusos**

El término de modelos difusos se usa para hacer referencia a los conjuntos difusos o las matemáticas difusas, análisis de datos difusos, como clasificación y agrupamiento difuso. Los fundamentos de cualquier modelo de lógica difusa son los conjuntos difusos. Los conjuntos difusos como su nombre lo indica, son básicamente la aplicación de la teoría de clases de límites imprecisos. En el concepto convencional de conjuntos, un elemento o es miembro de ese conjunto o no lo es. Por otro lado, los conjuntos difusos permiten que sus elementos pertenezcan parcialmente al conjunto. A cada elemento se le asigna un grado de pertenencia al set, este valor de pertenencia puede ir de 0 (no es un miembro del conjunto) hasta 1 (sí es miembro del conjunto). A la relación entre los valores de un elemento y su grado de pertenencia a un conjunto se le llama función de pertenencia. Como se mencionó

anteriormente, los modelos difusos son adecuados para tratar con la imprecisión que puede existir en los parámetros de cualquier sistema(21).

El proceso de toma de decisión en un sistema de manufactura a menudo involucra ambigüedades. Considere como ejemplo el contexto del agrupamiento pieza-máquina, y suponga un conjunto de piezas largas y cortas que se va describir en base al atributo de "longitud". Si  $A = \{\text{conjunto de piezas cortas}\}$ , entonces debido a que la longitud empieza en la unidad 0 (cm), ubicamos a este número como el límite inferior del conjunto A. El límite superior es un poco más difícil de definir, suponga que el límite superior del conjunto A es de 3cm. Ahora el problema es, ¿porqué una pieza con una longitud de 3cm se considera como corta mientras que una pieza de 3.1cm se consideraría larga? Obviamente este es un problema estructural, Aun si el límite superior se moviera hasta 4cm, problema persistiría. Esto lleva a una inapropiada agrupación de pieza-máquina. Una forma más natural de construir con el conjunto A es relajar la estricta separación entre las definiciones de corto y largo. Para ser más concretos, un conjunto de pequeñas piezas se puede definir por medio de una función de pertenencia como se muestra a continuación(21).

De esta forma, una pieza con una longitud de 3.5cm sería clasificada como corta en un grado e 0.5. Matemáticamente es función de pertenencia se expresa como:

$$\mu_A(x) = \begin{cases} 1 & \text{si } x \leq 3 \\ (4.0 - x) & \text{si } 3 < x \leq 4 \\ 0 & \text{si } x > 4 \end{cases}$$

Ecuación 1 Función de Pertenencia para un modelo difuso (21)

Asimismo, esta característica de imprecisión es también inherente a otros atributos de una pieza, como su forma y a diversas cuestiones relacionadas con seleccionar entre diferentes opciones como en el ruteo de la información cuando se dispone de rutas alternativas para procesar los componentes. Bajo estas circunstancias los conjuntos difusos pueden proveer de un ambiente más realista para la toma de decisiones. Es importante hacer notar que los conjuntos difusos no son probabilidades(21).

Una de las dificultades en el modelo difuso es escoger un valor apropiado para la función de pertenencia. Determinar o ajustar una buena función de pertenencia no es siempre fácil, esto es debido a que la definición de la función de pertenencia es subjetiva. Se debe tener cautela

para asegurarse de representar fielmente las características de cada pieza o del proceso de manufactura(21).

### ***Modelos difusos en los problemas de TG***

Los modelos difusos han avanzado en muchas formas y sus aplicaciones pueden ser encontradas en muchas disciplinas incluyendo la manufactura. La adecuación de los modelos difusos al problema de TG fue demostrada por primera vez por Rajagopalan (1975) y Batra y Rajagopalan (1977). Desde entonces muy pocos investigadores han investigado las aplicaciones de los modelos difusos en el problema de agrupamiento pieza-máquina. Los pocos trabajos publicados hasta ahora se pueden clasificar en las dimensiones como por ejemplo: los enfoques de tipo de valor de entrada usado (21).

Los enfoques difusos pueden ser clasificados en dos tipos. El primer tipo son las versiones difusas de los métodos clásicos o convencionales, por ejemplo el agrupamiento difuso por c-media y la programación matemática difusa. El segundo tipo son las versiones difusas de los enfoques inteligentes como redes neuronales difusas y sistemas expertos difusos (21).

### **1.3. Necesidades de más investigación, justificación y objetivo**

Los logros en el uso y la aplicación de los algoritmos que se acaban de revisar han sido alentadores para problemas de tamaño mediano y grande, se ha podido comprobar su eficiencia y la cercanía que ellos alcanzan con la solución óptima del problema. No obstante cada uno de ellos tiene limitaciones propias las cuales se está buscando solucionarlas mediante la hibridación de los algoritmos, ej. Las versiones difusas del algoritmo genético o el algoritmo hormiga, también se ha avanzado en combinar las características comunes y los fundamentos de funcionamiento de cada algoritmo para crear un algoritmo final a la medida del problema de TG y en el que se puedan introducir nuevas variables al problema.

Las últimas investigaciones tratan de incorporar en los modelos de solución nuevas variables al problema como el tamaño de grupo, la secuencia de las máquinas, la carga asignada a las máquinas, costos de compra de equipo, modelos con objetivos múltiples, y tasas de utilización de los equipos.

A pesar del avance en el desarrollo de los modelos de cómputo suave (algoritmo hormiga, algoritmo genético, recocido simulado, etc.) en los problemas de TG, no se han dejado de lado los métodos más tradicionales de solución (Programación Matemática), se continúa

utilizándolos ya que estos dan el óptimo global a los problemas con lo que se puede evaluar la eficiencia de los demás métodos. Por último, las investigaciones se han enfocado en la comparación de los diferentes métodos actuales al desarrollar medidas de eficiencia para calificarlos y puntos comparativos entre ellos para evaluar sus ventajas en la aplicación.

Sin embargo, los intereses actuales están enfocados a mejorar el rendimiento de la búsqueda de optimización en términos de velocidad de convergencia y en una forma de manejar todos los diferentes criterios usados. También está contemplado analizar y correr más experimentos con otros conjuntos de datos (incluidos casos industriales concretos). Las perspectivas a analizar también contemplan el enfoque de la lógica difusa, porque describe mejor la naturaleza de la información disponible acerca de la estabilidad de la piezas.

No obstante, a pesar de la investigación realizada para obtener las familias de piezas y los grupos de trabajo, hay pocos estudios acerca del diseño de celdas con el objetivo de minimizar el movimiento intercelular de piezas, y existen aún menos estudios que consideran parámetros propios de la producción como por ejemplo: tiempos de proceso, lotes de producción, capacidad y duplicidad de las máquinas y restricciones de tipo tecnológicas y económicas. Siguiendo por esta línea, se podría considerar a la Tecnología de Grupos no sólo como una metodología de diseño de un sistema productivo, sino también como una herramienta que puede ser usada para la programación y la planeación de las fases de manufactura.

Desde un punto de vista práctico muchas alternativas emergen, tal como la tasa de utilización de la máquina, el costo de transportar piezas o el costo de duplicar maquinaria se debe tomar en cuenta al diseñar las celdas de trabajo. Existe la necesidad de desarrollar un análisis sistemático para identificar los diferentes objetivos y criterios para construir y evaluar las alternativas.

Por lo tanto, el objetivo de la presente investigación es diseñar un algoritmo genético que resuelva el problema de formación de celdas de manufactura, con base en la revisión de aportaciones previas de fuentes de información de alto impacto.

# Capítulo 2 Método

---

Este capítulo hace una revisión de 3 metodologías que se encontraron en la literatura sobre Algoritmos Genéticos para solucionar un problema de formación de celdas de trabajo; fueron seleccionadas porque ofrecen un enfoque orientado mayormente a utilizar algoritmos genéticos, a pesar que combinan otras técnicas de agrupamiento o algún heurístico especialmente diseñado para tal fin. Otra característica de estas metodologías es que toman en cuenta sólo los puntos básicos del problema de formación de celdas, de modo que se garantiza que la metodología propuesta en base a la revisión de las aportaciones realizadas por otras personas pueda ser utilizada en cualquier problema práctico de Tecnología de Grupos.

Para terminar con la revisión previa se elaboró una sección que resalta por medio de una tabla comparativa las diferencias y similitudes entre los 3 enfoques y se agrega su análisis correspondiente.

## **2.1. Descripción de Metodologías de Algoritmos Genéticos aplicadas a problemas de Tecnología de Grupos**

La manufactura celular ha emergido como una estrategia de producción capaz de resolver ciertos problemas de complejidad y grandes tiempos indirectos de manufactura en sistemas de producción por lotes a principios de los 60.

Los algoritmos genéticos son un método competitivo que se puede usar para resolver problemas grandes y que presentan perturbaciones en la entrada de datos. Los algoritmos genéticos encuentran una buena solución en lugar de encontrar el óptimo global. Una de sus principales ventajas es que sólo requieren de una función de adaptación objetivo o una función de evaluación. No requieren una representación matemática del problema. Los algoritmos genéticos se pueden usar para resolver problemas no lineales definidos en espacios de búsqueda discretos, continuos o mixtos los cuales pueden ser restringidos o no restringidos (puede quedar mejor “acotados”); también pueden explorar diferentes regiones y dirigir la búsqueda hacia regiones más promisorias del espacio de soluciones

En un nivel conceptual las 3 metodologías presentadas ignoran muchos factores de manufactura y sólo consideran las operaciones de maquinado de todos los productos, por lo que el sistema de manufactura es representado por medio de una matriz de incidencia pieza-máquina de tipo binario. Esta matriz  $[A]$  es de orden  $P \times M$ , en donde  $P =$  número de productos o piezas y  $M =$  número de máquinas. Cada elemento  $a_{p,m} = 1$  indica la visita de un producto  $p$  a la máquina  $m$ , mientras que  $a_{p,m} = 0$  indica lo contrario

		Máquinas						
		M1	M2	M3	M4	M5	M6	M7
Piezas	P1	1	0	1	0	0	0	0
	P2	1	1	0	0	0	0	0
	P3	0	0	1	1	1	0	0
	P4	0	0	0	0	1	1	0
	P5	0	0	0	0	0	1	1
	P6	1	1	0	0	0	0	0
	P7	0	0	1	1	0	0	0
	P8	0	0	0	0	0	0	1
	P9	0	1	0	0	0	0	0
	P10	0	0	0	0	0	1	1
	P11	0	0	1	0	1	0	0

Figura 5 Matriz de Boctor (1991)  $7 \times 11$  (27)

La optimización de los problemas de formación de celdas han son problemas polinomiales no determinísticos complejos, lo cual significa que la cantidad de operaciones computacionales se incrementa exponencialmente a medida que aumenta el tamaño del problema (28).

El problema de formación de celdas de manufactura es un problema de optimización combinatoria que es el NP-complejo y por lo tanto los algoritmos de optimización producen soluciones globales óptimas en tiempo computacionales prohibitivos (27). Ninguno de los enfoques presentados en este capítulo garantiza alcanzar la solución óptima, pero proveen soluciones globales óptimas o cercanas al óptimo en un tiempo computacional razonable.

### 2.1.1. Algoritmo Evolutivo para la formación de celdas de Manufactura (Gonçalves, Resende)

Fernando M. Gonçalves originario de Río de Janeiro, Brasil; actualmente trabaja en la compañía JGP Asset Management con sede en Londres, en el sector de gestión de carteras. Sus intereses están orientados a los temas económicos como son macroeconomía, economías públicas y economías monetarias (29).

Mauricio G. C. Resende nació en Maceió, Alagoas, en el noreste de Brasil. Obtuvo el grado de Ph.D. en Investigación de Operación en 1987 por la Universidad de California, Berkeley. Es miembro del personal técnico en Bell Labs, anteriormente trabajó como asesor en el Advanced Decision Support Systems en AT&T. Sus intereses incluyen temas como la optimización combinatoria, la ingeniería de algoritmos, redes y grafos, método del punto interior, conjunto de datos masivos, programación matemática, metaheurísticos, diseño de redes, modelado de investigación de operaciones y computación paralela(30).

Esta metodología propone combinar un algoritmo genético con un heurístico de búsqueda local. El algoritmo genético se utiliza para generar la agrupación de las máquinas en celdas. El heurístico local se aplica al conjunto de estas celdas que el algoritmo acaba de generar para completar la agrupación de piezas y máquinas dentro de las celdas y mejorarlo cada vez que esto sea posible (27).

### ***Codificación y representación cromosómica.***

Este algoritmo genético utiliza un conjunto de llaves asignadas desde un espacio de soluciones que utiliza variables aleatorias  $U(0,1)$ , el cual mapea el espacio de soluciones aleatorio producido por las llaves aleatorias con el espacio de soluciones del problema original (31). Una característica importante de las llaves aleatorias es que todas las soluciones producidas por el operador de cruce son soluciones factibles. Esto se logra al mover la mayor parte del problema de factibilidad al procedimiento de evaluación de la adaptación. Si cualquier vector de llaves aleatorias puede ser interpretado como una solución factible, entonces cualquier cadena producción de una combinación genética es una solución factible. A través de la dinámica de un algoritmo genético el sistema aprende la relación entre los vectores de llave aleatoria y las soluciones con mejores valores objetivo.

Los cromosomas son vectores conformados por números aleatorios y cada cromosoma está hecho de  $M + 1$  genes en donde  $M$  es el número de máquinas. El  $M + 1$ ésimo gen se utiliza para determinar el número total de celdas

La posición de cada gen se usa para determinar el número de máquina, mientras que el valor representa la celda a la cual será asignada. Las siguientes expresiones son utilizadas para codificar el cromosoma:

$$nCells = [gene_{M+1} \times M],$$

$$Cell_i = [gene_i \times nCells] \quad i = 1, \dots, M.$$

Número de máquinas = 12													
Crom=	0.70	0.89	0.12	0.54	0.37	0.78	0.41	0.19	0.94	0.64	0.68	0.31	0.29

Todos los cromosomas de la primera generación se generan de manera aleatoria.

### ***Reproducción, Cruce y Mutación***

La reproducción se lleva a cabo al copiar a los mejores individuos de una generación a la siguiente en lo que se llama una estrategia elitista.

Se utiliza un parámetro de cruce uniforme el cual trabaja de la siguiente manera:

Después que se han seleccionado dos padres al azar de la población de cromosomas (incluidos los cromosomas que se han copiado directamente a la siguiente iteración debido a la estrategia elitista), de acuerdo a una probabilidad establecida de cruce, se genera un número aleatorio  $u \in [0,1]$  por cada gen en el cromosoma hijo.

En función de la relación de incremento o decremento que guarda el número aleatorio con la probabilidad de cruce se seleccionara a uno de los dos padres que heredará su información al cromosoma hijo (32).

Para la mutación, en cada generación se incorporan uno o más miembros nuevos a la población de la misma manera en que se crearon para la población inicial. Este proceso tiene el mismo efecto que aplicar un operador de mutación bit a bit con una probabilidad pequeña

### ***Heurístico de búsqueda local***

Combinan un algoritmo genético con un heurístico de búsqueda local. Este heurístico fue creado para mejorar la calidad de las soluciones mediante el refinamiento de los cromosomas generados cada vez que esto sea posible. Si la solución modificada es mejor que la solución original, esta es reemplazada por la nueva solución. El heurístico itera hasta que la calidad de la nueva solución sea de mayor calidad que la solución previa.

La forma en que opera este heurístico es la siguiente:

1. Crea una matriz con todos los posibles valores de la función que se trata de maximizar si un producto se asigna a determinada celda. Se seleccionan los valores mayores para cada producto y se forman las familias de piezas. Calcula la eficacia grupal.

2. El heurístico continúa calculando de nuevo para las familias de piezas, todos los posibles valores si una máquina se asignara a determinada familia de pieza. Nuevamente se calcula la eficacia grupal y si es mayor a la anterior, el heurístico itera utilizando la nueva agrupación de celdas de máquinas resultante. Si es menor, el heurístico se detiene y mantiene la asignación de máquinas del inicio.

El objetivo de este heurístico es tratar de incrementar en cada iteración y por cada pieza el valor de la función que evalúa la calidad de la posible solución, si es que esa pieza se asignara a dicha celda.

En resumen el heurístico propuesto hace una comparación sucesiva entre las celdas con máquinas y las mejores asignaciones de piezas con las familias de piezas y las mejores asignaciones de máquinas, hasta encontrar una solución que no pueda ser superada.

### ***Criterio de evaluación y de parada***

La función de adaptación que se usa es la eficacia grupal, la cual se define como:

$$Eficacia\ grupal = \mu = \frac{N_1 - N_1^{Out}}{N_1 + N_0^{In}}$$

En donde:

$N_1 =$  total de 1's en la matriz  $A$

$N_1^{Out} =$  total de 1's fuera de los bloques diagonalizados

$N_0^{In} =$  número total de 0's o vacíos dentro de los bloques diagonalizados

Entre más cerca esté el valor de la eficacia grupal a 1, mejor será el agrupamiento. Las razones por las que se escogió a la eficacia grupal para este algoritmo genético son las siguientes:

- En la literatura se ha considerado como una de las medidas estándar para reportar la calidad de las soluciones.
- Se puede incorporar tanto a la utilización de máquinas dentro de la celda como al movimiento intercelular.
- Tiene una alta capacidad de diferenciar entre una matriz con buena estructura de una matriz con una estructura deficiente.
- Genera matrices con bloques diagonalizados que son muy atractivas en la práctica.
- No requiere de ningún factor de peso.

El algoritmo termina cuando se han completado un determinado número de generaciones.

### **2.1.2. Algoritmo Genético Mejorado para solucionar problemas de formación de celdas (EnGGA)**

Teerawut Tunnukij fue alumno de doctorado a tiempo completo por la Universidad de Newcastle, graduado en Diciembre del 2008 en el área de Administración. En su tesis doctoral desarrolló el algoritmo descrito a continuación y estuvo bajo la supervisión de Chris Hicks y Tom McGovern (33).

Christian Hicks es profesor de Administración de Operaciones y cabeza de grupo relacionado con el tema de Estrategia y Operaciones. Sus áreas de interés son la Administración de Operaciones, Sistemas de manufactura, planeación y Control, Diseño, Simulación, Optimización, Administración de la Cadena de Suministro, Tecnología de la Información, Investigación de Operaciones, Manufactura Esbelta, Manufactura Ágil(34) (35).

El EnGGA está basado en el Algoritmo Genético para Agrupamiento (Grouping Genetic Algorithm (GGA)) desarrollado por Falkenauer en 1998 (28) para eficientar los problemas de agrupamiento. El GGA difiere de un algoritmo genético clásico en dos importantes aspectos: se desarrolló una codificación especial para representar problemas de agrupamiento dentro de los cromosomas, y también se desarrollaron operadores genéticos especiales para adecuarse a la estructura de estos cromosomas. Con los enfoques clásicos de representación cromosómica se incurre en una redundancia significativa al representar la solución para un problema de agrupamiento. El orden de los elementos es parte de la solución, por lo que la repetición de los elementos incrementa el espacio de búsqueda y reduce potencialmente la efectividad del algoritmo genético. El esquema del GGA se enfoca en el contenido de los grupos y no en el orden(28).

En EnGGA reemplaza el heurístico estándar por un Heurístico Avaro, emplea una estrategia elitista basada en selección por el método de la ruleta. El EnGGA utiliza la codificación propuesta por Falkenauer 1998. Los operadores de cruce, mutación eliminativa y mutación divisiva se usaron sin hacerles modificaciones. El EnGGA incluye un proceso de reparación que rectifica los cromosomas infactibles que se pudieran producir en el proceso de evolución.

El EnGGa utiliza la matriz de incidencia pieza-máquina para representar la configuración inicial. El EnGGA puede solucionar los problemas de formación de celdas de manufactura sin la necesidad de predeterminar el número de celdas o el número de máquinas y piezas dentro

de cada celda. Sin embargo, no tiene caso agrupar todas las máquinas y todas las piezas dentro de solo una celda, ni tampoco tiene caso agrupar cada máquina dentro de una sola celda. Por lo tanto el número posible de celdas se define como  $2 \leq C \leq \min(M - 1, P - 1)$ .

### **Representación Genética**

La representación cromosómica consiste en tres secciones

1. Sección de piezas
2. Sección de máquinas
3. Sección de grupo

Cada gen en la sección de piezas y máquinas se conforma de un entero que representa el número de celda. Los números de pieza y máquinas se representan por medio de la posición de los genes dentro de la sección apropiada. Los enteros que representan a los números de celda en las secciones de piezas y máquinas son sólo para información porque los operadores genéticos trabajan solamente en la sección de grupos. La longitud de los cromosomas individuales puede cambiar ya que el número de celdas en cada alternativa puede variar.

$$largo_{Crom} = \sum P + \sum M + \sum C$$

La longitud cromosómica es igual a la suma de las piezas, las máquinas y el número de celdas. El orden en el cual las celdas en la sección de grupo se enlistan no es de importancia. Esta representación permite utilizar el enfoque de agrupamiento pieza-máquina. También permite aplicar los operadores de cruce y mutación en la porción del cromosoma que corresponde a las celdas. Como resultado los grupos se modifican como un todo, en lugar de modificarse individualmente

**Crom=** a b a b b a a a

### **Método para generar la población inicial**

La población inicial se genera aleatoriamente, mediante el siguiente proceso:

1. Se generan  $C$  celdas de manera aleatoria, en donde  $C$  es un entero aleatorio positivo en donde  $2 \leq C \leq M - 1$  si  $M < P$ , de otra forma  $2 \leq C \leq P - 1$ .
2. Se generan aleatoriamente  $C$  piezas y  $C$  máquinas.

3. Es entonces que se asignan las piezas y las máquinas a las celdas, para que cada celda contenga por lo menos una pieza y una máquina.
4. Se repiten los pasos 1-3 has obtener una población del tamaño requerido.

### ***Selección y Reproducción***

En el esquema de selección y reproducción los cromosomas son seleccionados aleatoriamente para ser modificados por los operadores de cruce y mutación, todos los cromosomas tienen la misma probabilidad de ser seleccionados. Las probabilidades de combinación y mutación son parámetros experimentales previamente especificados.

### ***Los operadores Genéticos***

En este algoritmo se adoptaron los operadores descritos en Falkenauer's 1998, sus operadores de cruce, mutación eliminativa y divisiva se adoptaron con modificaciones menores. Estos se integraron en un proceso de reparación que rectifica los cromosomas infactibles producidos por las operaciones genéticas. El operador de combinación incluye dos pasos:

1. Dos padres son aleatoriamente escogidos del padre. Todos los genes del primer padre se copian inicialmente al primer hijo, de la misma forma población
2. Se generan dos puntos aleatoriamente de la sección de grupo del cromosoma de cada, todos los genes del segundo padre se copian al segundo hijo. La sección cromosómica del segundo padre comprendida dentro de los puntos de cruce seleccionados se incrusta dentro del primer hijo, de la misma manera la sección del primer padre determinada por los puntos de cruce se incrusta en el segundo hijo. Todas las piezas y las máquinas que pertenecen a las celdas dentro de la sección incrustada son heredadas al cromosoma hijo.

Los pasos para la mutación son los siguientes:

1. Se escoge un padre de la población aleatoriamente
2. Se verifica el número de celdas:
  - a) Si el número de celdas es mayor a 2, se usará el operador de mutación eliminativa. Una de las celdas en la sección de grupo se selecciona de manera aleatoria y todos sus elementos se eliminan. Los elementos restantes se heredan al hijo.
  - b) Si el número de celdas es menor o igual a dos, se usará el operador de mutación divisiva. Con este tipo de operador, una celda que contiene por lo menos dos piezas y dos máquinas es seleccionada aleatoriamente y luego ésta se divide en dos nuevas

celdas. Se eligen aleatoriamente dos piezas y dos máquinas dentro de la celda seleccionada y se dividen en dos nuevas celdas. Esto asegura que cada nueva celda contenga al menos dos piezas y dos máquinas.

### *Proceso de reparación*

Los cromosomas producidos por los operadores genéticos pueden producir soluciones infactibles, por lo que se desarrolló un proceso de reparación para rectificar a estos cromosomas. El proceso de reparación consiste en 4 etapas:

1. Verificar y remover las celdas vacías. Cada elemento debe contener por lo menos una piezas y una máquina
2. Verificar el número de celdas. El posible número de celdas se define como  $2 \leq C \leq \min(M - 1, P - 1)$ :
  - Si el número de celdas dentro de los hijos producidos luego del paso 1 es uno, entonces un nuevo número de celda se insertará y las máquinas y piezas que aún no han sido asignadas ser ubicaran en la nueva celda.
  - Si el número de celdas es mayor que  $\min(M - 1, P - 1)$ , entonces se seleccionaran y se eliminaran al azar celdas hasta que el número total de celdas sea igual al  $\min(M - 1, P - 1)$ . Las piezas y máquinas que no han sido asignadas se ubicaran dentro de las celdas existentes por medio del heurístico Avaro.
3. Heurístico Avaro. El heurístico avaro evalúa la función de adecuación de todos los cromosomas posibles que pudieron haber sido producidas por todas las asignaciones alternativas de las máquinas y piezas que han quedado sin asignar. La medida de adecuación se mide en términos de la eficiencia grupal.

### *Criterio de Evaluación*

La mejor solución producida por los métodos basados en la matriz de incidencia pieza-máquina minimiza el número de vacíos dentro de los bloques diagonalizados y el número de elementos excepcionales (los 1's fuera de los bloques diagonalizados) los cuales representan en flujo intercelular. Se emplea la eficiencia grupal como la función objetivo para medir la calidad de los bloques formados en la diagonal.

### *Criterio de parada.*

El EnGGA termina cuando se han completado un determinado número de generaciones. El resultado final será la configuración de celdas representada por el cromosoma con el mayor valor en la función de adaptación.

#### **2.1.3. Algoritmo Genético para el diseño de celdas de manufactura (Murugan, Selladurai)**

V. Selladurai obtuvo su título de Ph.D. en 1994 por la Universidad de Bharathiar, es profesor del departamento de Ingeniería Mecánica en el Instituto Combatore de Tecnología en Combatore, Tamilnadu, India. Sus áreas de especialización incluyen a la Ingeniería Industrial, Ingeniería en Producción y Sistemas de Manufactura (36).

Esta metodología propone una nueva formación de células con la reducción en los tiempos de preparación entre máquinas. Su objetivo es presentar un procedimiento para obtener celdas de manufactura que consideren los datos de factores de secuencia y la reducción en los tiempos de preparación. El enfoque propuesto considera un algoritmo genético ligado a una técnica de agrupamiento por medida de similitud, el algoritmo genético es el responsable de proveerle a la técnica de agrupamiento las semillas iniciales de máquinas que serán los centroides de cada celda. Con los centroides una vez determinados ya es posible formar las familias de piezas alrededor de cada grupo de máquinas al asignar las piezas al centroide con el que tiene el mayor valor de similitud (37).

### *Representación Genética*

En esta metodología se consideran sólo dos celdas para agrupar a todas las piezas y máquinas. En la codificación se utilizan sólo los caracteres **a** y **b**, cada gen representa un número de celda y la posición de gen dentro del cromosoma representa un número de máquina. La longitud del cromosoma representa la cantidad de máquinas que se consideran en el problema.

La letra del alfabeto **a** dentro de cualquier posición del cromosoma, indica que la máquina correspondiente a esa posición será asignada a la celda número 1.

### *Inicialización*

La población inicial puede ser llevada a cabo de dos formas

- Se crea aleatoriamente una población

- Se inicia con una población previamente bien seleccionada

### ***Función de evaluación***

Cada individuo de la población se evalúa de acuerdo a la técnica de agrupamiento por medida de similitud. Con los centroides formados por cada cromosoma se calculan los coeficientes de similitud por cada pieza y se asignan a los centroides con los que tuvieron un mayor valor de similitud, los empates se rompen arbitrariamente.

Después de formar la solución final, esta se evalúa por medio de una función que combina la Eficiencia basada en Tecnología de Grupos con una medida de cohesión de los miembros de cada grupo y una eficiencia en los tiempos de preparación.

$$\beta = \lambda CPN + \mu GTE + \theta STE$$

en donde,

$\beta$  := Medida de rendimiento

$CPN$  := Cohesión de los miembros del mismo grupo

$GTE$  := Eficacia basada en Tecnología de Grupos

$STE$  := Eficiencia en los tiempos de preparación

$\lambda, \mu, \theta$  := Factores de peso para medir el rendimiento

### ***Selección y Reproducción***

La probabilidad de selección para cada cromosoma se calcula de la siguiente manera:

$$SI = \frac{F(t)}{F_{total}}$$

$$F_{Total} = \sum_{i=1}^n F(t)$$

En donde:

$F(t)$  = valor de adaptación para cada cromosoma

$F_{total}$  = total de la suma de todos los valor de adaptación

## **Cruce**

Los cromosomas que se cruzaran y los puntos de cruce son seleccionados aleatoriamente. El operador de cruce se controla por medio de la probabilidad llamada probabilidad de cruce y se le asignó un valor de 0.90, también se adoptó el tipo de cruce monopunto.

La forma más simple del operador de cruce en los algoritmos genéticos es el cruce monopunto. Este consiste en seleccionar al azar una única posición en la cadena de ambos padres e intercambiar las partes de los padres divididas por dicha posición. Este operador produce dos hijos que combinan las propiedades de ambos padres, lo que puede llevar a una mejora de la adaptación de los hijos respecto a la de los padres.

## **Mutación**

En esta metodología se utiliza la mutación por cambios. Se generan dos enteros aleatorios para cada cadena de genes y se intercambian los genes que corresponden a la posición estos números. El mismo procedimiento se repite para cada cromosoma que fue seleccionado para incurrir en el proceso de mutación.

## **Criterio de terminación**

El algoritmo finaliza después de un número predeterminado de iteraciones.

### **2.2.Evaluación de las Metodologías**

La siguiente matriz compara los pasos generales de cada metodología para resaltar sus diferencias y similitudes y destacar el nivel de detalle de cada una de ellas. Para la comparación se tomaron en cuenta las principales características que describen a los algoritmos genéticos y los puntos esenciales necesarios para su programación.

	<b>An evolutionary algorithm for manufacturing cell formation (27)</b>	<b>An enhanced Grouping Genetic Algorithm (28)</b>	<b>Manufacturing cell design with reduction in setup time through genetic algorithm (37)</b>
<b>Población inicial</b>	Se genera de forma aleatoria según la codificación establecida	Se genera de forma aleatoria según la codificación establecida	Se crea aleatoriamente una población o se inicia con una población previa bien seleccionada

	An evolutionary algorithm for manufacturing cell formation (27)	An enhanced Grouping Genetic Algorithm (28)	Manufacturing cell design with reduction in setup time through genetic algorithm (37)
<b>Codificación</b>	Números aleatorios $u \in [0,1]$	Números enteros $\mathbb{Z}$ , contiene una sección para piezas, máquinas y celdas.	Caracteres alfabéticos
<b>Representación</b>	La posición de cada gen determinar el número de máquina, su valor representa la celda a la cual será asignada.	Cada entero representa a una celda, y su posición representa el número de pieza o máquina	Cada carácter representa a una celda y la posición dentro de la cadena representa a un número de máquina
<b>Largo del cromosoma</b>	Fijo, cantidad total de máquinas + 1, $M + 1$	variable, depende del número de celdas que se asigna para cada cromosoma	Fijo, cantidad de máquinas del problema
<b>Cantidad de celdas</b>	variable, $[1, M]$	variable, limitada al intervalo $2 \leq C \leq \min(M - 1, P - 1)$	Fijo, sólo 2 celdas
<b>Selección</b>	aleatoria e incorpora una estrategia elitista	aleatoria, sin tendencias hacia escoger a los mejores individuos	Selección proporcional $SI = F(t)/F_{total}$
<b>Operador de Cruce</b>	Operador de cruce uniforme parametrizado		Monopunto, con un valor fijo a lo largo de todo el algoritmo
<b>Operador de Mutación</b>	Mutación bit a bit con baja tasa de mutación implementado a través de la		Mutación aleatoria bit a bit con valor fijo a lo largo de todo el algoritmo.

	An evolutionary algorithm for manufacturing cell formation (27)	An enhanced Grouping Genetic Algorithm (28)	Manufacturing cell design with reduction in setup time through genetic algorithm (37)
	creación de nuevos individuos en cada iteración.		
<b>Heurístico para completar la solución</b>	Heurístico de Búsqueda Local,	Ninguno, cada cromosoma es una solución completa	Técnica de agrupamiento por medida de similitud, se calculan coeficientes de similitud alrededor de los centroides aportados en el cromosoma
<b>Método extra</b>	Ninguno	Precisa de un método de reparación de los cromosomas después de los operadores genéticos	Ninguno
<b>Criterio de Evaluación</b>	Eficacia grupal	Minimiza el número de vacíos dentro de los bloques diagonalizados y el número de elementos excepcionales fuera de las celdas (Eficacia Grupal)	Función combinada entre Eficiencia grupal, cohesión de los miembros de cada grupo y una eficiencia en los tiempos de preparación.
<b>Criterio de Parada</b>	Al finalizar $x$ número de generaciones	Al finalizar $x$ número de generaciones	Al finalizar $x$ número de generaciones

Tabla 1 Análisis comparativo de las metodologías.

### **2.3.Áreas de oportunidad derivadas de la evaluación**

Derivado del análisis se encontró que hay la posibilidad de crear un algoritmo genético que simplifique el proceso de completar la solución parcial que aportan los cromosomas. En Tunnukij (28), la solución completa es descrita en el cromosoma, y construir completamente al azar; este aspecto puede entorpecer el cómputo de la solución final para problemas de gran tamaño. En Murugan(37) la forma de representación cromosómica se presenta muy limitada, de antemano es necesario predefinir un número de celdas, lo que estrecha el espacio de soluciones pero no sin contar con la certeza de que en esa reducción se encontrará el óptimo global o valores cercanos a él. En Gonçalves(27) es en donde se presenta una solución cromosómica que es parcialmente aleatoria (sólo en la asignación de máquinas y la cantidad de celdas), pero que también contiene la inteligencia de un heurístico de búsqueda local para completar la agrupación de las familias de piezas que dirige la búsqueda hacia regiones más promisorias del espacio de soluciones. Su heurístico podría ser reemplazado por un método simplificado que tome en cuenta los mismos aspectos del heurístico pero sin la necesidad de realizar iteraciones repetidas.

# Capítulo 3 Resultados

---

El capítulo 3 está estructurado de forma tal que en la primera sección se establece en términos generales en funcionamiento de cualquier algoritmo genético, sus características, parámetros y necesidades de código. La siguiente sección detalla todos los aspectos que conforman la propuesta de este trabajo de investigación el cual consiste de una metodología ecléctica y una aportación propia.

## 3.1. Impulsores

Los aspectos que me ayudaron a elegir este tema, fue que al hacer pruebas y programar un pequeño algoritmo genético que resolviese una ecuación lineal obtuve buenos resultados y el método brindaba buenas oportunidades para implementarlo en un problema de formación de celdas de manufactura.

La programación que requiere este tipo de algoritmo no es muy complicada y con conocimientos básicos de lógica de programación es posible sacar adelante al algoritmo con cualquier tipo de lenguaje de propósito general.

La revisión de la literatura también fue de gran ayuda para orientar mi visión acerca de los desarrollos en este campo y cómo se ha aplicado en el problema de formación de celdas. Es por ello que se pudieron definir y establecer los criterios de investigación y la ruta a seguir para cumplir con el objetivo.

## 3.2. Descripción Agregada de la propuesta

El Algoritmo Genético por Factores de Decisión (AGFD) se desarrolló a partir de la configuración general propuesta en el Algoritmo evolutivo para la formación de celdas (*An evolutionary algorithm for manufacturing cell formation*) (27). El Algoritmo Genético por Factores de Decisión elimina el heurístico para la asignación de piezas y lo reemplaza por un método de Asignación de Piezas basado en coeficientes que cuantifican la proporción de operaciones aportadas por piezas a las celdas y en la cantidad de densidad aportada a las celdas. Se utilizó la misma codificación de los cromosomas, sin embargo algunas

características cambiaron como la forma en que se utilizaron los operadores de selección, cruce y mutación.

### *Características de los Algoritmos Genéticos*

1. Procesan simultáneamente, no sólo una solución al problema, sino todo un conjunto de ellas. Estos algoritmos trabajan con una forma de representación de soluciones potenciales al problema (individuos). El conjunto de todos los individuos forma la población (2).
2. Trabajan con un código del conjunto de parámetros, no con el conjunto mismo (necesitan que el conjunto de parámetros del problema de optimización esté codificado en cadenas finitas sobre un determinado alfabeto). Por trabajar a nivel de código, y no con las funciones y sus variables de control, como los otros métodos, son más difíciles de “engañar”.
3. Buscan una población de puntos, no un único punto. Manteniendo una población de puntos muestrales bien adaptados, se reduce la probabilidad de caer en una cima falsa.
4. Emplean la función objetivo, no necesitan derivadas ni otra información complementaria. De este modo ganan en eficiencia y en generalidad.
5. Se valen de reglas de transición estocástica, no determinista. Los Algoritmos Genéticos se valen de operadores aleatorios para guiar la búsqueda de los mejores puntos; puede parecer extraño, pero la Naturaleza está llena de precedentes al respecto (38).

El siguiente pseudocódigo muestra un esquema general de un algoritmo evolutivo simple (AES):

```

BEGIN /*Algoritmo Genético Simple*/
  Generar la población inicial
  Computar la función de evaluación de cada individuo
  WHILE NOT stop DO
    BEGIN /*producir nueva generación*/
      FOR Tamaño Población/2 DO
        BEGIN /*Ciclo reproductivo*/
          Seleccionar dos individuos de la generación anterior para el
          cruce (probabilidad de selección proporcional a la función de
          evaluación del individuo)
          Cruzar con cierta probabilidad los dos individuos obteniendo dos
          descendientes
          Mutar los dos descendientes con cierta probabilidad.
          Computar la función de evaluación de los dos descendientes
          mutados
          Insertar los dos descendientes mutados en la nueva generación.
        END
      END
    END

```

Figura 6 Pseudocódigo de un Algoritmo Genético Simple (1).

El algoritmo procesa un conjunto de individuos que forma la población. Al comienzo del algoritmo se obtienen los datos de entrada al problema y se genera la población inicial, cuyos individuos se evalúan mediante la función de adaptación del algoritmo. El resto del algoritmo consiste en un bucle, en donde cada una de sus iteraciones es una generación en la que se produce un proceso de selección, que da mayores probabilidades de tener copias en la nueva población a los individuos más adaptados, seguido de un proceso de reproducción en el que se generan nuevos individuos a partir de la población mediante operaciones de mezcla y pequeñas alteraciones, y finalmente una evaluación de la nueva población (2).

Cualquier clase de algoritmos genéticos responden a un mismo esquema que necesita ser definido en los siguientes aspectos o puntos (28):

- I. Representación Genética
- II. Método para generar la población inicial
- III. Función de Evaluación
- IV. Esquema de selección y reproducción
- V. Operadores genéticos
- VI. Mecanismos para crear generaciones sucesivas
- VII. Criterio de parada
- VIII. Ajuste de parámetros

De acuerdo a esta clasificación se construyó y se definieron todas las características que conforman al Algoritmo Genético por Factores de Decisión, su estructura general se presenta en la siguiente figura:

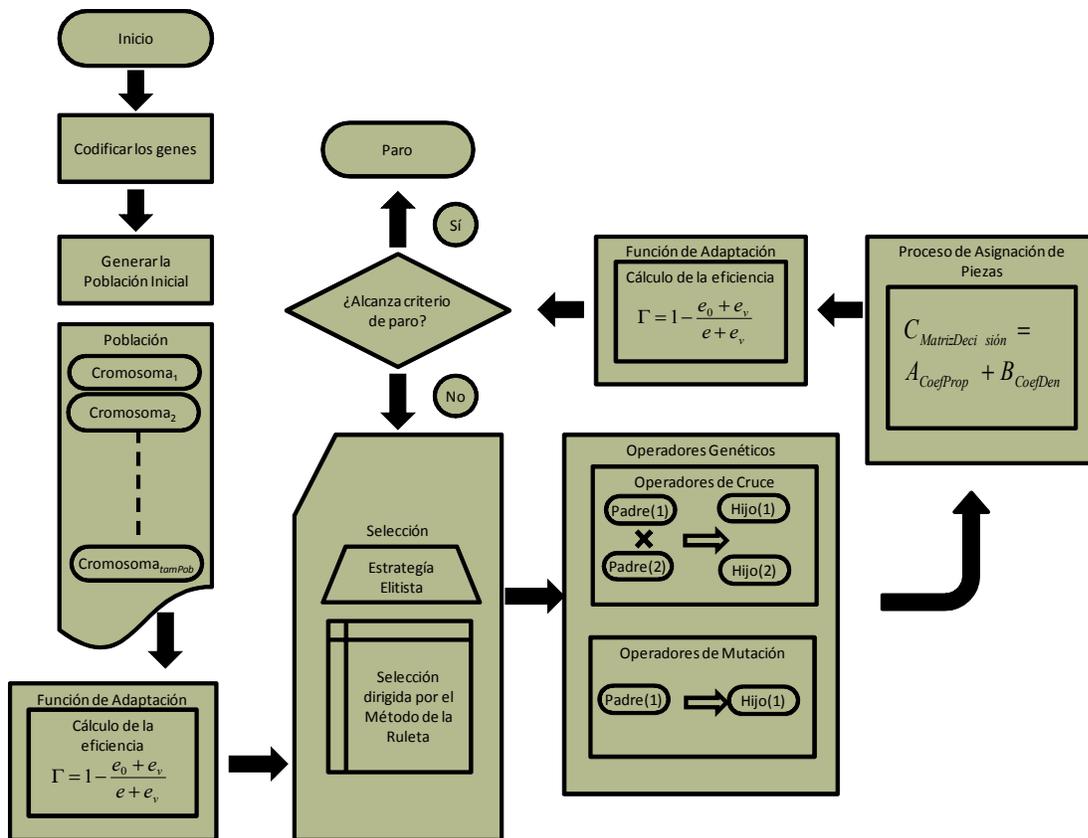


Figura 7 Estructura General del Algoritmo Genético por Factores de Decisión (AGFD).

### 3.3.Descripción Desagregada del Algoritmo Genético por Factores de Decisión

La propuesta se desarrolló integrando las aplicaciones en las diferentes metodologías que mostraron arrojar los mejores resultados y también por medio de un método original en la etapa de asignación de piezas a las celdas de máquinas ya formadas. Cabe resaltar que el contenido que lleve una referencia bibliográfica son aportaciones tomadas de otros autores, mientras que el contenido que no tenga una referencia es aportación propia

En seguida se describen detalladamente los 6 primeros puntos que fraguan al Algoritmo Genético con Factores de Decisión. Los últimos dos (Criterio de Parada y Ajuste de Parámetros) se definen en el apartado de 3.3.6 y 3.4, ya que estos no necesitan mayor definición que la asignación de valores.

### 3.3.1. Codificación y Representación de los Cromosomas

El cromosoma representa la solución parcial al problema de generación de familias de piezas y máquinas para un problema de Tecnología de Grupos. Este cromosoma está formado por una cadena de números enteros que representan, por su posición, al número de máquina; y por su valor, al número de celda al cual pertenecen. La forma en que se ideó la codificación del cromosoma sólo asigna máquinas a las celdas debido a que las piezas se asignan en un paso posterior del algoritmo. Cada número entero simboliza un gen dentro de la cadena del cromosoma. El largo de esta cadena es igual al número total de máquina más un gen extra el cual indica la cantidad de celdas que tendrá esa solución parcial en particular.

Este algoritmo genético puede solucionar el problema de formación de celdas de manufactura sin predeterminedar el número de celdas o el número de máquinas y piezas dentro de ellas, no obstante, no tiene caso agrupar todas las máquinas en una sólo celda o formar una celda por cada máquina, (28) así que se restringe el número mínimo de celdas a 2 y el número máximo al total de máquinas menos una.

El siguiente procedimiento describe la generación de la población inicial en el Algoritmo Genético por Factores de Decisión:

1. Generar una cadena de números aleatorios de largo  $(l) M + 1$ , los cuales formarán los genes de la cadena cromosómica

$$Crom_i = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 0.3 & 0.5 & 0.8 & 0.9 & 0.5 & 1 & 0.7 & 0.8 & 0.2 & 0.9 & 0.9 & 0.1 & 0.3 \\ \hline \end{array}$$

2. Convertir cada gen a un número entero que designe la celda que se asigna a dicha máquina. Empezar con el  $(M + 1)$ ésimo gen, convertir este valor a un número entero que se encuentre dentro del rango  $[2, M - 1]$ , el cual nos dará el total de Celdas para ese cromosoma  $(i)$

$$z \in [0,1] \text{ v. c.}$$

$$f(z) = (z \cdot (M - 1 + 2)) + 2 \in \mathbb{Z}$$

$$N_i = f(z)$$

- Continuar convirtiendo cada número aleatorio del intervalo  $[0,1]$  a su valor equivalente dentro del Intervalo  $[1,N]$ , en donde  $N$  es el número total de celdas para cada cromosoma ( $i$ ), valor que se acaba de obtener del paso anterior.

$$z \in [0,1] \text{ v. c.}$$

$$f(z) = ((N + 1) \cdot z) + 1 \in \mathbb{Z}$$

$$Crom_i = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 & m_8 & m_9 & m_{10} & m_{11} & m_{12} & \\ \hline 1 & 6 & 5 & 1 & 4 & 1 & 1 & 5 & 2 & 1 & 2 & 3 & 6 \\ \hline \end{array}$$

- Repetir este proceso para cada cromosoma ( $i$ ) de una población de ( $o$ ) elementos, lo cual origina la población inicial con la que empieza el algoritmo.

### 3.3.2. Selección y Reproducción

El método de selección permite orientar la búsqueda hacia los puntos más promisorios con la mayor adaptación observada hasta el momento. El operador de selección genera a partir de la población actual una población intermedia del mismo tamaño, reproduciendo con un mayor número de copias a los individuos más aptos y eliminando o asignando un número menor de copias a los individuos menos aptos. El operador de selección no produce puntos nuevos en el espacio de búsqueda, sino que determina qué individuos dejarán descendencia y en qué cantidad en la próxima generación (39).

Este algoritmo genético utiliza una regla de supervivencia probabilística. John Holland postuló que la estrategia óptima de solución consiste en aumentar exponencialmente el número de copias del mejor individuo observado respecto al peor. Este método se conoce como selección proporcional o por ruleta (39).

Con el método probabilístico o también conocido como selección sesgada por ruleta, la adecuación de un cromosoma en particular determina el tamaño de su segmento asignado en la ruleta. Cuando se generan un número aleatorio para seleccionar a los individuos que pasaran a la siguiente iteración, análogamente se produce un giro con la ruleta segmentada que determinará a la siguiente población en donde los mejores cromosomas tienen un mayor segmento de la ruleta y por lo tanto una mayor probabilidad de ser seleccionados (28).

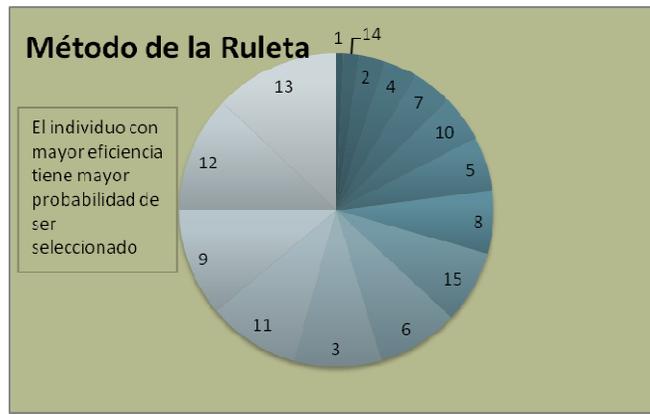


Figura 8 Representación Gráfica del Método de la Ruleta (2)

La probabilidad de selección  $p_i$  de un individuo ( $i$ ) con este método es proporcional a su adaptación relativa de éste con respecto a la población(2):

$$p_i = \frac{f(i)}{\bar{f}}$$

Siendo  $\bar{f}$  la adaptación media de la población.

A continuación se describe el procedimiento para seleccionar a la población

1. Obtener la medida de adaptación o eficiencia  $f(i)$  para cada ( $i$ ) de la población y calcular el promedio  $\bar{f}$  para toda la población
2. Ordenar los cromosomas en orden descendente respecto a su eficiencia
3. Genera la probabilidad  $p_i$  para cada individuo ( $i$ ) de la población.
4. Calcular la probabilidad acumulada  $q_i$  para cada cromosoma ( $i$ ) y así obtener el intervalo  $a_i$  de selección.

$$q_0 := 0$$

$$q_i := \sum_{1}^i p_i, \forall i$$

$$q_{i-1} < a_i < q_i$$

5. Generar ( $O - E$ ) números aleatorios e identificar a los individuos en cuyos intervalos caen dichos números aleatorios y copiarlos a la población de la siguiente iteración.

Los  $E$  individuos restantes se obtienen a partir del operador de reproducción.

La reproducción se lleva a cabo al copiar los mejores individuos de una generación a la siguiente, en lo que se conoce como una Estrategia Elitista o Elitismo(27). El elitismo consiste en asegurar la supervivencia de los mejores individuos de la población.

Dentro de los parámetros con los que se define a este algoritmo se cuenta con la Tasa de Elitismo, una medida porcentual para determinar la cantidad de los mejores (*i*) individuos que se copiarán sin ninguna modificación a la población de la siguiente iteración.

“El Elitismo mejora la búsqueda local a expensas de la perspectiva global”

Por lo tanto debemos utilizar el elitismo cuidadosamente, teniendo en cuenta las características del problema. En general, el porcentaje de la población que puede formar parte de la élite debe ser pequeño, no mayor a un 1 ó un 2% del tamaño de la población. Sin embargo es una técnica muy útil, que no sólo acelera la convergencia, sino que asegura que si en algún momento de la evolución del algoritmo hemos alcanzado una buena solución, ésta no se perderá en generaciones posteriores (2).

### 3.3.3. Operadores Genéticos

En cada nueva generación se crean algunos individuos que no estaban en la población anterior. De esta forma el algoritmo genético va accediendo a nuevas regiones del espacio de búsqueda. Los nuevos individuos se crean aplicando ciertos “operadores genéticos” a individuos de la población anterior. Los operadores que suelen estar presentes en todo algoritmo genético son los operadores de cruce y mutación. El operador de cruce combina propiedades de dos individuos de la población anterior para crear nuevos individuos. El operador de mutación crea un nuevo individuo realizando algún tipo de alteración, normalmente pequeña, en un individuo de la población anterior (2), con la aplicación del operador de mutación se logra traer características inesperadas a los hijos que no existían en los padres (40).

#### *Cruce*

El operador de cruce o de recombinación es el operador de búsqueda más importante en los algoritmos genéticos. Este es un operador sexuado que intercambia el material genético de un par de padres produciendo descendientes que normalmente difieren de sus padres. La idea central es que segmentos distintos de padres diferentes con alta adaptación deberían combinarse en nuevos individuos que tomen ventaja de esta combinación (39).

El algoritmo genético explora las regiones con mayor adaptación, ya que generaciones sucesivas de selección y combinación producen un número creciente de puntos en estas regiones (39).

En el Algoritmo Genético por Factores de Decisión se utiliza un parámetro llamado Tasa de Cruce para designar un valor porcentual que establezca la cantidad de individuos de toda la población a los que se les va a aplicar el operador de recombinación; esto permite que en algunos casos no se realice el cruce y se mantengan los padres. Una vez determinado este parámetro se genera un número aleatorio para cada individuo ( $i$ ) de la población y se compara con la tasa de Cruce, si es menor o igual a esta tasa, habrá recombinación, de otra forma, se mantendrá al cromosoma sin ninguna modificación en la población.

*Generar*  $p(i)_c \forall i \in \{1, 2, \dots, O\}$

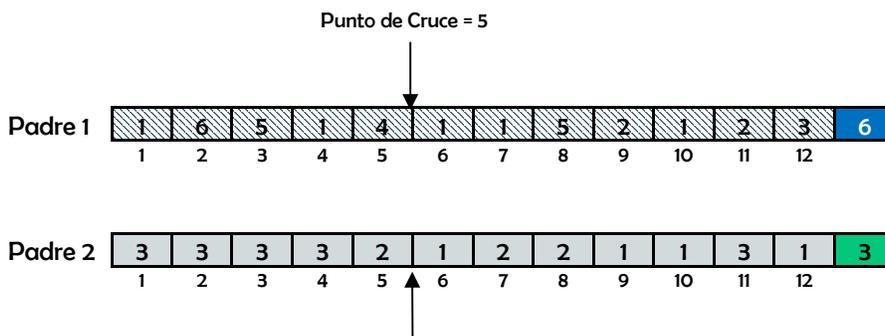
*Cuando*  $p(i)_c \leq T_c \rightarrow C(i)$

Para cada par de padres que se someterán al proceso de recombinación genética, se escoge un número al azar dentro del rango  $[2, M]$  llamado Punto de Cruce ( $k$ ), el cual designa la posición del gen en donde se cortará la cadena cromosómica para combinar dos trozos de cadenas de diferentes padres.

$$\text{Generar } k := \{k \mid k \in [2, M], k \in \mathbb{Z}\} \quad \forall p(i)_c \leq T_c$$

$$\vec{p} = (g_1, \dots, g_k, g_{k+1}, \dots, g_{l-1}, g_l) \quad \vec{r} = (g_1, \dots, g_k, j_{k+1}, \dots, j_{l-1}, g_l)$$

$$\vec{q} = (j_1, \dots, j_k, j_{k+1}, \dots, j_{l-1}, j_l) \quad \vec{s} = (j_1, \dots, j_k, g_{k+1}, \dots, g_{l-1}, j_l)$$

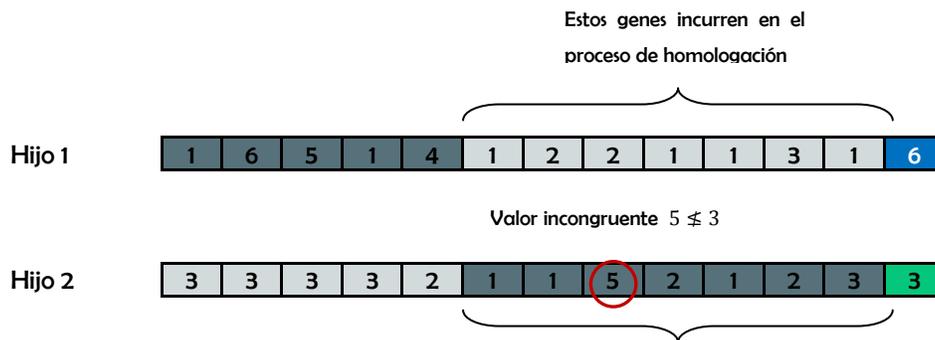


En el presente algoritmo, el último gen de cada cromosoma es fundamental porque representa el número de celdas que tendrán esa posible solución, es por ello que a cada par de genes que serán cruzados, el último gen del padre1 pasa directamente al hijo1, lo mismo pasa del padre2 al hijo2. De esta forma se explica porque el intervalo al cual pertenece el Punto de Cruce no alcanza la longitud total de la cadena cromosómica ( $M + 1$ ).

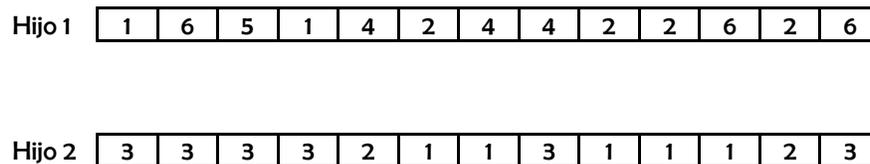
Una vez realizado el cruce, se pueden observar inconsistencias entre el número de celdas totales y posiblemente que el valor de algunos genes sobrepase el número máximo de celdas asignado a cada cromosoma. Es por ello que se creó un proceso de reparación que se aplica inmediatamente después de la recombinación genética, en este proceso se traduce cada valor que corresponde a una determinada cantidad de celdas totales a su correspondiente valor de acuerdo al nuevo número de celdas.

$$j_x \forall x \quad x := \{x \mid x \in [k + 1, \dots, l - 1]\}$$

$$g_x = \begin{cases} 1 & \text{si } 0 \leq r \leq \frac{1}{j_l} \\ 2 & \text{si } \frac{1}{g_l} < r \leq \frac{2}{j_l} \\ g_j & \text{si } \frac{g_{j-1}}{g_l} < r \leq \frac{g_j}{g_l} \\ \vdots & \vdots \\ j_l & \text{si } \frac{g_{l-1}}{g_l} < r \leq \frac{g_l}{g_l} \end{cases} \quad (\forall j \in \{1, 2, \dots, l\})$$



Una vez que se ha finaliza el proceso de reparación, los cromosomas hijos están listos para pasar al siguiente operador.



### Mutación.

La mutación es una permutación en un bit en un lugar aleatorio de la cadena cromosómica, permitiendo la alteración aleatoria del material genético. La mutación sirve para evitar la pérdida de diversidad producto de genes que han convergido a un cierto valor para toda la población, y que por lo tanto no pueden ser recuperados por el operador de cruce (39). Este operador funciona por medio de un parámetro o probabilidad asignada por el usuario llamada Tasa de Mutación ( $T_m$ ). Habitualmente la tasa de aplicación del operador es bastante pequeña comparada con el operador de cruce (2).

En muchos casos la mutación produce individuos con peor adaptación que los individuos originales, ya que la mutación puede romper las posibles correlaciones entre genes que se hayan formado con la evolución de la población. Sin embargo, contribuyen a mantener la diversidad de la población, que es fundamental para el buen funcionamiento del algoritmo (2).

La forma en que se realizaron las mutaciones para el Algoritmo Genético por Factores de Decisión fue la siguiente:

1. Se genera una probabilidad de mutación para cada individuo ( $i$ ) de la población y se compara con la tasa de Mutación, si la probabilidad es menor a este parámetro se procede con la mutación.

$$\text{Generar } p(i)_M \forall i \in \{1, 2, \dots, O\}$$

$$\text{Cuando } p(i)_M \leq T_M \rightarrow M(i)$$

*en caso contrario permanece igual*

2. Para cada individuo con probabilidad de mutación menor o igual a la Tasa de Mutación se genera un número aleatorio entero dentro del intervalo  $[1, l - 1]$  el cual indicará la posición del cromosoma que se mutará.

$$\text{crom}_i := (g_1, g_2, g_3, \dots, g_l)$$

$$x := \{x \mid x \in [1, l - 1], x \in \mathbb{Z}\} \quad \forall p(i)_M \leq T_M$$

3. Para el gen a mutar ( $g_x$ ) se produce un nuevo número aleatorio ( $r$ ) dentro del intervalo  $[0, 1]$ , el cual se convertirá en un número de celda nuevo ( $g'_x$ ) de acuerdo a la cantidad total de celdas asignadas para dicho cromosoma ( $i$ ), de la siguiente manera:

$$\text{para } g_x \quad \forall p(i)_M \leq T_M$$

$$\text{con } r := \{r \mid r \in [0, 1], r \in \mathbb{R}\} \text{ uniformemente aleatorio}$$

$$g'_x = \begin{cases} 1 & \text{si } 0 \leq r \leq \frac{1}{g_l} \\ 2 & \text{si } \frac{1}{g_l} < r \leq \frac{2}{g_l} \\ g_j & \text{si } \frac{g_{j-1}}{g_l} < r \leq \frac{g_j}{g_l} \\ \vdots & \\ \vdots & \\ g_l & \text{si } \frac{g_{l-1}}{g_l} < r \leq \frac{g_l}{g_l} \end{cases} \quad (\forall j \in \{1, 2, \dots, l\})$$

4. Se actualiza el nuevo valor ( $g'_x$ ) dentro del cromosoma.

$$crom'_i := (g_1, g_2, g_x, \dots, g_l) = (g_1, g_2, g'_x, \dots, g_l) \quad \forall i \ p(i)_M \leq T_M$$

En muchos casos la mutación produce individuos con peor adaptación que los individuos originales, ya que la mutación puede romper las posibles correlaciones entre genes que se hayan formado con la evolución de la población. Sin embargo, contribuyen a mantener la diversidad de la población, que es fundamental para el buen funcionamiento del algoritmo (2).

### 3.3.4. Asignación de Piezas

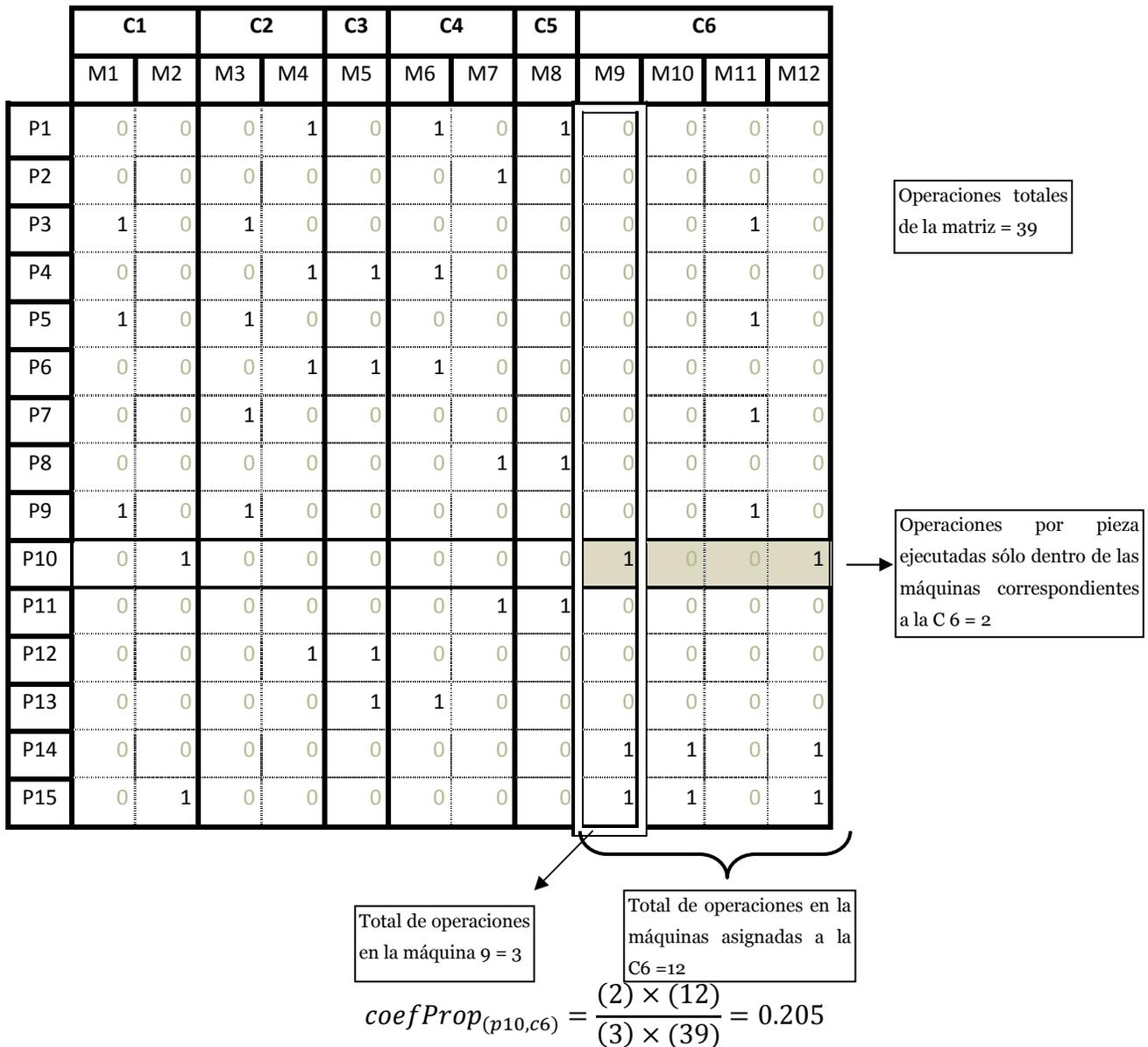
Para poder evaluar a la población de soluciones generada, se debe completar la solución parcial descrita en los cromosomas, ya que estos sólo relacionan a las máquinas con las celdas. Las piezas aún no han sido asignadas a ninguna celda. De esta forma se ha desarrollado un coeficiente que se calcula para cada pieza respecto a cada celda. Este coeficiente determina la aportación individual por pieza a la eficiencia de la solución global de cada individuo.

El coeficiente total consta de dos coeficientes parciales, la fórmula para el primer coeficiente parcial o Coeficiente de Proporción es la siguiente:

$$coefProp = \frac{\left( \sum_1^m j_{(m,p)} \in C_n, \forall p \right) \times \left( \sum_1^p \sum_1^m j_{(m,p)} \in C_n \right)}{\left( \sum_1^m j_{(m,p)} \forall p \right) \times \left( \sum_1^p \sum_1^m j_{(m,p)} \right)} \forall n$$

El primer coeficiente dentro de la fórmula es la representación matemática sobre la proporción que aportaría una pieza si esta se asignara a una celda en particular. El cálculo compensa la proporción de las operaciones por pieza que caen dentro de cada celda, por la

cantidad de operaciones dentro de la celda respecto al total de operaciones. La siguiente figura muestra un ejemplo del cálculo de este coeficiente para la pieza 10 en la celda 6



El algoritmo procede a realizar el cálculo de este coeficiente para cada pieza y cada celda formada desde la solución parcial proporcionada por los cromosomas.

El segundo coeficiente, llamado Coeficiente de Densidad, es una medida de densidad de cada pieza respecto a la mayor área probable para cada celda, esto es si se asignaran a dicha celda todas las piezas para que el alto de la celda fuera el mayor posible. Lo anterior se determina primero calculando el área de cada celda, que está dada en su dimensión de Anchura por el número de máquinas por cada celda y en la dimensión de alto por la cantidad total de

operaciones de la máquina que cumpla con la condición de tener el mayor número de operaciones y que sí tenga una operación en la pieza que se está evaluando. El divisor de la relación está dado por la anchura de la celda, la fórmula es la siguiente:

$$coefDen = \frac{\sum_1^m j_{(m,p)} \in C_n, \forall p}{\left(\sum_1^m j_{(m,p)} \in C_n, \forall p\right) \left(\sum_1^m \in C_n\right)} \forall n$$

En donde  $n \in \{1, 2, \dots, N\}$

s. a.

$$j_{(m,p)} = 1 \wedge j_{(m,p)} > m$$

$$CoefDen = \frac{\sum_1^m j_{(m,p)} \in C_n, \forall p}{\left(\sum_1^m j_{(m,p)} \in C_n, \forall p\right) \left(\sum_1^m m \in C_n\right)} \forall n$$

en donde  $n \in \{1, 2, \dots, N\}$

s. a.

$$j_{(m,p)} = 1 \wedge j_{(m,p)} > m$$

En un contexto más mundano esta fórmula tiene el siguiente razonamiento: si todas las piezas que tienen operaciones en una misma máquina de la celda se agruparan juntas, cuál sería el aporte de esa pieza al conjunto de operaciones dentro de la celda

La matriz que a continuación se presenta muestra la forma de obtener los valores para calcular el coeficiente de densidad para la pieza 10 en la celda 6:

	C1		C2		C3	C4		C5	C6			
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
P1	0	0	0	1	0	1	0	1	0	0	0	0
P2	0	0	0	0	0	0	1	0	0	0	0	0
P3	1	0	1	0	0	0	0	0	0	0	1	0
P4	0	0	0	1	1	1	0	0	0	0	0	0
P5	1	0	1	0	0	0	0	0	0	0	1	0
P6	0	0	0	1	1	1	0	0	0	0	0	0
P7	0	0	1	0	0	0	0	0	0	0	1	0
P8	0	0	0	0	0	0	1	1	0	0	0	0
P9	1	0	1	0	0	0	0	0	0	0	1	0
P10	0	1	0	0	0	0	0	0	1	0	0	1
P11	0	0	0	0	0	0	1	1	0	0	0	0
P12	0	0	0	1	1	0	0	0	0	0	0	0
P13	0	0	0	0	1	1	0	0	0	0	0	0
P14	0	0	0	0	0	0	0	0	1	1	0	1
P15	0	1	0	0	0	0	0	0	1	1	0	1

Alto posible de celda o mayor número de operaciones por máquina, en donde la pieza posee una operación en dicha máquina

Ancho de Celda o número de máquinas

$$coefDen_{(p10,c6)} = \frac{(3)}{(3) \times (4)} = 0.25$$

Se evalúa el coeficiente de Densidad para todas las intersecciones celda-pieza en la matriz. Al finalizar estos cálculos, se obtienen dos matrices de tamaño  $P \times C$ , una por cada coeficiente. Se procede a sumar estas dos matrices. De la matriz resultante es de donde se seleccionarán los valores mayores para asignar las piezas a cada celda, los empates se rompen al azar; y así completar una combinación de piezas-máquinas-celdas que representará una posible solución al problema de formación de celdas de manufactura.

$$A_{CoefProp} = \begin{bmatrix} m_p n_c & \cdots & n_c \\ \vdots & \ddots & \vdots \\ m_p & \cdots & m_p n_c \end{bmatrix}$$

$$B_{CoefDen} = \begin{bmatrix} m_p n_c & \cdots & n_c \\ \vdots & \ddots & \vdots \\ m_p & \cdots & m_p n_c \end{bmatrix}$$

$$C_{MatrizDecisión} = A_{CoefProp} + B_{CoefDen}$$

### 3.3.5. Función de Evaluación

La evolución de la población depende de la calidad relativa de los individuos que compiten por aumentar su presencia en la población y por participar en las operaciones de reproducción (libro algo gen). El presente algoritmo emplea la medida de eficiencia de agrupamiento como la función para medir la calidad de las celdas formadas propuesto por Kumar y Chandrasekharan en 1990, esta medida es ampliamente usada en la literatura (28):

$$\Gamma = 1 - \frac{e_o + e_v}{e + e_v} \quad \forall i$$

en donde:

$e_v$  = el número de vacíos dentro de las celdas

$e_o$  = el número de operaciones fuera de las celdas

$e$  = número total de operaciones dentro de la matriz

### 3.3.6. Criterio de Parada

El algoritmo propuesto tiene una condición de parada que depende de la cantidad de Iteraciones ( $t$ ) a las que se ajuste este parámetro.

Para obtener buenos diseños de celdas de manufactura se debe maximizar la utilización de las máquinas dentro de cada celda y minimizar el flujo intercelular de piezas. Se han identificado 3 enfoques para agrupar los métodos usados para formar celdas:

- Agrupamiento por medio de familias de piezas, que luego agruparan dentro de sí a las máquinas en celdas
- Agrupamiento de máquinas, el cual forma celdas basándose en las similitudes de las rutas de piezas y luego asigna piezas a las celdas
- Agrupamiento pieza-máquina, el cual forma las familias de piezas y las celdas con máquinas simultáneamente.

### 3.4.Verificación del Modelo

Las matrices que se muestran a continuación son un ejemplo de la calidad de solución que arroja el algoritmo genético con factor de Decisión.

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
P1			1	1		1				
P2	1						1			1
P3		1			1			1		
P4				1		1			1	
P5		1			1			1		
P6			1			1			1	
P7							1			1
P8		1			1			1		
P9			1	1		1			1	
P10	1						1			1
P11	1						1			1
P12	1						1			1
P13		1			1			1		
P14			1	1		1			1	
P15		1			1			1		

Figura 10 Matriz Inicial, Chan y Milner (1982) 10 × 15 (27)

Matriz resultante luego de ser resuelta por el algoritmo propuesto:

	M2	M5	M8	M3	M4	M6	M9	M1	M7	M10
P3	1	1	1							
P5	1	1	1							
P8	1	1	1							
P13	1	1	1							
P15	1	1	1							
P1				1	1	1				
P4					1	1	1			
P6				1		1	1			
P9				1	1	1	1			
P14				1	1	1	1			
P2								1	1	1
P7									1	1
P10								1	1	1
P11								1	1	1
P12								1	1	1

Figura 11 Matriz solución, Chan y Milner (1982) 10×15 (27)

### 3.4.1. Resultados

Para demostrar el rendimiento del presente algoritmo, éste se programó en Visual Basic Applications (VBA) para Excel Office 2007 y se utilizó una computadora con las siguientes características:

<b>Marca</b>	<b>Acer Aspire Mod. 5536</b>
<b>Sistema Operativo</b>	<b>Windows XP Professional SP3</b>
<b>Procesador</b>	<b>AMD Athlon Dual Core 2.10 GHz</b>
<b>Memoria RAM</b>	<b>2 GB</b>
<b>Software</b>	<b>Excel de Microsoft Office 2007</b>
<b>complementario</b>	

Se probó en 28 matrices que se recolectaron de la literatura sobre Tecnología de Grupos. Las matrices que se probaron abarcan diferentes tamaños, las más pequeñas tienen un tamaño de  $5 \times 7$  hasta matrices de  $40 \times 100$ . Estas matrices han servido de referente en varios artículos y se tuvo acceso a ellas de forma indirecta a través de (27).

Se trataron cada una de las 28 matrices de la misma forma y con los mismos parámetros:

Parámetro	Valor
Tamaño de la Población	20
Número de Iteraciones	50
Tasa de Elitismo	20%
Tasa de Cruce	90%
Tasa de Mutación	25%

El Tamaño de la Población y el Número de Iteraciones se decidieron de acuerdo a pruebas previas realizadas con el programa que corre al algoritmo genético, también se encontró en la literatura (38) que lo usual es iterar el algoritmo de 50 a 500 veces. Las Tasas de Elitismo, Cruce y Mutación se asignaron conforme a las recomendaciones en (2). Todos los valores de estos parámetros se mantuvieron constantes para cada matriz. Se replicaron 10 series en cada matriz propuesta y se reportan el mínimo, el máximo y el promedio por cada una de ellas.

La medida que se utilizó para validar el funcionamiento del algoritmo fue la eficiencia la cual ya fue explicada en el punto 3.3.5

Los resultados obtenidos por el Algoritmo Genético por Factores de Decisión se compararon con las metodologías previamente descritas en el capítulo 2 y los datos arrojados se presentan a continuación.

		Tamaño	Eficiencia			Tiempos		
			Mínimo	Promedio	Máximo	Mínimo	Promedio	Máximo
1	King and Nakornchai (1982)	5×7	73.68	73.68	73.68	0.36	0.39	0.42
2	Waghodekar and Sahu (1984)	5×7	65.22	65.22	65.22	0.34	0.36	0.39
3	Seiffodini (1989)	5×18	79.59	79.59	79.59	0.72	0.76	0.80
4	Kusiak (1992)	6×8	76.92	76.92	76.92	0.44	0.46	0.48
5	Kusiak and Chow (1987)	7×11	55.56	58.39	59.26	0.59	0.63	0.67
6	Boctor (1991)	7×11	62.50	69.58	70.37	0.61	0.63	0.67
7	Seiffodini and Wolfe	8×12	54.24	66.89	68.29	0.67	0.73	0.77
8	Chandrasekaran y Rajagopalan (1986a)	8×20	62.92	79.35	85.25	1.13	1.18	1.28
9	Chandrasekaran y	8×20	56.88	58.39	58.72	1.08	1.18	1.27

		Tamaño	Eficiencia			Tiempos		
			Mínimo	Promedio	Máximo	Mínimo	Promedio	Máximo
	Rajagopalan (1986b)							
10	Mosier y Tambe (1985a)	10×10	62.50	69.43	73.33	0.69	0.74	0.78
11	Chan y Milner (1982)	10×15	57.50	86.12	92.00	1.03	1.06	1.14
12	Askin y Subramanian (1987)	14×24	45.69	56.15	62.69	1.86	2.00	2.09
13	McCormick et al. (1972)	24×16	38.14	43.13	47.06	2.33	2.42	2.55
14	Srinivasan et al. (1990)	16×30	48.12	55.10	62.59	2.78	2.94	3.06
15	King (1980)	16×43	37.65	42.01	51.92	4.09	4.26	4.39
16	Carrie (1973)	18×24	39.13	45.06	52.34	2.28	2.50	2.64
17	Mosier and Taube 85b	20×20	36.40	38.97	40.23	9.42	10.82	12.06
18	Kumar et al. (1986)	20×23	34.36	36.54	38.89	2.64	2.75	2.94
19	Carrie (1973)	20×35	46.75	54.38	66.49	3.89	4.17	4.39
20	Boe y Cheng (1991)	20×35	34.50	42.71	52.31	4.05	4.26	4.56
21	Chandrasekaran y Rajagopalan (1989) Matriz 1	24×40	39.86	52.96	66.41	5.08	5.43	5.67
22	Chandrasekaran y Rajagopalan (1989) Matriz 2	24×40	38.19	47.29	51.53	5.14	5.51	5.89
23	Chandrasekaran y Rajagopalan (1989) Matriz 3 o Matriz 4	24×40	41.01	46.08	52.29	4.92	5.56	6.11
24	Chandrasekaran y Rajagopalan (1989) Matriz 5	24×40	35.66	36.27	36.88	5.27	5.53	5.80
25	McCormick et al.	27×27	33.20	45.43	48.75	3.81	4.05	4.33
26	Carrie (1973)	28×46	29.02	34.43	37.67	6.41	7.08	7.59
27	Kumar y Vanelli (1987)	30×41	31.94	35.76	39.44	5.86	6.47	6.91
28	Stanfel (1985)	30×50	33.13	38.48	44.44	7.64	8.00	8.70

Tabla 2 Resultados de Eficiencias y Tiempos obtenidos mediante el AGFD

# Capítulo 4 Discusión

---

El capítulo 4 corresponde al último capítulo del trabajo presentado y en él se organizan las conclusiones del modelo propuesto, se presentan los resultados aplicando el modelo y también se comparan estos con los resultados que se obtuvieron en otras investigaciones, con esta información fue posible concluir y determinar el grado de mejora que se obtuvo con los cambios que se realizaron en esta metodología

## 4.1.Mecanismos Internos

En la forma en la que evolucionan los valores de adaptación se pueden observar muchos aspectos interesantes. Uno de ellos es la forma en cómo el algoritmo va alcanzando la solución final a través de todas las iteraciones. Se puede observar que, por la naturaleza del algoritmo, una vez alcanzada una solución con un valor alto en su función de adaptación, esta solución puede perderse en una iteración posterior y el algoritmo reporta una disminución en el valor de adaptación de toda una generación. A través del tiempo en los resultados de cada iteración se pueden observar estas subidas y bajadas, pero la tendencia general es siempre en sentido de maximizar el valor de la función de adaptación.

Lo anterior es debido a un rasgo de programación del Algoritmo Genético por Factores de Decisión, como se estableció en los puntos 3.3.2 y 3.3.3, este algoritmo maneja el elitismo al momento de seleccionar a los individuos de entre la población. Sin embargo, también los mejores individuos pueden ser objeto del operador de recombinación, lo cual desmiembra a estos individuos superiores del resto y pasan a ser segmentos de código que construyen a nuevos hijos en una iteración posterior.

Otro rasgo trascendente al que hay que prestar atención es cuando a través de las iteraciones el algoritmo llega a un resultado del cual no puede moverse sino hasta después de muchas generaciones. Frecuentemente se observa que una vez que el algoritmo ha alcanzado cierto número de generaciones y ha obtenido un valor mayor a los anteriores, en varias generaciones posteriores ese valor se mantiene y al parecer, al algoritmo le cuesta trabajo encontrar otros puntos en el espacio de soluciones para obtener mejores resultados a los ya obtenidos. Claramente no se puede establecer que el algoritmo no ha encontrado el óptimo global porque el valor en el cual se atasca está muy por debajo de la eficiencia esperada y porque

eventualmente el algoritmo encuentra una mejor solución. También se puede advertir que este fenómeno es cíclico y se puede repetir varias veces en una serie del algoritmo genético.

De nueva cuenta se advierte que el resultado final del algoritmo está muy ligado al resultado de la primera iteración y a la primer población de individuos en general. Si se observan los resultados de una serie del algoritmo en donde la primer mejor eficiencia haya sido especialmente un valor alto respecto al promedio, el algoritmo alcanza valores cada vez mayores en un menor número de iteraciones, por lo que se espera que encuentre el óptimo o un valor cercano al óptimo en una menor cantidad de tiempo. De acuerdo a este comportamiento, del resultado obtenido en la primera iteración, los valores siguientes evolucionaran a partir de ese primer valor y el resultado final del algoritmo está ligado tanto al proceso de búsqueda del algoritmo genético como a la calidad de la primer solución obtenida.

El algoritmo está construido sobre una base de asignación de celdas y máquinas aleatoria (citar los puntos en dónde explico esto); y debido a esto se encuentra que existen pocas soluciones con un número elevado de celdas.

La naturaleza del algoritmo que aquí se presentó es la distribución de máquinas y piezas entre un cierto número de celdas determinadas para cada individuo de forma completamente aleatoria y sin ningún tipo de mecanismo que asegure que a cada celda se le va asignar por lo menos una máquina y una pieza.

De esta forma, nos enfrentamos a un fenómeno en dónde frecuentemente  $x$  número de celdas terminan sin ninguna máquina y por consiguiente sin ninguna pieza adjudicada, debido a esto el algoritmo también cuenta con un mecanismo para eliminar las celdas que se quedaron vacías, por lo que al finalizar la creación de una posible solución; comúnmente el número inicial de celdas se ve reducido. Cuando el número de celdas al construir la solución inicial es un número cercano al límite superior del rango  $[2, M - 1]$ , también es mayor la probabilidad de que una o más de sus celdas tengan que ser canceladas por falta de asignación de máquinas. Lo anterior crea una tendencia de determinar soluciones válidas y eficientes dentro de un rango más cerrado de número de celdas que el establecido inicialmente.

Un rasgo principal de este algoritmo se encuentra en la parte de la codificación de las posibles soluciones. Como se explicó en el punto 3.3.1 cada cromosoma representa a las máquinas siendo asignadas a un grupo de celdas para que formen parte de cada grupo de trabajo. Posteriormente el método de asignación de piezas hará lo suyo con cada pieza. No hay una

razón plausible que impida intercambiar a los elementos para llegar a una solución al algoritmo, cualquier método que integre entre sus funciones estas características sería también un algoritmo válido.

#### 4.2. Análisis externo

Para validar la propuesta presentada en esta investigación se requiere comparar los resultados obtenidos contra los resultados de las metodologías que se presentaron en el Capítulo 2. Se probaron 28 matrices tomadas de la literatura, no obstante, no están disponibles los resultados de todas las matrices para ambas metodologías, pero se cuenta con un número suficiente de valores para mantener la integridad del análisis comparativo entre las metodologías de Gonçalves(27) y Tunnukij(28) con la metodología propuesta. Murugan(37) no reporta resultados de su metodología utilizando estas matrices, excepto una, por lo que no fue posible tomar en cuenta para el análisis.

		Tamaño	Eficiencia		
			AGFD	Goncalvez (27)	EnGGA (28)
1	King and Nakornchai (1982)	5×7	73.68	73.68	82.35
2	Waghodekar and Sahu (1984)	5×7	65.22	62.5	69.57
3	Seiffodini (1989)	5×18	79.59	79.59	79.59
4	Kusiak (1992)	6×8	76.92	76.92	76.92
5	Kusiak and Chow (1987)	7×11	59.26	53.13	60.87
6	Boctor (1991)	7×11	70.37	70.37	70.83
7	Seiffodini and Wolfe	8×12	68.29	68.3	69.44
8	Chandrasekaran y Rajagopalan (1986a)	8×20	85.25	85.25	85.25
9	Chandrasekaran y Rajagopalan (1986b)	8×20	58.72	58.72	58.72
10	Mosier y Tambe (1985a)	10×10	73.33	70.59	
11	Chan y Milner (1982)	10×15	92.00	92	
12	Askin y Subramanian (1987)	14×24	62.69	69.86	
13	McCormick et al. (1972)	24×16	47.06	69.33	53.26
14	Srinivasan et al. (1990)	16×30	62.59	52.58	68.99
15	King (1980)	16×43	51.92	67.83	57.53

		Tamaño	Eficiencia		
			AGFD	Goncalvez (27)	EnGGA (28)
16	Carrie (1973)	18×24	52.34	54.86	57.73
17	Mosier and Taube 85b	20×20	40.23	54.46	
18	Kumar et al. (1986)	20×23	38.89	42.96	
19	Carrie (1973)	20×35	66.49	49.65	77.91
20	Boe y Cheng (1991)	20×35	52.31	58.07	57.98
21	Chandrasekaran y Rajagopalan (1989) Matriz 1	24×40	66.41	100	100
22	Chandrasekaran y Rajagopalan (1989) Matriz 2	24×40	51.53	85.11	85.11
23	Chandrasekaran y Rajagopalan (1989) Matriz 3 o Matriz 4	24×40	52.29	73.51	73.51
24	Chandrasekaran y Rajagopalan (1989) Matriz 5	24×40	36.88	51.97	53.29
25	McCormick et al. (1972)	27×27	48.75	54.27	54.82
26	Carrie (1973)	28×46	37.67	44.62	
27	Kumar y Vanelli (1987)	30×41	39.44	58.48	63.31
28	Stanfel (1985)	30×50	44.44	59.66	

Tabla 3 Comparación de resultados obtenidos por el algoritmo propio y dos enfoques más

El gráfico siguiente es un comparativo de las eficiencias arrojadas por el método propio y las otras metodologías con las que se está comparando y que se cuentan con sus respectivos resultados.

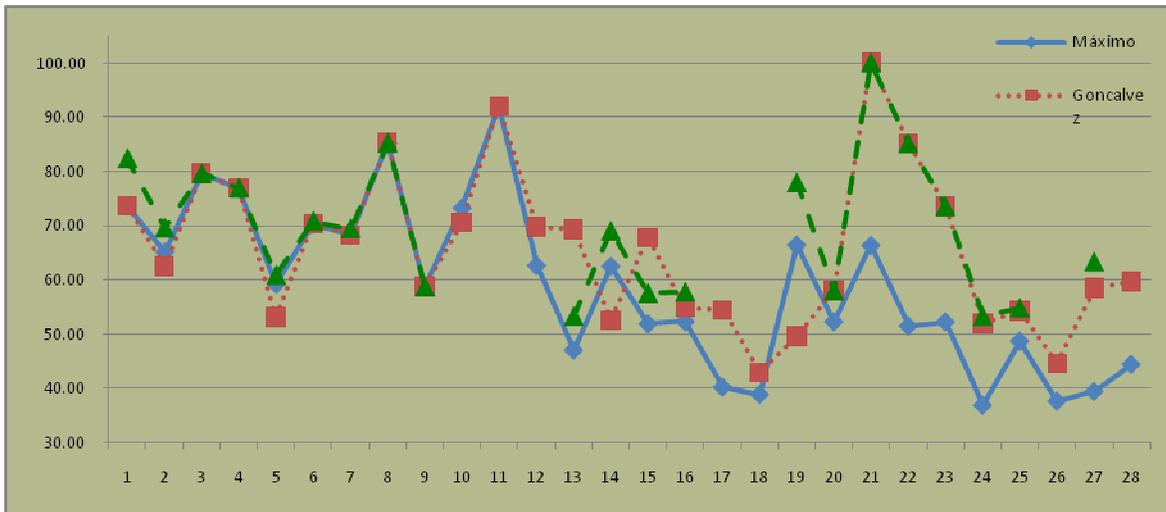


Gráfico 1 Comparación de los resultados entre las metodologías estudiadas y el AGFD, elaboración propia con base en (27)

Los parámetros que cada algoritmo utiliza están resumidos en la siguiente tabla:

Valores utilizados en los Algoritmos Genéticos			
Parámetro	Goncalvez(27)	EnGGA(28)	AGFD
Tamaño de la Población	3 * No. de filas	100, (Matriz No. 1000)	No. 13 20
Número de Iteraciones	150	50	50
Tasa de Elitismo	20%	(-)	20%
Tasa de Cruce	70%	60%, 70%, 80%, 90%, 100%	90% s. a.
Tasa de Mutación	30%	10%, 20%, 30%, 40% s. a.	25%

$p_c + p_m \leq 1$

$p_c + p_m \leq 1$

Tabla 4 Valores utilizados en los Algoritmos Genéticos (27, 28)

De los resultados obtenidos se destaca que hasta la matriz No. 11, los valores son muy cercanos entre sí y en varios de ellos los resultados son los mismos. A partir de la matriz No. 12 en adelante claramente se observa una disparidad de las eficiencias obtenidas por la metodología sugerida. Goncalvez(27) y el Tunnukij(28) en tales matrices, obtuvieron altas eficiencias y los resultados obtenidos por ambos en varios casos son los mismos. Una característica importante en las matrices en donde no se obtuvieron altas eficiencias o por lo menos similares a las encontradas por otras metodologías, es su tamaño. El tamaño mínimo

de estas matrices es de  $14 \times 24$  y las pruebas para cada una de ellas se llevaron a cabo utilizando los mismos parámetros. El EnGGA de Tunnukij prueba su metodología de la matriz 13 en adelante con un tamaño de población 1000 y 50 iteraciones. En Goncalvez, ejecuta su algoritmo genético tomando en consideración que el tamaño de la población es igual a 3 veces el número de filas en cada problema y con 150 iteraciones. Ambos parámetros en estas metodologías están muy por encima de nuestros valores para tamaño de la población y número de generaciones.

Otro aspecto importante a destacar son los valores que toman los parámetros de Cruce, Mutación y Elitismo. Los valores que esta investigación y las otras metodologías utilizaron presentan leves diferencias (de valor) pero cuentan con la característica en común que sus valores se encuentran dentro de un mismo rango para cada tipo de tasa.

Vale la pena resaltar que a pesar de que todas las metodologías aquí descritas y contra las que se probó la metodología propuesta son todos Algoritmos Genéticos, las diferencias en cada uno de sus principales componentes descritos en el punto 2.2 alteran la forma en cómo producen una solución a cada problema. La forma en que fueron diseñados sus principales componentes por muy similares que parezcan, contienen diferencias que afectan su forma de producir soluciones al problema de Formación de Celdas. La manera en que cada metodología aplicó el operador de mutación, por ejemplo, por mucho que mantenga la integridad de la finalidad de un operador de mutación, el código de programación que describe cada operador no será el mismo para ninguno de los tres algoritmos. Las funciones propias de cada algoritmo producen diferentes comportamientos en sus mecanismos internos que afectan la calibración que es posible darle a los parámetros de cruce, mutación y selección con los que se trabajan.

Por lo anterior no es posible establecer un punto de comparación válido para evaluar los resultados en base a las diferencias entre las tasas de mutación, cruce y selección que se observan en cada metodología en contraste con los valores para los parámetros de número de iteración y tamaño de Población.

Puede que la forma en la que afectan estos valores a los resultados no sea tan relevante como los valores para tamaño de la población y número de iteración. La evidencia sugiere que dependiendo del tamaño de las matrices, es necesario incrementar el número de individuos de la población y también el número de iteraciones. Los resultados en donde difieren por mucho

entre la metodología propuesta y las demás, se presentan en las matrices de mayor tamaño y nos hacen pensar que en las pruebas a este algoritmo no se le dio ni el número de iteraciones suficientes ni tampoco un número de individuos por población adecuado. Es por ello que podemos suponer que incrementando estos valores también se incrementa la probabilidad de encontrar mejores soluciones para la matriz en cuestión.

En conclusión se puede inferir que las diferencias de los resultados de las eficiencias entre esta metodología con las demás es que en las matrices de tamaño mayor a  $14 \times 24$  es necesario ejecutar el algoritmo con parámetros de tamaño de Población y Número de Iteraciones mayores a los presentados en esta investigación.

### 4.3. Aportaciones

Algoritmo Genético con Factores por Decisión cuenta con su propio método para asignar las piezas a las celdas de maquinaria ya formadas. Principalmente se basa en calcular la aportación a la eficiencia final que cada pieza aportará a dicha celda. Por ello es un método más directo que el heurístico propuestos por Gonçalves (27) y también otorga una mejor dirección de búsqueda que la representación en el cromosoma de todas las máquinas y piezas que conforman la matriz de incidencia.

Un rasgo de programación del Algoritmo Genético por Factores de Decisión es que a pesar de que se maneja el elitismo en el momento de seleccionar a las poblaciones, los mejores individuos también pueden ser objeto del operador de cruce, lo cual desintegra a estos superindividuos y pasan a ser segmentos cortados de código que constituyen a sus hijos. Por lo tanto se abre una brecha para generar ajustes o cambios en la programación del código y ver qué pasaría si se dejaran íntegros a estos individuos para ver el comportamiento general del Algoritmo Genético. Pensamos que de hacer válido este ajuste el algoritmo sería capaz de evolucionar más rápidamente hacia mejores valores del espacio de soluciones.

Sin embargo cuando son tratados como actualmente lo hace el algoritmo, podemos notar los cambios en la adaptación de los resultados a la función objetivo. Ver que existen iteraciones con un valor menor a la iteración que le antecede, permitiría estudiar de mejor forma su desarrollo el cual, a pesar de esos “tropiezos”, siempre retorna valores que vayan mejorando a través del tiempo.

#### 4.4. Futuras Investigaciones derivadas del estudio

Cada algoritmo genético cuenta con características propias y funcionan en base a los parámetros de tasa de cruce, mutación y elitismo, y al número de iteraciones y tamaño de la población, Para sacar el mayor provecho posible del algoritmo es necesario determinar los valores adecuados para estos parámetros. En la literatura no se ha hablado sobre un método confiable para asignar los valores a los parámetros con los cuales trabaja cualquier Algoritmo Genético. En el caso de la presente investigación se descubrió que es necesario realizar un análisis detallado sobre la asignación de estos parámetros para encontrar los mejores valores que al ejecutarse con el Algoritmo arrojen resultados más cercanos al óptimo y en el menor número de iteraciones. Se propone realizar un diseño de experimentos para determinar si existe una relación entre el valor de los parámetros y la convergencia del algoritmo genético hacia el óptimo global.

Un aspecto que requiere de un análisis detallado es el comportamiento de las mejores eficiencias durante cada iteración. Características como el número de resultados iguales en cada iteración sin que se haya producido un cambio significativo o la diferencia consecutiva de los mejores valores por iteración para poder calcular un número eficiente de iteraciones son aspectos que merecen una particular atención estudiar para entender completamente el funcionamiento de cada Algoritmo Genético.

En el caso de este fenómeno en el que se observa que no hay una mejora en el valor del algoritmo por varias generaciones, lo cual indica una convergencia prematura hacia óptimos locales, se propone trabajar con el valor de la probabilidad de mutación. Como se recordará el operador de mutación, diversifica a la población dentro del conjunto de soluciones vecinas y trae características inesperadas a los hijos que no existían en los padres(40) Es por ello que se propone trabajar con tasas de mutación que vayan incrementándose a medida que avanzan las iteraciones del algoritmo para sacar a la población de posibles convergencias prematuras hacia óptimos locales o para que la población pueda explorar regiones en el espacio de soluciones que el operador de Cruce no ha podido sondear.

En (art. Capítulo de libro) se exponen investigaciones sobre Algoritmos Genéticos que han tenido buenos resultados modificando la probabilidad de Mutación a medida que aumenta el número de iteraciones. Sin embargo, consideramos que se puede acceder a un método dinámico de asignación de probabilidades de cruce, mutación y elitismo para satisfacer las

necesidades del algoritmo en diferentes momentos de su progreso; de forma tal que se construya una relación entre el comportamiento de los valores de la adaptación media extendida de todos los miembros de las poblaciones y su uniformidad de acuerdo al espacio global de soluciones con el incremento en las probabilidades para crear características diferentes en las poblaciones que ya compartan uno o varios genes en un alto porcentaje de sus miembros.

Cómo se planteó en la sección 4.1, el resultado final y la evolución del algoritmo están muy ligados al resultado de la primera iteración y a la primer población de individuos en general. Se propone optimizar el algoritmo, dotándole buenas soluciones de inicio, guiando la búsqueda y reduciendo el número de soluciones posibles, aplicadas al modelo. Por ello se piensa que otro posible método para hacer más eficiente la búsqueda de cualquier algoritmo genético es solucionando un heurístico previo, el cual alimente al algoritmo con un conjunto de individuos de buena calidad para que conformen a la población inicial. De esta manera se estaría dando un paso adelante en la obtención de mejores descendientes en generaciones posteriores y muy posiblemente se podría llegar al óptimo global, o a un valor cercano a él, en un menor número de iteraciones.

Este tipo de heurísticos tendrían que ser rápidos y sencillos para no complicar un algoritmo cuya principal característica es la resolución del problema de tal forma que represente un ahorro operativo. Si puede generar resultados superiores a los del modelo actual se estaría incrementando la capacidad del algoritmo de actuar de forma veloz y eficiente.

Anteriormente se estableció que el algoritmo asigna un número de celdas inicialmente, pero que no está obligado a asignar una máquina a cada celda. Por lo que soluciones con un mayor número de celdas, son raras. Dado que este comportamiento surge de una condición enteramente aleatoria, se supone que debe existir algún tipo de relación que determine un rango más cerrado de celdas para que el algoritmo lo pueda explorar sin perder tiempo en soluciones fuera de ese esquema. Si se pudiese realizar esta mejora, incidiría directamente en mejorar la eficiencia del algoritmo, porque se podría reducir el espacio de soluciones factibles y se dotaría al algoritmo de un rasgo de dirección de búsqueda más estrecho para ganar en tiempo y en velocidad.

La forma en la que el presente algoritmo llega a la solución es primero agrupando las máquinas en grupos de trabajo dentro de los cromosomas, para luego asignar las piezas a cada

celda por medio de la fórmula de asignación. Como se discutió en un punto anterior no hay una razón plausible que impida intercambiar el orden en que se puede proceder a la formación de celdas. Se podría empezar primero codificando los cromosomas para que sean estos los que contengan a las piezas en lugar de las máquinas.

Es por ello que un área de posible investigación es estudiar las posibilidades de solución si se transponen las matrices para primero generar cromosomas que agrupe a las piezas en familias de piezas y luego utilizar el método de asignación con las máquinas para completar la solución. Posteriormente se compararían los resultados de cada matriz resuelta por las dos variaciones del mismo problema. No es necesario hacer ningún cambio al algoritmo, sólo basta con transponer cada matriz para que el cambio en la forma de proceder del algoritmo se realice de manera automática. Creemos que se pueden presentar resultados favorables en matrices en donde no se llegó a soluciones iguales o superiores a las reportadas en otras investigaciones. Potencialmente nos podremos preguntar lo siguiente: ¿de qué forma funciona mejor el algoritmo?, ¿cómo afecta la cantidad de piezas vs la cantidad de máquinas?, ¿Existe una correspondencia latente a descubrir entre piezas y máquinas.

Especulando acerca de una posible relación que afecte el funcionamiento del algoritmo entre máquinas y piezas, esta podría ser en sí un objeto de estudio e ir más allá para integrar también en esta relación a las operaciones y al total de celdas. Lo ideal sería encontrar una relación entre todos estos elementos para aprovecharla en beneficio de descubrir una solución óptima o cercana al óptimo para el problema de formación de celdas de manufactura.

Para matrices de mayor tamaño (mayores a  $24 \times 16$ ), en donde este algoritmo no obtuvo los mejores resultados, otro método propuesto para llegar a mejores soluciones es generar a través del algoritmo una solución parcial no muy buena con pocas celdas y luego separar el problema por celda y en cada una de ellas volver a aplicar el algoritmo para ir escalando hacia mejores soluciones. Al final del proceso que se llegaría a una solución con más celdas que la solución inicial. Pensamos que de esta forma se podría incrementar la eficiencia al tomar el problema y dividirlo en partes generando una solución dentro de cada celda para al final agruparlas todas en una solución global.

Este algoritmo fue validado utilizando ejemplos bien conocidos de la literatura de Formación de celdas de manufactura. Considerando los factores de programación del algoritmo, los

resultados obtenidos con él fueron satisfactorios. El programa requiere de menos de un minuto en tiempo computacional en cada situación, inclusive en los problemas más grandes.

Esta investigación presenta al Algoritmo Genético por Factores de Decisión que puede solucionar el problema de formación de celdas sin necesidad de predeterminedar el número de celdas o el número de máquinas y piezas dentro de cada celda. El Algoritmo Genético por Factores de Decisión emplea una fórmula de asignación de piezas para completar cada solución representada por los cromosomas, al mismo tiempo que incorpora características personalizadas para determinar las posibles eficiencias que la combinación de piezas y familias otorga a la solución final y así escoger la más alta.

Aunque la estructura general de los algoritmos evolutivos facilita enormemente el diseño de un algoritmo de resolución de un nuevo problema, sólo la práctica y la experiencia con este tipo de algoritmos nos permite sacarles el máximo partido, ya que sus resultados son muy dependientes de la correcta elección de sus componentes y parámetros (2). Sólo el uso y la experiencia serán los factores que determinen la profundidad del conocimiento que se tenga acerca de los algoritmos genéticos.

Si el Algoritmo Genético está bien construido la población evolucionará a lo largo de las generaciones sucesivas de tal manera que la adaptación media extendida a todos los individuos de la población, así como a la adaptación del mejor individuo se irán incrementando hacia el óptimo global. Este algoritmo demostró obtener los mismos resultados y en algunos casos superiores para matrices de hasta cierto tamaño ( $14 \times 24$ ), lo cual prueba su funcionalidad y la robustez de su programación.

Si bien es cierto que no se obtuvieron los mismos resultados comparados con las otras metodologías, es completamente factible y viable hacer los ajustes que requiere este algoritmo para estar a la par de sus similares y así usarlo en la resolución y aplicación de cualquier problema práctico y real de formación de celdas de manufactura.

# Referencias

---

1. Moujahid A, Inza I, Larrañaga P. Algoritmos Genéticos. In press 2004.
2. Araujo L, Cervigón C. Algoritmos evolutivos: un enfoque práctico. Alfaomega; 2009.
3. ISO 7144 : Documentation. Presentation of theses and similar documents 7144-86. (1986).
4. Weissberg R, Buker S. Writing up research : experimental research report writing for students of English. Englewood Cliffs, NJ: Prentice Hall Regents; 1990.
5. Wilkinson AM. The scientist's handbook for writing papers and dissertations / Antoinette Miele Wilkinson Englewood Cliffs, N.J.: Prentice Hall; 1991.
6. Reglamento de Estudios de Posgrado del Instituto Politécnico Nacional. (1991).
7. Groover MP. Solutions manual : automation, production systems and computer-integrated manufacturing. Upper Saddle River, NJ: Prentice Hall; 2001.
8. Kamrani AK, Parsaei HR. A methodology for the design of manufacturing systems using group technology. Production Planning & Control. 1994;5(5):450.
9. Burbidge JL. The simplification of material flow systems. International Journal of Production Research. 1982;20(3):339.
10. Malakooti B, Yang Z. Multiple criteria approach and generation of efficient alternatives for machine-part family formation in group technology. IIE Transactions. 2002;34(9):837.
11. Bedworth DD, Henderson MR, Wolfe PM. Computer-integrated design and manufacturing. New York: McGraw-Hill; 1991.
12. Onyeagoro EA. Group technology cell design: a case study. Production Planning & Control. 1995;6(4):365.
13. Ivanov EK, Grayson TJ. Group production organization and technology. Business Publications; 1968.

14. Mitrofanov SP, Forest Products Lab Madison WIS. Scientific Principles of Group Technology. Defense Technical Information Center; 1961.
15. Petrov VA. Flowline Group Production Planning [By] V. A. Petrov; English Translation [Of the Russian] by E. Bishop; Technical Advisor and Editor T. J. Grayson. Business Publications; 1968.
16. Castellón UJId, editor. Tecnología de Grupos: Universitat Jaume I de Castellón.
17. Chen HG, Guerrero HH. A general search algorithm for cell formation in group technology. International Journal of Production Research. 1994;32(11):2711.
18. Wemmerlov U, Hyert NL. Research issues in cellular manufacturing. International Journal of Production Research. 1987;25(3):413.
19. Domingo R, González C, Calvo R. Análisis del Diseño y Programación de Celdas en Entornos de Fabricación Ágil. Información tecnológica. 2004;15:55-60.
20. Shu Ming N. On the characterization and measure of machine cells in group technology. Operations Research. 1996;44(5):735.
21. Venugopal V. Soft-computing-based approaches to the group technology problem: a state-of-the-art review. International Journal of Production Research. 1999;37(14):3335.
22. Islier AA. Group technology by an ant system algorithm. International Journal of Production Research. 2005;43(5):913-32.
23. Sharif HH, El-Kilany KS, Helaly MA. A Genetic Algorithm Approach to the Group Technology Problem. International MultiConference of Engineers & Computer Scientists 2008. 2008:1755-60.
24. Vakharia AJ, Chang YL. Cell formation in group technology: a combinatorial search approach. International Journal of Production Research. 1997;35(7):2025-44.
25. Recocido Simulado, Contract No.: Document Number|.
26. Morales AAM. Algoritmo Basado en Tabú Search para el Cálculo del Índice de Transmisión de un Grafo. 2006;1(1):31-9.

27. Gonçalves JF, Resende MGC. An evolutionary algorithm for manufacturing cell formation. *Computers & Industrial Engineering*. 2004;47(2-3):247-73.
28. Tunnukij T, Hicks C. An Enhanced Grouping Genetic Algorithm for solving the cell formation problem. *International Journal of Production Research*. 2009;47(7):1989 - 2007.
29. Gonçalves JF. [cited]; Available from: <http://ideas.repec.org/f/pgo262.html>.
30. Resende MGC. [cited]; Available from: [http://copios2010.org/?page\\_id=527](http://copios2010.org/?page_id=527).
31. Bean JC. Robust Encodings of OR Problems for Genetic Algorithms. 1995 January 24.
32. Tomasz DG. Genetic Algorithms Reference. Tomasz Gwiazda; 2006.
33. Tunnukij T. [cited]; Available from: <http://www.ncl.ac.uk/nubs/research/researchstudents.htm>.
34. Hicks C. Citywall, Citygate, St James Boulevard, Newcastle upon Tyne, NE1 4JH: Newcastle University; [17 January, 2008; cited]; Available from: <http://www.ncl.ac.uk/nubs/staff/profile/chris.hicks>.
35. Hicks C. [cited 17/01/2011]; Available from: <http://www.staff.ncl.ac.uk/chris.hicks/>.
36. Selladurai V. [cited]; Available from: <http://www.citindia.com/Administration/Mechanical/Prof%20v%20s.htm>.
37. Murugan M, Selladurai V. Manufacturing cell design with reduction in setup time through genetic algorithm. *Asian Research Publication Network*; 2007.
38. Piedad Tolmos R-P. Introducción a los algoritmos genéticos y sus aplicaciones. ASEPUMA. Asociación Española de Profesores Universitarios de Matemáticas aplicadas a la Economía y la Empresa; 2002.
39. Estévez P. Optimización Mediante Algoritmos Genéticos. *Anales del Instituto de Ingenieros de Chile*. 1997;109(2):83-92.
40. Solimanpur M, Vrat P, Shankar R. A multi-objective genetic algorithm approach to the design of cellular manufacturing systems. *International journal of production research*. 2004;42(7):1419-41.

