



CENTRO DE INVESTIGACIONES
EN OPTICA, A.C.

“APLICACIÓN DE METAHEURÍSTICAS EN EL PROBLEMA DE ACOMODO DE PATRONES DE CORTE EN MATERIALES FINITOS”



Tesis para obtener el grado de Doctora en Ciencias (Óptica)
(Versión definitiva. Incluye cambios sugeridos por revisores.)

Presenta: M. en O. Anabel Rodríguez Rodríguez

Director de Tesis: Dr. Francisco Javier Cuevas de la Rosa

León · Guanajuato · México
Noviembre de 2023

*Dedicado a mi familia
con todo mi respeto y cariño.*

AGRADECIMIENTOS

Agradezco a Dios y a todos los seres de luz que han guiado mi camino para llegar hasta aquí.

Agradezco al gobierno mexicano, al CONAHCYT y al CIO por darme la oportunidad de estudiar un posgrado de excelencia como es el Doctorado en Ciencias (Óptica).

Especial agradecimiento al Doctor Francisco Javier Cuevas de la Rosa por guiar mi trabajo científico.

Agradezco a cada profesor que aportó un granito de arena a mi formación profesional y por ser una inspiración para llegar al final de esta investigación.

A mi familia, porque sin ellos no hubiese sido posible cumplir otra de mis metas.

A mis amigos por su colaboración y sincera amistad a lo largo de mi estancia en México.

A todos y cada uno de los que han hecho posible que hoy esté cumpliendo mi sueño,

Muchas Gracias.

RESUMEN

En la presente investigación se resuelve el problema de optimización de acomodo de patrones de corte en materiales finitos mediante el uso de estrategias metaheurísticas con la finalidad de aprovechar el material de acomodo y minimizar el desperdicio. Específicamente se implementó un algoritmo genético que al inicio genera soluciones aleatorias que forman la población inicial. La misma está compuesta por individuos o soluciones que son una combinación de patrones que se acomodan dentro del material de manera que se obtenga el menor desperdicio. Durante el proceso evolutivo estas soluciones aleatorias se someten a los operadores genéticos: selección, cruce y mutación, para encontrar nuevas soluciones que hereden la información genética de sus antecesores y evolucionen o mejoren en cada generación para obtener el menor desperdicio posible.

Es necesario aplicar un método de reparación si la solución aleatoria que arroja el algoritmo genético contiene patrones traslapados. En este trabajo se propone un método de reparación para evitar el traslapes de píxeles entre patrones. El procedimiento propuesto recorre cada uno de los patrones y si encuentra al menos un píxel solapado, realiza cuatro movimientos: arriba, abajo, izquierda y derecha, y se repite hasta que no aparecen más solapamientos. Además, se define una estructura llamada muérganos, que contiene patrones que se anidan de forma más compacta para minimizar el desperdicio. Esta estructura permite anidar elementos de forma más eficiente.

Para completar el procedimiento, se implementó un algoritmo secuencial codicioso (SGA) para acomodar los patrones en el área disponible del material. Se realizó una comparación entre las soluciones obtenidas con y sin muérganos obteniéndose mejores resultados utilizando muérganos, con un aprovechamiento del material superior al 70%. Al comparar el método propuesto en esta investigación con otros revisados en la literatura, los resultados experimentales demuestra que las soluciones mejoran en más de 1%, lo que hace que esta investigación sea competitiva.

Índice general

Dedicatoria	I
Agradecimientos	II
Resumen	III
Índice de figuras	VII
Índice de tablas	X
1 INTRODUCCIÓN	1
1.1 Objetivo General	2
1.1.1 Objetivos Específicos	2
1.2 Planteamiento del Problema	2
1.3 Hipótesis	3
1.4 Antecedentes	3
1.5 Aportaciones e Innovaciones de la Investigación	7
1.6 Organización de la Tesis	8
2 CONCEPTOS BÁSICOS	10
2.1 Formulación del Problema de Acomodo de Patrones	10
2.2 Características y Complejidad del Problema	11
2.3 Enfoques de Resolución	11
2.4 Aplicaciones Industriales	11
2.5 Métodos Heurísticos	12
2.5.1 Ejemplos de Métodos Heurísticos	12
2.5.1.1 Algoritmo del Vecino más Cercano	12
2.5.1.2 Algoritmo de Inserción	12
2.6 Métodos Metaheurísticos	12
2.6.1 Algoritmos Genéticos	13
2.6.2 Recocido Simulado	13

2.6.3	Búsqueda Tabú	13
2.6.4	Optimización por Enjambre de Partículas	13
2.7	Ventajas de las Metaheurísticas sobre Métodos Tradicionales	14
2.8	Funcionamiento de los Algoritmos Genéticos	14
2.8.1	Estructura de los Algoritmos Genéticos	14
2.9	Aplicaciones de los Algoritmos Genéticos	15
2.10	Algoritmos genéticos aplicados al problema de patrones de corte en ma- teriales	16
2.10.1	Ventajas de los Algoritmos Genéticos en el Acomodo de Patrones de Corte	16
2.10.2	Desventajas de los Algoritmos Genéticos en el Acomodo de Pa- trones de Corte	17
2.11	Comparación de los Algoritmos Genéticos con Otros Métodos Metaheurísti- cos	17
2.12	Conclusiones Parciales	18
3	METODOLOGÍA	19
3.1	Proceso de selección y binarización de imágenes	20
3.2	Proceso de calibración para calcular el desperdicio en píxeles y en metros	21
3.3	Codificación del cromosoma	24
3.3.1	Restricciones en la población inicial	26
3.4	Método de Reparación	26
3.5	<i>Muéganos</i>	27
3.5.1	Estructura y detección de los <i>Muéganos</i>	27
3.6	Algoritmo Goloso Secuencial	29
3.7	Conclusiones Parciales	29
4	IMPLEMENTACIÓN	31
4.1	Lenguaje de Programación Implementado	31
4.2	Interfaz Gráfica de Usuario	31
4.2.1	Parámetros en la Interfaz Gráfica de Usuario	32
4.3	Algoritmo Genético	34
4.4	Población Inicial	35
4.4.1	Codificación del Cromosoma	35
4.5	Función de Aptitud	37
4.5.1	Sensibilidad de los pesos en la Función Objetivo	39
4.6	Operadores Genéticos	41
4.6.1	Selección de Boltzmann	41

4.6.2	Cruce en 2 Puntos	44
4.6.3	Mutación	44
4.7	Reparación del Cromosoma	45
4.8	Muéganos	45
4.8.1	Algoritmo de Relleno por Difusión	45
4.9	Conclusiones Parciales	47
5	RESULTADOS	49
5.1	Comparaciones con otras investigaciones	54
5.2	Conclusiones Parciales	57
6	CONCLUSIONES Y TRABAJOS A FUTURO	58
A	INVESTIGACIÓN EN EL PLANO COMERCIAL	60
A.1	Datos reales tomados de una fábrica de calzado.	60
B	ALGORITMO DE RELLENO	
	(Flood Fill)	62
C	ALGORITMO DE ETIQUETADO DEL CROMOSOMA	63
D	ALGORITMO DE MUÉGANOS	64
E	ALGORITMO DE REPARACIÓN	65
F	APORTE CIENTÍFICO: CONGRESOS, EVENTOS Y PUBLICACIONES	67

Índice de figuras

1.1	Resultado obtenido utilizando un algoritmo genético por Francisco Cuevas et al.	4
3.1	Diagrama de flujo del proceso de reparación descrito en la sección de metodología.	19
3.2	Material y patrones binarizados. (a) Muestra de material de acomodo. (b–d) Patrones que son acomodados dentro del material.	20
3.3	Calibración. (a) Muestra un patrón de tablero de ajedrez sobre el material. (b) La cámara captura el material para el proceso de acomodo.	22
3.4	Montaje del arreglo óptico para calibración utilizado en la fábrica Tecno-boots. (a) Tablero de ajedrez en tamaño real. (b) Captura del material de acomodo en tamaño real.	23
3.5	La codificación del cromosoma contiene todos los patrones de acomodo. Cada patrón está representado por cinco alelos: identificador (P_k), ángulo de rotación (R_k), posición en filas y columnas dentro del material (i_k, j_k), y el bit de acomodo (B_k).	25
3.6	Ejemplos de los movimientos de los patrones durante el proceso de reparación. El área negra representa el material de acomodo, el área blanca representa los patrones y el área gris representa los traslapes entre patrones.	27
3.7	Estructura del <i>Muégano</i> . Cada <i>muégano</i> representa una lista de patrones que lo contienen.	28
3.8	Ejemplo de un resultado con patrones y muéganos acomodados. Dentro de la elipse roja: ejemplo de un <i>muégano</i> formado por dos patrones. Cada patrón tiene un nivel de gris diferente.	28
3.9	Diagrama de flujo del algoritmo goloso secuencial para rellenar el área del material no utilizado.	29
4.1	Interfaz gráfica de usuario para seleccionar el material de acomodo y los patrones de corte.	32
4.2	Parámetros del AG en la GUI.	33

4.3	Diagrama del Flujo del Algoritmo Genético.	35
4.4	Población inicial implementada como una lista en Python.	36
4.5	Un patrón se codifica como un gen en la codificación del cromosoma y está compuesto por cinco alelos.	36
4.6	Valores que se toman en cuenta en la población inicial para cada solución aleatoria.	37
4.7	Pesos vs Patrones Acomodados utilizando dos parámetros en la función objetivo. La fila amarilla muestra el mejor resultado, las rojas los peores. .	40
4.8	Pesos vs Patrones Acomodados utilizando tres parámetros en la función objetivo. La fila amarilla muestra el mejor resultado, las rojas los peores. .	41
4.9	Comportamiento de las funciones de selección de Boltzmann: (a) Enfriamiento Lineal. (b) Enfriamiento Geométrico. (c) Enfriamiento Exponencial.	43
4.10	Ejemplo de la cruce en 2 puntos	44
5.1	Comparación del desperdicio entre los resultados obtenidos con y sin <i>muérganos</i> . El diagrama de cajas y bigotes azules muestra una mejora del desperdicio obtenido con respecto al diagrama de cajas y bigotes naranja sin <i>muérganos</i> . La media de los resultados con <i>muérganos</i> es de 28,9%. . .	52
5.2	Comparación del tiempo entre los resultados obtenidos con y sin <i>muérganos</i> . El diagrama de cajas y bigotes azules muestra una mayor dispersión del tiempo de cálculo, y la media de los resultados es de 305 s debido al tamaño de la estructura <i>muérganos</i> . Sin embargo, se compensa con un desperdicio menor.	53
5.3	Resultado del acomodo (a) sin <i>muérganos</i> y (b) con <i>muérganos</i> . El resultado obtenido con <i>muérganos</i> genera menos desperdicio que la solución sin <i>muérganos</i> debido a la mínima separación entre patrones.	54
5.4	Resultados obtenidos con el método de reparación utilizando la forma de los patrones de otras investigaciones. Usamos la forma de los patrones (a) del artículo de Fischetti y Luzzi, (b) del artículo de Sato et al., y (c) del artículo de Mundim et al. para comparar el desperdicio y la eficiencia. . .	56
A.1	Patrones reales de piel para botas de adultos después del proceso de binarización.	61
A.2	Patrones reales de calzado de niño después del proceso de binarización. .	61
F.1	Certificado de Participación en el Congreso Riao / Optilas / MOPM 2019.	67
F.2	Certificado de Participación en el VII Congreso Internacional de Robótica y Computación (CIRC 2020).	68

F.3	Certificado de Imaging and Applied Optics Congress & Optical Sensor and Sensing Congress 2020.	69
F.4	Constancia de participación en el Foro de Inteligencia Artificial 2021. . .	70
F.5	Constancia de participación en el Congreso Mexican Optics and Photonics Meeting (MOPM) 2021.	71
F.6	Constancia de participación en el evento Día Internacional de la Niña y la Mujer en la Ciencia 2023.	72
F.7	Certificado de la Publicación en la revista Applied Sciences.	73

Índice de tablas

3.1	Variables de calibración.	22
3.2	Variables que caracterizan a los patrones en la población inicial.	25
3.3	Alelos que identifican a cada patrón.	25
4.1	Parámetros utilizados en el AG.	34
4.2	Variables de la función de aptitud utilizadas en el trabajo de Maestría.	38
5.1	Resultados obtenidos sin <i>Muéganos</i>	50
5.2	Resultados obtenidos con <i>Muéganos</i>	51
5.3	Resultados comparativos entre esta investigación y otras revisadas en el estado del arte.	57

CAPÍTULO 1

INTRODUCCIÓN

La optimización del acomodo de patrones en materiales finitos es un problema complejo que requiere soluciones eficientes. El acomodo de patrones de corte es una fase fundamental del proceso productivo en las industrias que requieren obtener piezas o patrones en los tamaños y formas requeridas con el menor desperdicio posible. Este proceso es clave en un importante número de industrias manufactureras, como la metalúrgica, la del papel, la textil, la del calzado y la del vidrio, en las cuales se hace indispensable el acomodo de patrones para maximizar el área de utilización y minimizar el desperdicio.

El acomodo óptimo de los patrones de corte es de interés para la industria ya que conlleva a un ahorro del material y a reducir los costos de producción. Si una compañía reduce la cantidad de material desperdiciado, esto significa un ahorro considerable de la materia prima y menor gasto para la empresa, lo que hace posible que el cliente se beneficie, ya que si el empresario reduce sus precios de producción hace a la compañía más competitiva en el mercado.

El problema de acomodo de patrones de corte es de gran complejidad, catalogado como un problema *NP*-completo, tanto por las características y variables que involucra como por las técnicas que se utilizan para abordarlo. Además, el espacio de búsqueda de la solución puede ser muy grande, lo que requiere un tiempo exponencial para resolverlo. El acomodo de patrones en materiales finitos es un problema de optimización combinatoria a resolver en un importante número de industrias manufactureras, como la metalúrgica, papelera, textil, calzado y vidrio. El acomodo óptimo es esencial para el mejor uso de los materiales y para minimizar el desperdicio mediante el uso de técnicas de Visión por Computador, Inteligencia Computacional y Procesamiento Digital de Imágenes [1]. Sin embargo, aún no existe un método global dada la complejidad del problema.

Es una temática en constante evolución y en la actualidad existen muchas investigaciones enfocadas en la solución de este problema.

Las metaheurísticas han surgido como un enfoque prometedor para abordar este reto, proporcionando un conjunto de algoritmos que pueden encontrar soluciones cercanas al óptimo en un tiempo razonable. Son especialmente útiles para problemas difíciles de resolver con métodos tradicionales, como el problema de la colocación de patrones en materiales finitos. En esta tesis de doctorado, se explora la aplicación de metaheurísticas al problema de acomodo de patrones en materiales finitos. El objetivo es desarrollar algoritmos novedosos que puedan mejorar las soluciones actuales y proporcionar información sobre el proceso de optimización subyacente. Con este trabajo, se espera contribuir al avance de este campo y proporcionar herramientas prácticas para aplicaciones en el mundo real.

1.1. Objetivo General

Desarrollar un sistema de visión para el acomodo de patrones de corte en materiales finitos utilizando métodos metaheurísticos, enfatizado en la industria manufacturera, principalmente en la industria del calzado.

1.1.1. Objetivos Específicos

- Revisar las técnicas implementadas hasta el momento y realizar un análisis comparativo.
- Desarrollar algoritmos computacionales de procesamiento de imágenes para pre-procesar las imágenes que intervienen en el problema.
- Implementar algunas metaheurísticas con la finalidad de disminuir el desperdicio de material y lograr una mejor distribución de los patrones.
- Implementar una aplicación computacional capaz de procesar la información obtenida en el arreglo experimental y tomar decisiones.

1.2. Planteamiento del Problema

En el presente trabajo, se plantea maximizar el número de patrones acomodados en materiales finitos, minimizando el desperdicio atendiendo a las siguientes restricciones:

- Que no existan patrones traslapados.

- Que todos los patrones estén acomodados dentro de los límites del material.
- Los defectos o agujeros en el material de acomodo deben tomarse como regiones no válidas del corte.
- Permitir la rotación u orientación de los patrones.

1.3. Hipótesis

La implementación de metaheurísticas y algoritmos de visión e inteligencia artificial aplicados en el problema de acomodo de patrones de corte en materiales finitos como la piel reducirá los desperdicios en comparación al trabajo manual que actualmente se realiza en las industrias.

1.4. Antecedentes

El Problema de corte de material fue formulado por primera vez en 1939 por el economista ruso Kantorovich [2], conocido por su teoría y desarrollo de técnicas para la asignación óptima de recursos y fue el primero que introdujo el término de la programación lineal. En 1961 Gilmore y Gomory [3] propusieron un modelo de resolución a este tipo de problemas mediante una matriz aplicando la programación lineal.

Los métodos heurísticos y metaheurísticos [4,5] pueden aplicarse a problemas de optimización complejos para aproximar soluciones, como el problema de acomodo de patrones. Estos métodos guían el proceso de búsqueda aproximando soluciones óptimas. Algunos ejemplos de algoritmos metaheurísticos incluyen algoritmos golosos (Greedy Algorithm) [6–8], recocido simulado (Simulated Annealing, SA) [9], optimización por enjambre de partículas (Particle Swarm Optimization, PSO) [10,11], y Algoritmos Genéticos (Genetic Algorithms, GAs) [12–15].

El recocido simulado [9] es un método probabilístico que se aproxima al óptimo global permitiendo algunos movimientos subóptimos. SA es un método de optimización estocástico basado en los principios de la mecánica estadística. Simula el proceso de recocido enfriando lentamente una sustancia para obtener una estructura cristalina fuerte. La fortaleza de la estructura depende de la velocidad de enfriamiento de los metales. SA puede utilizarse para aproximar soluciones globales en una búsqueda en el espacio extendido de un problema de optimización.

Los algoritmos golosos o codiciosos [6–8] toman decisiones óptimas locales en cada paso, lo que puede conducir a soluciones subóptimas. La optimización por enjambre de

partículas [10, 11] es un algoritmo metaheurístico de optimización inspirado en la simulación del comportamiento de un enjambre de partículas observado en la naturaleza, como peces, pájaros y hormigas, entre otros.

Los algoritmos genéticos (AGs por sus siglas en español) son una de las técnicas más populares entre los algoritmos evolutivos. Se han inspirado en la Teoría de la Evolución de Darwin [13], basándose en el principio de la selección natural. Los AGs fueron implementados por John Holland, investigador de la Universidad de Michigan, a finales de la década de 1960 [15]. Los AGs son estrategias de búsqueda estocástica basadas en el mecanismo de selección natural, ya que involucran aspectos de la genética natural, imitando la evolución biológica como estrategia de búsqueda para resolver problemas [12]. Los AGs presentan tres ventajas significativas en problemas de optimización: adaptabilidad, robustez y flexibilidad [16].

A pesar de los métodos y técnicas desarrollados para automatizar el proceso de acomodo o anidamiento (conocido por *nesting* en idioma inglés), muchas industrias siguen realizando el procedimiento manualmente, lo que se traduce en una pérdida sustancial de material y tiempo al realizarlo. Se mencionan algunas investigaciones centradas en el problema de acomodo de patrones.

Yamily Zusuki Marín en su tesis de Maestría [17] realizó una simulación del acomodo de patrones utilizando algoritmos genéticos. Utilizó la codificación genética generando puntos de posicionamiento a partir del tamaño del material de acomodo con el propósito de encontrar la posición idónea para cada patrón.

Francisco Cuevas et al. [14], utilizaron un AG para el acomodo de patrones, donde la función objetivo maneja cuatro variables y se tratan los patrones anidados y no superpuestos en una misma variable, a diferencia del presente trabajo que separa estos conceptos y a cada uno le da un peso. Un ejemplo del resultado obtenido de esta investigación se muestran en la Figura 1.1.



Figura 1.1: Resultado obtenido utilizando un algoritmo genético por Francisco Cuevas et al.

Kubagawa Sato et. al. [18] propusieron nuevos algoritmos de separación y compacta-

ción para el problema de anidamiento en dos dimensiones abiertas. El trabajo desarrolló una adaptación del polígono no-fit para considerar la profundidad de penetración de las piezas. Se enfocó en un esquema de compactación iterativo, en el que el paso clave es un algoritmo de separación basado en un mapa de obstrucción.

Yi-Ping Cui et al. [19] implementaron un algoritmo que produce patrones a partir de artículos utilizando un procedimiento heurístico secuencial. Un patrón contiene tres ítems, que se producen a partir de un algoritmo codicioso que los organiza utilizando el algoritmo izquierda-abajo, y genera una posición a partir de las esquinas del primer patrón para colocarlos en una orientación específica (vertical u horizontal). Tras la generación, los valores de los elementos se ajustan mediante una fórmula de corrección de valores. Defu Zhang et al. [20] propusieron una técnica recursiva para resolver problemas de anidamiento dividiendo el material en áreas y asignando prioridades a cada elemento. Estas prioridades dependen de su tamaño para llenar eficientemente el rectángulo donde se coloca.

Gómez y Terashima-Marín [21] utilizaron un marco evolutivo para construir hiperheurísticas que resuelven el problema de anidamiento, que considera la minimización de dos objetivos: el número de patrones a acomodar y el tiempo necesario para esta tarea. También integra tres algoritmos evolutivos multiobjetivo que construyen una solución, y decide cuál de las tres soluciones es la mejor en cada paso del algoritmo. Leung et al. [22] presentaron un algoritmo de búsqueda inteligente en dos etapas para un problema bidimensional de empaquetado en tiras. En primer lugar, un algoritmo heurístico selecciona un elemento rectangular de entre todos los elementos a anidar para un espacio dado. En segundo lugar, se combinan una búsqueda local y un algoritmo de recocido simulado para mejorar las soluciones del problema.

Neuenfeldt Júnior et al. [23] exploraron valores de referencia (predictores) para la función objetivo óptima de problemas de optimización combinatoria duros basados en técnicas de minería, que pueden utilizarse para evaluar la calidad de las soluciones heurísticas para el problema de anidamiento 2D. Wei et al. [8] presentaron un método basado en la escalera de mejor ajuste *branch-and-bound* para resolver este problema. Además, se utiliza una heurística golosa para acelerar el proceso y generar una solución completa a partir de una parcial. Yin, A. H., et al. [24] propusieron un algoritmo de Programación Dinámica Heurística Rápida (QHDP). Al principio, se establece un problema de la mochila o *knapsack* unidimensional en función de la altura y la anchura de pequeños bloques rectangulares, y se genera un conjunto de discretización eficiente para cada uno. Después, cada valor del conjunto se utilizó como una posible coordenada de la línea de corte para la división del subproblema. Pinheiro et al. [25] presentaron un algoritmo de encogimiento que opera dentro de un algoritmo genético de clave aleatoria (RKGA) y reglas de coloca-

ción bien conocidas (por ejemplo, abajo-izquierda) para mejorar las soluciones parciales.

D.N.H. Rashid et al. [26] diseñaron el método *Ants Nesting Algorithm* (ANA), inspirado en las hormigas *Leptothorax*, y emula el comportamiento de las hormigas que buscan posiciones para depositar granos mientras construyen un nuevo nido. ANA es un algoritmo continuo que actualiza la posición del agente de búsqueda añadiendo la tasa de cambio: paso o velocidad. Calcula la tasa de cambio de forma diferente, ya que utiliza las soluciones actual y anterior y los valores de aptitud para generar pesos durante el proceso de optimización. Se utiliza el teorema de Pitágoras para obtener los pesos, que guían a los agentes de búsqueda durante las fases de exploración y explotación.

Es posible considerar algoritmos de aprendizaje automático para resolver el problema de anidamiento, por ejemplo la red neuronal puede integrarse en el procesamiento de imágenes como en la investigación de Ding et al. [27], que presentó un método de mejora de imágenes difusas que puede aplicarse al procesamiento de imágenes en la fase inicial para resolver el problema de anidamiento. Zhang et al. [28] propusieron transformar las características extraídas mediante un extractor de características autosupervisado y entrenado para reducir el desajuste en la distribución de características. Poshyanonda y Dagli [29] propusieron un algoritmo genético para generar una secuencia de patrones y utilizar un algoritmo de asignación para colocar la secuencia utilizando un método de deslizamiento integrado con una red neuronal.

En el problema de anidamiento, es importante abordar la cuestión de los traslapes entre patrones. Para evitar los traslapes, es necesario aplicar técnicas como la reparación de soluciones no factibles o el concepto de trama sin ajuste para comprobar el traslape entre cualquier elemento bidimensional de forma genérica. Mundim et al. [30] utilizaron el concepto *no-fit raster*, que sirve para comprobar traslapes entre cualquier patrón bidimensional de forma genérica. Además, implementan un algoritmo genético de clave aleatoria sesgada (BRKGA) para determinar la secuencia en la que se empaquetan estos patrones. Una vez obtenida la secuencia, proponen dos heurísticas basadas en los movimientos abajo-izquierda y en el concepto de *no-fit raster*, que se utilizan para ordenar los patrones en el contenedor observando criterios objetivos.

Kierkosz y Luczak [31] propusieron una heurística de una sola pasada, colocando los ítems en diferentes posiciones del material para evitar los traslapes mediante la técnica del polígono no ajustado *no-fit polygon* (NFP por sus siglas en inglés), generando así diferentes soluciones parciales, hasta encontrar la mejor. Los detalles del algoritmo NFP se encuentran en [32]. Fischetti et al. [6] utilizaron una estrategia codiciosa aplicando un modelo de Programación Mixta de Números (MIP) para el problema de anidamiento. Sato et al. [18] propusieron una adaptación del NFP para considerar la profundidad de penetración de los elementos. Se basa en un esquema de compactación iterativo que utiliza

un algoritmo de separación basado en un mapa de obstrucción rápido de búsqueda local (FOMLS).

Los algoritmos de reparación son esenciales para mejorar y evitar el traslape entre patrones en el problema de anidamiento. Estos algoritmos pueden desarrollarse utilizando un enfoque de reparación goloso o aleatorio o una estrategia metaheurística para encontrar soluciones viables. Sin embargo, eliminar los patrones traslapados no es la solución más eficaz, ya que puede provocar un mayor desperdicio de material. Existen varios métodos posibles para reparar un individuo que no es una solución factible. Thomas Bäck et al. [33] explicaron que uno de estos métodos consiste en crear una versión reparada que sea una solución local para ese individuo concreto. Esta versión reparada puede sustituir al individuo original en la población con cierta probabilidad. Según Whitley D. et al. [34] la reparación de un individuo no factible está relacionada con el aprendizaje y la evolución, factores que interactúan entre sí para mejorar y encontrar una solución reparada y viable denominada efecto Baldwin. Para un problema concreto se diseña un algoritmo de reparación específico a través de la reparación golosa o aleatoria, o heurística que guíe el proceso de reparación.

Esta investigación ofrece una solución factible al problema de acomodo de patrones en materiales finitos. Se implementó un algoritmo genético que en la población inicial genera soluciones aleatorias que contienen una combinación de patrones que serán acomodados en el material. A estas soluciones se aplican los operadores genéticos: selección, cruce y mutación, con el objetivo de encontrar nuevas soluciones a lo largo del proceso evolutivo evaluando cada individuo mediante la función objetivo y en cada iteración se guarda la mejor solución, hasta encontrar la más factible al final del algoritmo. Si esta solución se obtiene con traslapes, es sometida un proceso de reparación para eliminar el solapamiento entre patrones. Además durante el procedimiento se implementaron los muérganos, que permite que durante el proceso evolutivo se encuentren patrones que tengan entre sí un límite de píxeles contiguos vecinos para repetir esta estructura como un patrón que minimice el desperdicio. Finalmente se aplica un algoritmo goloso secuencial para rellenar las áreas del material que estén disponibles.

1.5. Aportaciones e Innovaciones de la Investigación

1. Un método de reparación propuesto se puede aplicar en una amplia gama de técnicas heurísticas y metaheurísticas de aproximación de soluciones para resolver el problema de anidamiento.
2. Un método de calibración para mejorar la medición y el escalado de artículos y

- material para el proceso de anidación.
3. Mejora de la eficiencia y reducción de residuos mediante el uso de *muéganos* como concepto innovador.
 4. La estructura *muégano* facilita la anidación de elementos de forma más compacta y minimizando los residuos.
 5. La optimización local derivada de la formación y posición óptima de *muéganos*.
 6. El algoritmo secuencial goloso (SGA) permite rellenar las áreas disponibles en el material con la selección aleatoria de patrones.
 7. Mejora de la aplicación de algoritmos metaheurísticos cuando se trabaja con cromosomas o soluciones reparadas (por ejemplo, Algoritmos Genéticos).
 8. Los resultados experimentales ponen de manifiesto que el método propuesto, en particular incorporando *muéganos*, conduce a mejorar la reducción de residuos y la calidad de las soluciones en comparación con las metodologías anteriores.
 9. Comprobar la optimalidad del software aplicando las diferentes técnicas que serán implementadas y comparando con resultados de técnicas recientes que hayan sido publicadas.
 10. Realizar un aporte significativo para la industria a partir de los resultados que se deriven de esta investigación.
 11. Generar conocimiento en el área de visión e inteligencia computacional, aportando un método robusto y novedoso que permita resolver el problema de acomodo de patrones.

1.6. Organización de la Tesis

Esta tesis se divide en 6 capítulos los cuales reflejan la investigación realizada, a continuación, se menciona una breve descripción de su contenido:

- **Capítulo 1. Introducción.**

Se realiza la introducción de la investigación para poner en contexto al lector con la descripción y la justificación del proyecto, así como los antecedentes e innovaciones alcanzadas a partir de los resultados obtenidos.

- **Capítulo 2. Conceptos Básicos.**

Se detallan los fundamentos teóricos sobre el problema de acomodación de patrones y se explican las técnicas heurísticas y metaheurísticas que se utilizan en la resolución del problema.

- **Capítulo 3. Metodología.**

Se describen las técnicas aplicadas en esta investigación. El algoritmo de calibración, las herramientas de optimización para la solución del problema y el algoritmo de reparación propuesto.

- **Capítulo 4. Implementación.**

Se detallan todas las restricciones que se tienen en cuenta y se explican los algoritmos implementados para la solución del problema en cuestión.

- **Capítulo 5. Resultados.**

Se muestran las pruebas que se realizaron durante el proceso de implementación del algoritmo, variando los patrones de corte y el material de acomodación, y se presentan los resultados obtenidos en las pruebas.

- **Capítulo 6. Conclusiones.**

Se realiza un análisis de los resultados teniendo en cuenta el cumplimiento de los objetivos, así como las ventajas y desventajas de la solución propuesta. Finalmente se mencionan los trabajos a futuro que se derivan de la presente investigación.

CAPÍTULO 2

CONCEPTOS BÁSICOS

El problema de acomodo de patrones, también conocido como el problema de corte y empaquetado, es un desafío fundamental en el campo de la optimización y la logística. Este problema se presenta en diversas aplicaciones industriales, como el corte de planchas de metal, el acomodo de piezas en procesos de manufactura, el empaquetado de objetos en contenedores y la disposición de elementos en circuitos impresos, entre otros. El objetivo principal del problema de acomodo de patrones es optimizar la disposición de un conjunto de patrones dentro de un área o espacio de manera eficiente, minimizando el desperdicio de material y maximizando la utilización del espacio disponible.

En el ámbito de la optimización, los problemas a menudo requieren encontrar soluciones óptimas en un espacio de búsqueda complejo y vasto. Las técnicas tradicionales de optimización, como los métodos exactos, pueden ser computacionalmente costosas y poco eficientes para resolver problemas de gran escala. Para abordar esta problemática, se han desarrollado dos clases de algoritmos: heurísticos y metaheurísticos. En este capítulo, se presentará una definición de estos conceptos, su aplicación en problemas de optimización y se brindarán ejemplos ilustrativos de algunas técnicas existentes.

2.1. Formulación del Problema de Acomodo de Patrones

Dada una serie de patrones con formas y tamaños predefinidos, y un área de acomodo con dimensiones fijas, el problema de acomodo de patrones consiste en determinar la posición y orientación óptima para colocar cada patrón dentro del área sin superponerse. Cada patrón tiene asociada una cantidad conocida y un área ocupada, y el objetivo es encontrar la disposición que maximice la utilización del espacio y minimice el área desperdiciada.

Además, pueden existir restricciones adicionales, como limitaciones en la orientación de los patrones o la necesidad de respetar un orden específico de acomodo.

2.2. Características y Complejidad del Problema

El problema de acomodo de patrones es conocido por ser un problema de optimización combinatoria *NP-duro*, lo que significa que no se conoce un algoritmo eficiente que pueda encontrar una solución óptima en tiempo polinómico para instancias de gran tamaño. Esto se debe a que el espacio de búsqueda crece exponencialmente con el número de patrones y el área de acomodo, lo que hace que la resolución exacta sea computacionalmente inviable para problemas de gran escala. Además, la existencia de restricciones adicionales y la naturaleza discreta de las soluciones aumentan aún más la complejidad del problema.

2.3. Enfoques de Resolución

Dada la complejidad del problema, se han desarrollado diversas técnicas de resolución para encontrar soluciones aproximadas de alta calidad. Entre los enfoques más comunes se encuentran:

- **Métodos Heurísticos:** Estos enfoques se basan en reglas empíricas y conocimientos específicos del problema para guiar la búsqueda hacia soluciones satisfactorias. Ejemplos de métodos heurísticos incluyen el algoritmo del vecino más cercano y el algoritmo de inserción.
- **Métodos Metaheurísticos:** Estas estrategias generales de búsqueda utilizan mecanismos adaptativos para explorar el espacio de soluciones de manera más eficiente. Entre las metaheurísticas ampliamente utilizadas se encuentran el algoritmo genético, el recocido simulado y la búsqueda tabú.
- **Métodos de Descomposición:** Divide el problema en subproblemas más manejables y resuelve cada uno de forma independiente antes de combinar las soluciones para obtener una solución global.

2.4. Aplicaciones Industriales

El problema de acomodo de patrones tiene aplicaciones prácticas en diversas industrias, incluyendo la manufactura, el diseño gráfico, la construcción y la logística. Por ejemplo, en la industria de la manufactura, la optimización del corte de planchas de metal o

madera puede reducir significativamente el desperdicio de material, lo que se traduce en ahorros económicos y ambientales.

2.5. Métodos Heurísticos

Los métodos heurísticos son algoritmos de búsqueda que proporcionan soluciones aproximadas para problemas de optimización sin garantizar la optimalidad. Estos enfoques se basan en reglas empíricas, conocimientos específicos del problema o intuición del diseñador para guiar la búsqueda hacia soluciones satisfactorias. Aunque no ofrecen garantías de encontrar la mejor solución, son especialmente útiles cuando los métodos exactos resultan inviables debido al tamaño del espacio de búsqueda o a la complejidad computacional.

2.5.1. Ejemplos de Métodos Heurísticos

2.5.1.1. Algoritmo del Vecino más Cercano

El algoritmo del vecino más cercano (Nearest Neighbor Algorithm) es utilizado en el Problema del Viajante de Comercio (TSP). Comienza desde una ciudad inicial y, en cada paso, selecciona la ciudad más cercana que aún no ha sido visitada, formando una ruta aproximada. Aunque puede ser rápido y efectivo en ciertos casos, puede conducir a soluciones subóptimas, ya que no tiene en cuenta el global del problema.

2.5.1.2. Algoritmo de Inserción

El algoritmo de inserción (Insertion Algorithm) es también utilizado en el TSP. Comienza con una ruta vacía y, en cada iteración, inserta una ciudad en la posición óptima para minimizar la distancia total de la ruta. La calidad de la solución depende en gran medida de la elección inicial y de los criterios de inserción utilizados.

2.6. Métodos Metaheurísticos

Los métodos metaheurísticos son estrategias de búsqueda generales y flexibles que buscan soluciones óptimas o aproximadas en espacios de búsqueda complejos. A diferencia de las heurísticas, no se basan en conocimientos específicos del problema, sino que utilizan mecanismos adaptativos para guiar la búsqueda hacia soluciones prometedoras. Estas técnicas ofrecen una mayor capacidad de exploración y explotación en el espacio de soluciones y son adecuadas para problemas difíciles y de gran escala.

2.6.1. Algoritmos Genéticos

Los algoritmos genéticos (Genetic Algorithm, GA por sus siglas en inglés) están inspirados en la evolución biológica. El término GA se originó a finales de los años 60, gracias a John Henry Holland [15], para referirse a principios de la evolución biológica aplicados con inteligencia artificial, donde se simula la teoría de Charles Darwin [13] sobre la supervivencia del más fuerte o apto. Utiliza operadores como selección, cruce y mutación para evolucionar una población de soluciones candidatas hacia soluciones óptimas. Es adecuado para problemas de optimización combinatoria y de búsqueda global [12].

2.6.2. Recocido Simulado

El método de recocido simulado (Simulated Annealing, SA) está basado en el proceso físico de recocido de metales, propuesto en 1983 por Scott Kirkpatrick, C. Daniel Gelatt y Mario P. Vecchi [9], guiando la búsqueda para problemas de optimización global. Permite movimientos hacia soluciones peores en las primeras etapas para escapar de óptimos locales y, gradualmente, reduce la probabilidad de aceptar soluciones peores a medida que avanza la búsqueda. Es útil para problemas donde las soluciones vecinas tienen una alta correlación con la calidad de la solución actual.

2.6.3. Búsqueda Tabú

La búsqueda Tabú (Tabu Search) data de finales de los 70, el nombre y la metodología son atribuidos a Fred Glover [35], esta técnica guía un procedimiento de búsqueda local para explorar el espacio de soluciones más allá del óptimo local. Permite movimientos no permitidos temporalmente para escapar de óptimos locales y explorar áreas no exploradas. Es efectivo para problemas de optimización combinatoria y donde las soluciones no se pueden mejorar directamente con movimientos locales.

2.6.4. Optimización por Enjambre de Partículas

La optimización por enjambre de partículas (Particle Swarm Optimization, PSO) está basado en el comportamiento de búsqueda colectiva de enjambres de partículas. Las soluciones candidatas (partículas) se mueven por el espacio de búsqueda, buscando soluciones óptimas a través de interacciones con otras partículas. Este método fue propuesto por los investigadores James Kennedy, Eberhart y Shi [36, 37] basados en el comportamiento de las partículas en la naturaleza. Es adecuado para problemas de optimización continua.

2.7. Ventajas de las Metaheurísticas sobre Métodos Tradicionales

Las metaheurísticas ofrecen ventajas significativas sobre los métodos tradicionales, especialmente en problemas complejos y de gran escala:

- **Eficiencia Computacional:** Las metaheurísticas son más adecuadas para abordar problemas que involucran un gran número de variables y restricciones, ya que no exploran todas las posibles soluciones, sino que se enfocan en áreas prometedoras.
- **Adaptabilidad:** Las metaheurísticas son más flexibles y adaptables a diferentes tipos de problemas, ya que no requieren información específica del dominio.
- **Exploración y Explotación:** Las metaheurísticas encuentran un equilibrio entre la exploración de nuevas soluciones y la explotación de soluciones prometedoras, lo que les permite escapar de óptimos locales y buscar soluciones más cercanas al óptimo global.

Los AGs son una familia de técnicas de optimización y búsqueda inspirados en el proceso de evolución natural. Se han convertido en una herramienta poderosa para resolver problemas de optimización, especialmente en aquellos donde los métodos tradicionales resultan ineficientes o inviables debido al tamaño del espacio de búsqueda o a la complejidad computacional [12, 16].

2.8. Funcionamiento de los Algoritmos Genéticos

Los Algoritmos Genéticos se basan en el concepto de evolución y selección natural, donde se simula el proceso de reproducción, mutación y selección para encontrar soluciones óptimas o aproximadas a problemas de optimización. La estructura básica de un Algoritmo Genético consiste en una población de individuos, cada uno representando una posible solución al problema. A lo largo de generaciones, los individuos evolucionan mediante operadores genéticos, buscando mejorar la calidad de las soluciones.

2.8.1. Estructura de los Algoritmos Genéticos

Representación del Individuo: Cada individuo en la población es codificado como un cromosoma que representa una solución del problema. La representación puede ser binaria, entera, real o cualquier otra forma adecuada para el problema en cuestión.

Generación Inicial de la Población: Se crea una población inicial de individuos aleatoriamente o utilizando métodos heurísticos. La diversidad de la población inicial es clave para explorar diferentes áreas del espacio de soluciones.

Evaluación de la Aptitud (Fitness): Cada individuo es evaluado en función de su aptitud, que mide la calidad de su solución. La aptitud guía la selección y evolución de los individuos en las generaciones siguientes.

Operadores Genéticos

Selección: Los individuos con mayor aptitud tienen una mayor probabilidad de ser seleccionados para la reproducción. La selección busca favorecer a las soluciones más prometedoras y preservar su información genética.

Cruce: Se eligen dos individuos padres y se combinan sus cromosomas para crear uno o más descendientes. El cruce busca combinar características favorables de los padres para generar soluciones potencialmente mejores. Entre los métodos de cruce más utilizados se encuentran el cruce en un punto, el cruce en múltiples puntos y el cruce uniforme.

Mutación: Con una baja probabilidad, se realizan cambios aleatorios en el cromosoma de un individuo. La mutación introduce diversidad en la población y evita converger a óptimos locales. Los métodos de mutación incluyen la mutación de un solo bit, la mutación por inversión y la mutación por intercambio. Reemplazo: Los descendientes generados reemplazan a los individuos menos aptos de la población anterior. Esto asegura que las soluciones de alta calidad persistan en las generaciones siguientes.

Condiciones de Terminación: El algoritmo se ejecuta durante un número predeterminado de generaciones o hasta que se cumpla una condición de convergencia.

2.9. Aplicaciones de los Algoritmos Genéticos

Los Algoritmos Genéticos se han aplicado con éxito en una amplia variedad de problemas de optimización, incluyendo:

- Problemas de optimización combinatoria, como el Problema del Viajante de Comercio (TSP), el Problema de la Mochila (Knapsack Problem) y el Problema de Programación de Tareas (Scheduling Problem).
- Optimización de funciones matemáticas y técnicas de aprendizaje automático.
- Diseño y optimización de estructuras, circuitos y sistemas complejos.
- Problemas de planificación y logística en la industria y el transporte.

2.10. Algoritmos genéticos aplicados al problema de patrones de corte en materiales

Los Algoritmos Genéticos (AG) han demostrado ser eficaces para abordar el problema de acomodo de patrones de corte en materiales finitos. En esta aplicación, el objetivo es optimizar la disposición y el corte de patrones en un material, como planchas de metal, madera u otros recursos, de manera que se minimice el desperdicio y se maximice la utilización del material disponible.

Codificación de Individuos: Cada individuo representa una disposición de patrones en el material. La codificación puede ser una representación binaria donde cada bit indica si un patrón se incluye o no, o una representación más compleja con coordenadas y orientación de los patrones.

Función de Aptitud (Fitness): Se define una función de aptitud que evalúa la eficiencia de la disposición de patrones, considerando el desperdicio de material y la utilización del espacio.

Generación Inicial de la Población: Se crea una población inicial de individuos mediante métodos aleatorios o heurísticos.

Operadores Genéticos: Se aplican operadores de cruce y mutación para generar nuevos individuos en cada generación. El cruce combina características de dos individuos para crear descendientes, mientras que la mutación introduce pequeños cambios aleatorios.

Evaluación de la Aptitud: Se evalúa la aptitud de los nuevos individuos generados y se seleccionan los mejores para la próxima generación.

Condiciones de Término: El algoritmo se ejecuta durante un número predeterminado de generaciones o hasta que se cumpla una condición de convergencia.

En resumen, los AGs son una herramienta valiosa y versátil para abordar el problema de acomodo de patrones de corte en materiales finitos. Debido a su capacidad para realizar una búsqueda global y mantener la diversidad en la población los convierte en una opción sólida para problemas complejos de optimización.

2.10.1. Ventajas de los Algoritmos Genéticos en el Acomodo de Patrones de Corte

- *Exploración y Diversidad:* Los AG tienen la capacidad de explorar diferentes combinaciones de patrones y mantener la diversidad en la población, lo que ayuda a escapar de óptimos locales y encontrar soluciones más eficientes.

- *Flexibilidad y Adaptabilidad:* Los AG son flexibles y pueden adaptarse a diferentes representaciones de individuos, lo que permite abordar problemas con distintas características y restricciones.
- *Búsqueda Global:* Los AG son capaces de realizar una búsqueda global en el espacio de soluciones, lo que es beneficioso en problemas donde la optimización local no garantiza una solución óptima.
- *Fácil Implementación:* Los AG son relativamente fáciles de implementar y se pueden aplicar en problemas de optimización complejos sin requerir un conocimiento exhaustivo del dominio.

2.10.2. Desventajas de los Algoritmos Genéticos en el Acomodo de Patrones de Corte

- *Tiempo de Ejecución:* Los AG pueden ser computacionalmente costosos, especialmente para problemas de gran escala con una alta cantidad de patrones.
- *Parámetros Sensibles:* La eficiencia de los AG puede depender de la selección adecuada de parámetros, como el tamaño de la población, la tasa de mutación y el operador de cruce.
- *Soluciones Aproximadas:* Aunque los AG pueden encontrar soluciones de alta calidad, no garantizan la óptima, ya que son algoritmos de búsqueda aproximada.

2.11. Comparación de los Algoritmos Genéticos con Otros Métodos Metaheurísticos

Comparado con otros métodos metaheurísticos, como el Recocido Simulado o la Búsqueda Tabú, los Algoritmos Genéticos tienen la ventaja de ser más flexibles y fáciles de implementar. También son más adecuados para problemas con un espacio de búsqueda más grande y una mayor cantidad de restricciones. Sin embargo, en algunos casos, métodos más específicos y adaptados pueden superar a los AG en términos de eficiencia y calidad de las soluciones.

Debido a su estructura modular y sus operadores genéticos permiten abordar una amplia gama de problemas de optimización, proporcionando soluciones aproximadas de alta calidad en tiempos razonables. Su capacidad para escapar de óptimos locales y explorar el espacio de soluciones hace que los GA sean especialmente adecuados para problemas

de gran escala y complejidad. Su flexibilidad y la capacidad de adaptación los convierten en una herramienta valiosa para resolver desafíos en diversas áreas de la ciencia, la ingeniería y la industria. Es importante destacar que se debe tener en cuenta el equilibrio entre el tiempo de ejecución y la calidad de las soluciones obtenidas.

2.12. Conclusiones Parciales

El problema de acomodo de patrones es un desafío apasionante y relevante en el campo de la optimización, con amplias aplicaciones en la industria y la logística. Si bien es un problema *NP*-duro, las técnicas de resolución basadas en heurísticas y metaheurísticas ofrecen soluciones aproximadas de alta calidad en tiempos razonables. La elección de la técnica de resolución adecuada dependerá de la complejidad del problema y las restricciones específicas de la aplicación. A través de la investigación y el desarrollo continuo de técnicas de optimización, es posible mejorar la eficiencia y la calidad de las soluciones en el contexto del problema de acomodo de patrones.

Las heurísticas y metaheurísticas son poderosas herramientas para abordar problemas de optimización en diversas áreas. Mientras que las heurísticas utilizan conocimiento específico del problema para guiar la búsqueda, las metaheurísticas son estrategias generales que no dependen de información particular del dominio. Al aplicar metaheurísticas al problema de acomodo de patrones, se logra una optimización más eficiente y efectiva, con ventajas significativas sobre los métodos tradicionales en términos de escalabilidad y calidad de las soluciones encontradas.

El capítulo siguiente *Metodología 3* se explican los detalles de la metodología empleada en la presente investigación.

CAPÍTULO 3

METODOLOGÍA

En esta investigación se propone un método de reparación que minimiza el desperdicio de material eliminando la menor cantidad de patrones traslapados de la solución. Además, para mejorar el traslape entre elementos, se propone una estructura denominada *muéganos*. En ella se almacenan los elementos que tienen un límite de píxeles vecinos adyacentes entre elementos. El objetivo de los *muéganos* es obtener los elementos más cercanos entre sí y luego repetirlos en el anidamiento. Por último, se implementó un algoritmo secuencial goloso (SGA) para rellenar los espacios disponibles en el material.

La figura 3.1 muestra el diagrama de flujo del proceso de reparación descrito en la metodología. El diagrama comienza con el método de calibración de la cámara para obtener las coordenadas correctas. Posteriormente, se codifican los cromosomas para aplicar el método de reparación. Los *muéganos* se crean a partir de patrones no traslapados y que entre ellos exista el menor desperdicio. Por último, el algoritmo goloso optimiza los resultados.

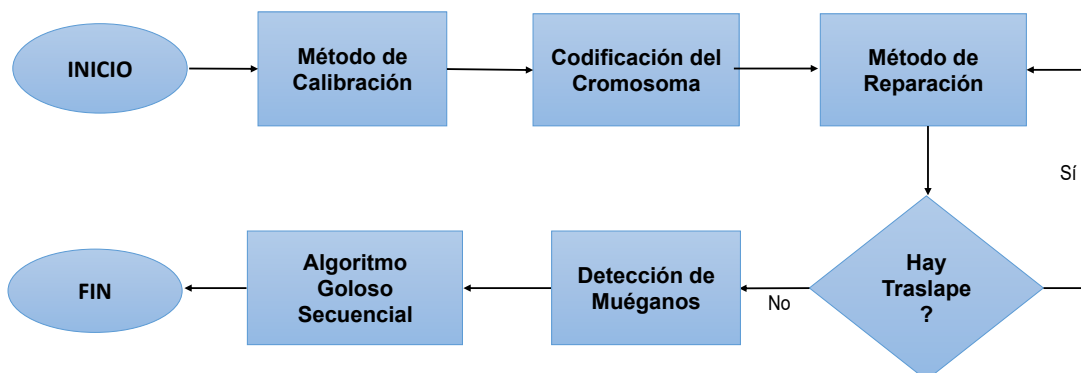


Figura 3.1: Diagrama de flujo del proceso de reparación descrito en la sección de metodología.

3.1. Proceso de selección y binarización de imágenes

En primer lugar, se seleccionan las imágenes digitales del material y los patrones, y se lleva a cabo un proceso de binarización de éstas imágenes mediante el método OTSU [38]. El área válida del material y los patrones se binarizan con valores 0 y 1, respectivamente, y el área no válida (defectos o agujeros) reciben otro valor distinto de 0 y 1. Las imágenes almacenadas en una estructura de datos junto con el área de cada una se calculan contando el número de píxeles válidos que forman parte del anidamiento. En la Figura 3.2 se muestran ejemplos de imágenes binarizadas. Éstas fueron tomadas en tamaño real en la fábrica de calzado Tecnoboots, son muestras del material y patrones de piel de vaca.

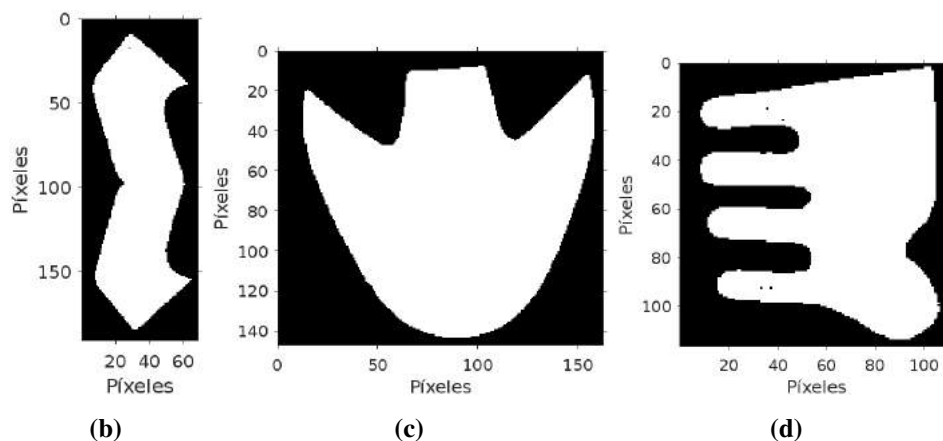
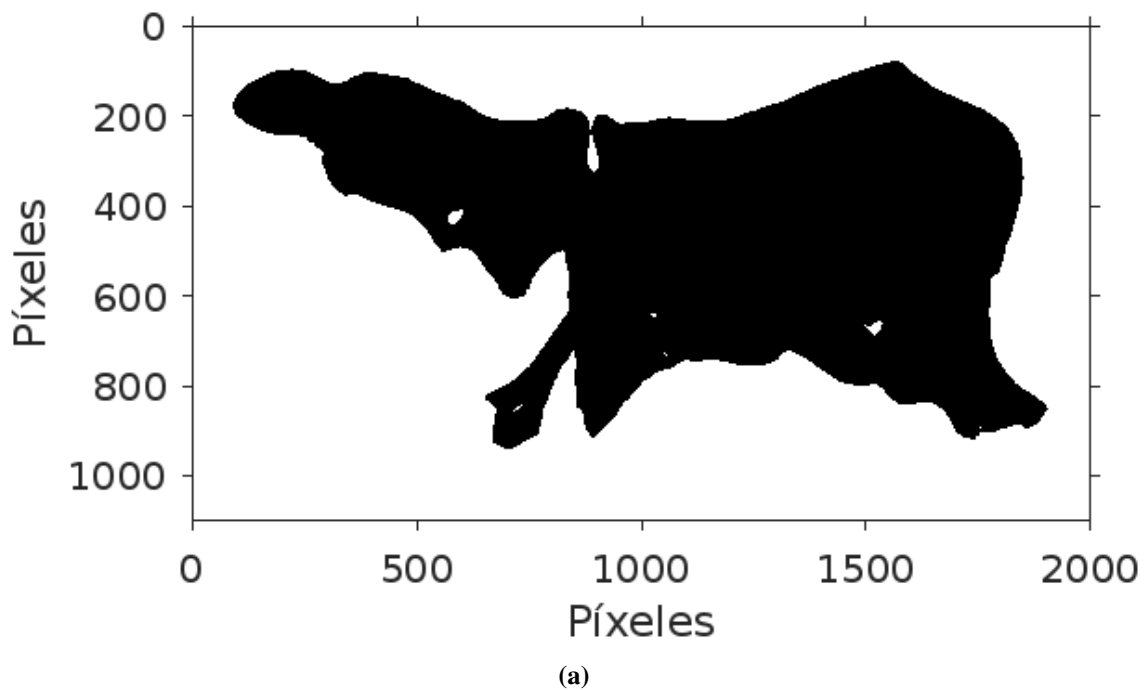


Figura 3.2: Material y patrones binarizados. (a) Muestra de material de acomodo. (b–d) Patrones que son acomodados dentro del material.

Una vez binarizados todos los elementos, se aplica un algoritmo de ordenación de burbujas [39], para ordenarlos de mayor a menor área en píxeles, esto significa que se acomodan primero los patrones con el área más grande, así sucesivamente hasta acomodar los de menor área.

3.2. Proceso de calibración para calcular el desperdicio en píxeles y en metros

Las imágenes del material y los patrones se transforman a partir del tamaño original en metros cuadrados del fotograma utilizando un factor de escala para reducir el material y los patrones a píxeles. Las coordenadas de los píxeles se transforman en coordenadas *mundo* (3D) para una aplicación en el mundo real. Este proceso implica un método de calibración de la cámara [40] y una transformación lineal directa (DLT) [41].

En primer lugar, la cámara se calibra utilizando un conjunto de imágenes de tablero de ajedrez desde diferentes perspectivas. Una imagen de tablero de ajedrez es una cuadrícula en la que se alternan cuadrados blancos y negros. La cuadrícula se utiliza para mostrar el tamaño de los cuadrados en dimensiones métricas y en píxeles. La calibración de la cámara proporciona los parámetros intrínsecos (Ecuación (3.1)), como la distancia focal y el centro óptico, y los parámetros de distorsión (Ecuación (3.2)), como los coeficientes de distorsión radial y tangencial. La calibración de la cámara ayuda a reducir la distorsión en las imágenes. La tabla 3.1 muestra las variables correspondientes a esta sección.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.1)$$

$$d = [k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3]. \quad (3.2)$$

Una vez obtenidos los parámetros de la cámara y los coeficientes de distorsión, la cámara se fija a cierta distancia del material. Esta disposición permite obtener la transformación de coordenadas 3D a coordenadas 2D. Este mapeado se representa como la ecuación (3.3):

$$\mathbf{x}_i = M\mathbf{X}_i, \quad (3.3)$$

donde $M \in \mathbb{R}^{3 \times 4}$ representa la matriz de proyección, $\mathbf{x}_i = [u_i, v_i, 1]$ es la i -ésima coordenada 2D, y $\mathbf{X}_i = [X_i, Y_i, Z_i, 1]$ representa la i -ésima coordenada 3D para $i = 1, 2, \dots, n$, y n es el número de puntos necesarios.

Tabla 3.1: Variables de calibración.

Variables	Descripción
C	Número de imágenes del tablero
f_x, f_y	Distancia focal de los ejes x y y , respectivamente
c_x, c_y	Centro óptico
k_1, k_2, k_3	Distorsión radial
p_1, p_2	Distorsión tangencial
K	Matriz de calibración
d	Vector de distorsión
x_i	i -ésima coordenada 2D (píxeles)
u, v	Coordenadas en píxeles para los ejes x y y , respectivamente
M	Matriz de proyección
m_i	i -ésimos valores de la matriz de proyección
X_i	i -ésima coordenada 3D (<i>mundo</i>)
X_i, Y_i, Z_i	Coordenadas <i>mundo</i> 3D en los ejes x , y y z , respectivamente
$waste_p$	Desperdicio de material en píxeles
$waste_m$	Desperdicio de material en m^2
$area_p$	Área total del material en píxeles
$area_m$	Área total del material en m^2

Se coloca un tablero de ajedrez sobre el material y se captura la disposición. La disposición se muestra en la figura 3.3. Cada esquina del tablero de ajedrez físico coincide con los píxeles correspondientes. El tablero de ajedrez debe cubrir la mayor parte del material para obtener la mejor calibración. La figura 3.4 muestra el arreglo óptico utilizado para tomar las imágenes reales en la fábrica de calzado Tecno boots.

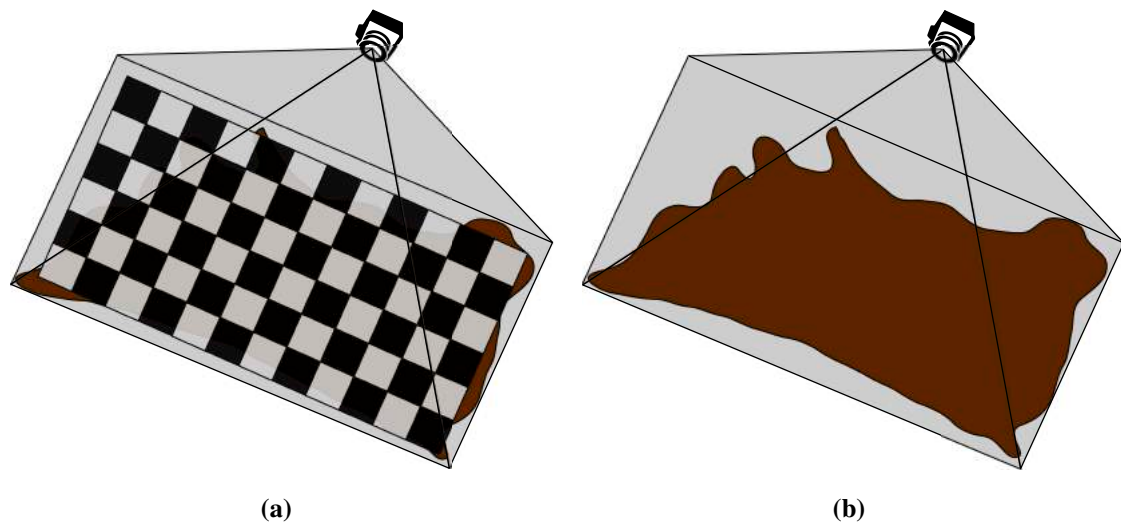


Figura 3.3: Calibración. (a) Muestra un patrón de tablero de ajedrez sobre el material. (b) La cámara captura el material para el proceso de acomodado.

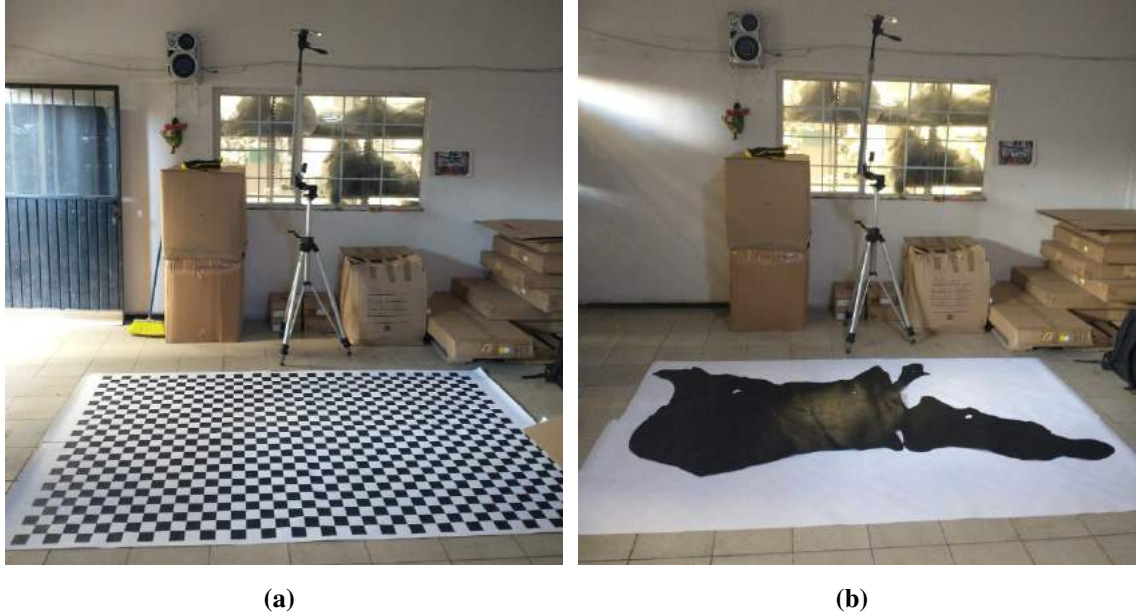


Figura 3.4: Montaje del arreglo óptico para calibración utilizado en la fábrica Tecno boots. (a) Tablero de ajedrez en tamaño real. (b) Captura del material de acomodo en tamaño real.

La matriz de proyección utilizada para transformar las coordenadas 2D en coordenadas 3D se calcula como sigue en la ecuación (3.4):

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1 X_1 & -u_1 Y_1 & -u_1 Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1 X_1 & -v_1 Y_1 & -v_1 Z_1 & -v_1 \\ \vdots & & & & & & & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_n X_n & -u_n Y_n & -u_n Z_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_n X_n & -v_n Y_n & -v_n Z_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ \vdots \\ m_{33} \\ m_{34} \end{bmatrix} = \mathbf{0}. \quad (3.4)$$

Por último, los parámetros m_i se obtienen mediante el método SVD. La ecuación (3.4) se obtiene a partir de la ecuación (3.3), sustituyendo las coordenadas de la imagen en x_i , los coeficientes de la matriz de proyección en M y las coordenadas mundo en X_i como aparece en la ecuación (3.5).

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}, \quad (3.5)$$

Desarrollando la ecuación (3.5) se obtienen las ecuaciones (3.6),(3.7) y (3.8).

$$u_i = m_{11}X_i + m_{12}Y_i + m_{13}Z_i + m_{14}, \quad (3.6)$$

$$v_i = m_{21}X_i + m_{22}Y_i + m_{23}Z_i + m_{24}, \quad (3.7)$$

$$1 = m_{31}X_i + m_{32}Y_i + m_{33}Z_i + m_{34}, \quad (3.8)$$

Luego es necesario dividir las coordenadas u_i y v_i por la ecuación (3.8) que es el factor de escala para normalizar, dando como resultado las ecuaciones (3.9) y (3.10).

$$\frac{u_i}{1} = \frac{m_{11}X_i + m_{12}Y_i + m_{13}Z_i + m_{14}}{m_{31}X_i + m_{32}Y_i + m_{33}Z_i + m_{34}} \quad (3.9)$$

$$\frac{v_i}{1} = \frac{m_{21}X_i + m_{22}Y_i + m_{23}Z_i + m_{24}}{m_{31}X_i + m_{32}Y_i + m_{33}Z_i + m_{34}} \quad (3.10)$$

Finalmente para obtener todos los valores de la matriz de proyección M , se transforman las ecuaciones (3.9) y (3.10) como una ecuación lineal del tipo $Ab = 0$ para llegar a la ecuación (3.4).

El desperdicio se calcula seleccionando un rectángulo alrededor del material. El área del rectángulo se calcula en píxeles cuadrados y metros cuadrados. La relación (r) del desperdicio en píxeles es equivalente a la relación del desperdicio en m^2 , como se muestra en la ecuación (3.11).

$$r = \frac{waste_p}{area_p} = \frac{waste_m}{area_m}, \quad (3.11)$$

donde $waste_m$ es calculado a partir de la ecuación (3.12):

$$waste_m = \frac{waste_p \cdot area_m}{area_p}. \quad (3.12)$$

El desperdicio del material en metros se calcula después que se realiza el acomodo de los patrones en el material.

3.3. Codificación del cromosoma

La codificación del cromosoma está dada por una estructura de datos en la que se almacenan todos los patrones que serán tomados en cuenta para el proceso de acomodo. Cada individuo de la población se codifica como una posible solución al problema de acomodo, la mayoría de estas soluciones se obtienen con traslapes, por lo que es necesario establecer restricciones (Sección 3.3.1) que guíen el proceso de búsqueda desde el inicio del algoritmo. En la tabla 3.2 se describen las variables que se tienen en cuenta en la población inicial.

Tabla 3.2: Variables que caracterizan a los patrones en la población inicial.

Variables	Descripción
<i>Imagen_de_acomodo</i>	Material con los patrones acomodados
<i>Codificación_del_cromosoma</i>	Conjunto de alelos para cada patrón
<i>Fitness</i>	Valor de Aptitud
<i>Patrones_traslapados</i>	Número de patrones que están traslapados
<i>Píxeles_traslapados</i>	Número de píxeles traslapados
<i>Patrones_fuera_área</i>	Número de patrones que están fuera de área
<i>Píxeles_fuera_área</i>	Número de píxeles que están fuera del área de acomodado
<i>Área_ocupada_por_patrones</i>	Número de píxeles de los patrones acomodados.
<i>Patrones_acomodados</i>	Número de patrones acomodados

El tamaño del cromosoma está definido por la cantidad de alelos que representa a cada patrón como se muestra en la figura 3.5.

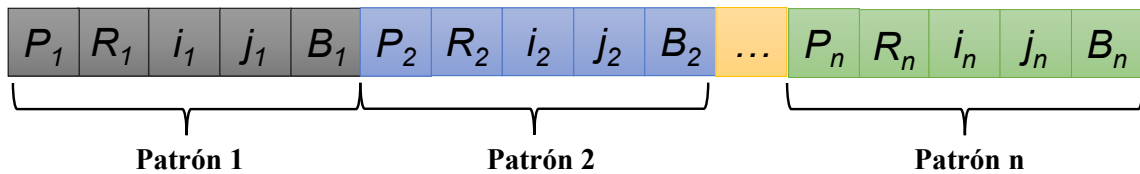


Figura 3.5: La codificación del cromosoma contiene todos los patrones de acomodado. Cada patrón está representado por cinco alelos: identificador (P_k), ángulo de rotación (R_k), posición en filas y columnas dentro del material (i_k, j_k), y el bit de acomodado (B_k).

Cada patrón está representado por cinco alelos como se muestra en la tabla 3.3.

Tabla 3.3: Alelos que identifican a cada patrón.

Variables	Descripción
P_k	Identificador del patrón
R_k	Ángulos de rotación
i_k	Posición de la fila dentro del material
j_k	Posición de la columna dentro del material
B_k	Bit de acomodado

El bit de acomodado (B_k) identifica si el patrón es acomodado o no. Si $B_k = 1$ el patrón es acomodado, si $B_k = 0$ no es acomodado. En nuestro caso, al generar la codificación

del cromosoma, se consideraron todos los alelos B_k con un valor de 1 para cada patrón. El tamaño de la estructura cromosómica es n , que define el número de patrones que son acomodados. El índice k es el número de los patrones colocados en la posición (i_k, j_k) , donde k está en el rango de $1, 2, \dots, n$.

3.3.1. Restricciones en la población inicial

Cuando generamos la población inicial de manera aleatoria, se tienen en cuenta las siguientes restricciones:

1. Las posiciones en filas y columnas deben estar dentro del área válida del material.
2. Se permite un porcentaje de píxeles ($P\%$) fuera del área válida de acomodo (*Pixeles Fuera Área*).

Para definir las posiciones de las filas y columnas de cada elemento, se comprueba que los valores generados aleatoriamente se encuentran dentro del área válida del material, teniendo en cuenta que sólo se permite un porcentaje ($P\%$) de píxeles fuera del área válida, tal y como se muestra en las Ecuaciones (3.13) y (3.14):

$$i_k = \text{fila_material} - (\text{fila_patron} + \text{fila_patron} \cdot (1 - P\%)), \quad (3.13)$$

$$j_k = \text{columna_material} - (\text{columna_patron} + \text{columna_patron} \cdot (1 - P\%)), \quad (3.14)$$

donde *fila_material* y *columna_material* son las posiciones en las filas y columnas dentro del material, respectivamente; *fila_patron* y *columna_patron* son las cantidades en píxeles de filas y columnas de los patrones, respectivamente. Las restricciones permiten guiar la búsqueda dentro del espacio válido del material desde el inicio del algoritmo.

3.4. Método de Reparación

El método de reparación trabaja sobre las soluciones obtenidas a partir del algoritmo genético. Recorre cada patrón que esté traslapado y realiza cuatro movimientos (arriba, abajo, izquierda y derecha). Este proceso se repite hasta que no hayan traslapos entre los patrones. Si el número de píxeles traslapados al principio es igual al número de píxeles traslapados después de realizar los movimientos, el proceso de reparación se detiene, ya que significa que los movimientos generan traslapos con otros patrones y que no hay espacio para mover éste. Si hay patrones traslapados al final del proceso de reparación, éstos son eliminados.

La figura 3.6 muestra un ejemplo de cómo se realizan los movimientos de los patrones. En este ejemplo se tienen 3 patrones para acomodar, en los cuales, el elemento 2 se traslapa con el elemento 3. Los movimientos se realizan de izquierda a derecha: abajo, arriba, derecha e izquierda, y una vez realizados los cuatro movimientos, se calcula el número de píxeles traslapados. Entonces, el movimiento de la izquierda generó cero traslapes; por lo tanto, nos quedamos con esa solución. Si hubiera traslapes en la primera iteración, nos quedaríamos con la solución más pequeña y se repite el proceso hasta que el número de traslapes sea igual a cero.

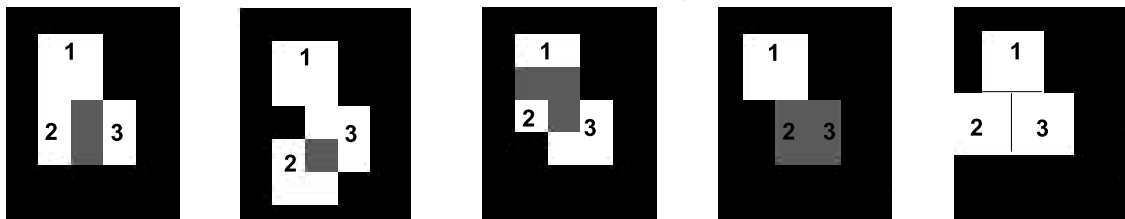


Figura 3.6: Ejemplos de los movimientos de los patrones durante el proceso de reparación. El área negra representa el material de acomodo, el área blanca representa los patrones y el área gris representa los traslapes entre patrones.

3.5. Muéganos

En México, el *muégano* es un dulce que físicamente parece varios trozos de dulces pegados con miel o caramelo. Una vez endurecido, es muy difícil separarlos. El nombre *muégano* también se utiliza con una connotación social para denominar a un conglomerado de dos o más personas unidas por un fuerte vínculo.

3.5.1. Estructura y detección de los *Muéganos*

El concepto de *muéganos* puede aplicarse en el proceso de reparación para que los patrones puedan acomodarse tan pegados como sea posible y disponerse en una estructura. Cada *muégano* representa una lista de elementos que tienen píxeles vecinos adyacentes; esto significa que si dos elementos están unidos por un número limitado de píxeles, entonces se ha formado un *muégano*.

Para detectar los *muéganos* se aplicó el Algoritmo de Relleno por Difusión, el cual recibe una imagen que contiene las soluciones reparadas y cuando encuentra uno o varios píxeles contiguos entre patrones, éstos se etiquetan con un mismo nivel de gris.

Una vez que la imagen contiene los patrones etiquetados con un nivel de gris, se realiza el proceso de selección del mejor *muégano*. Se consideran dos condiciones fundamentales para la selección de un *muégano*:

1. Dos o más patrones tienen píxeles contiguos.
2. Hay un límite de píxeles contiguos entre dos patrones.

Los píxeles contiguos son píxeles adyacentes entre bordes de elementos vecinos. Si hay más píxeles adyacentes entre elementos, hay menos desperdicio.

Se implementó una estructura de datos que contiene la información de cada *muégano*, es decir, el número de patrones que contiene cada *muégano* como se muestra en la Figura 3.7. El número de patrones depende del mayor número de píxeles adyacentes entre los bordes de los elementos vecinos.

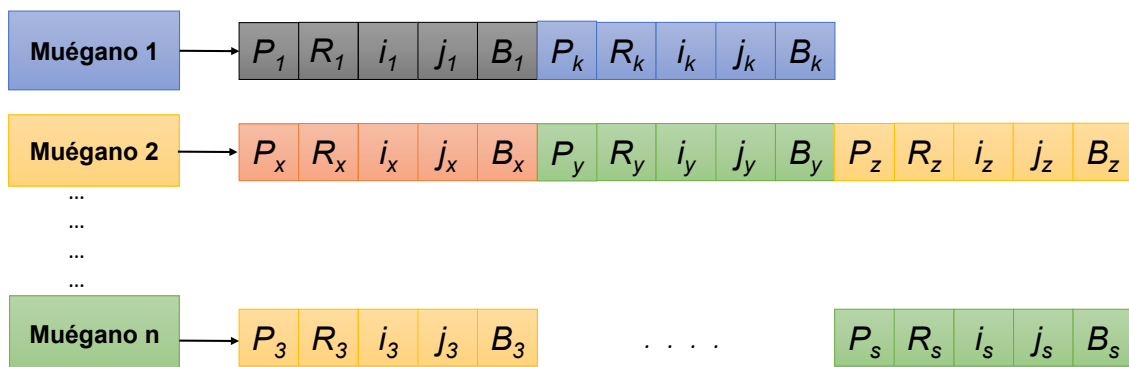


Figura 3.7: Estructura del Muégano. Cada muégano representa una lista de patrones que lo contienen.

Un ejemplo de la formación de un *muégano* se muestra en la figura 3.8.

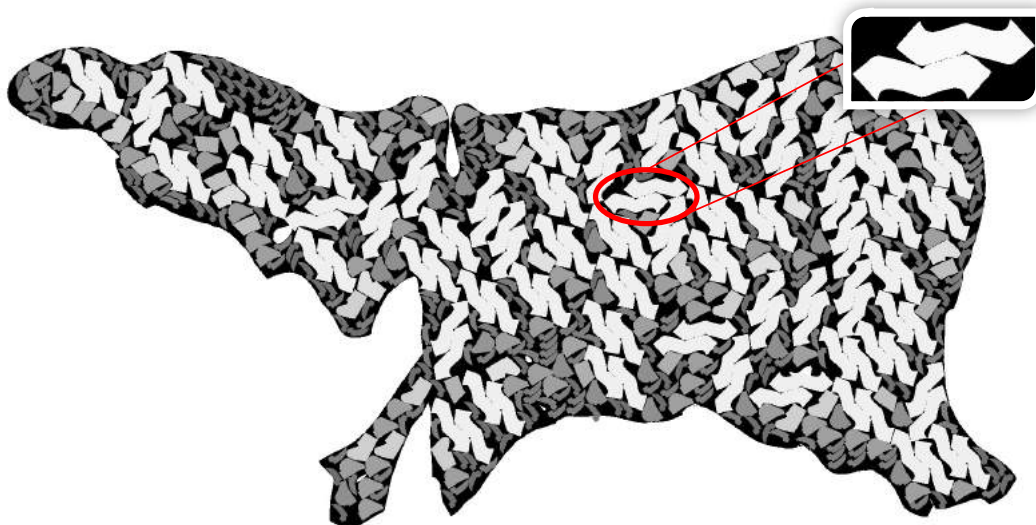


Figura 3.8: Ejemplo de un resultado con patrones y muéganos acomodados. Dentro de la elipse roja: ejemplo de un muégano formado por dos patrones. Cada patrón tiene un nivel de gris diferente.

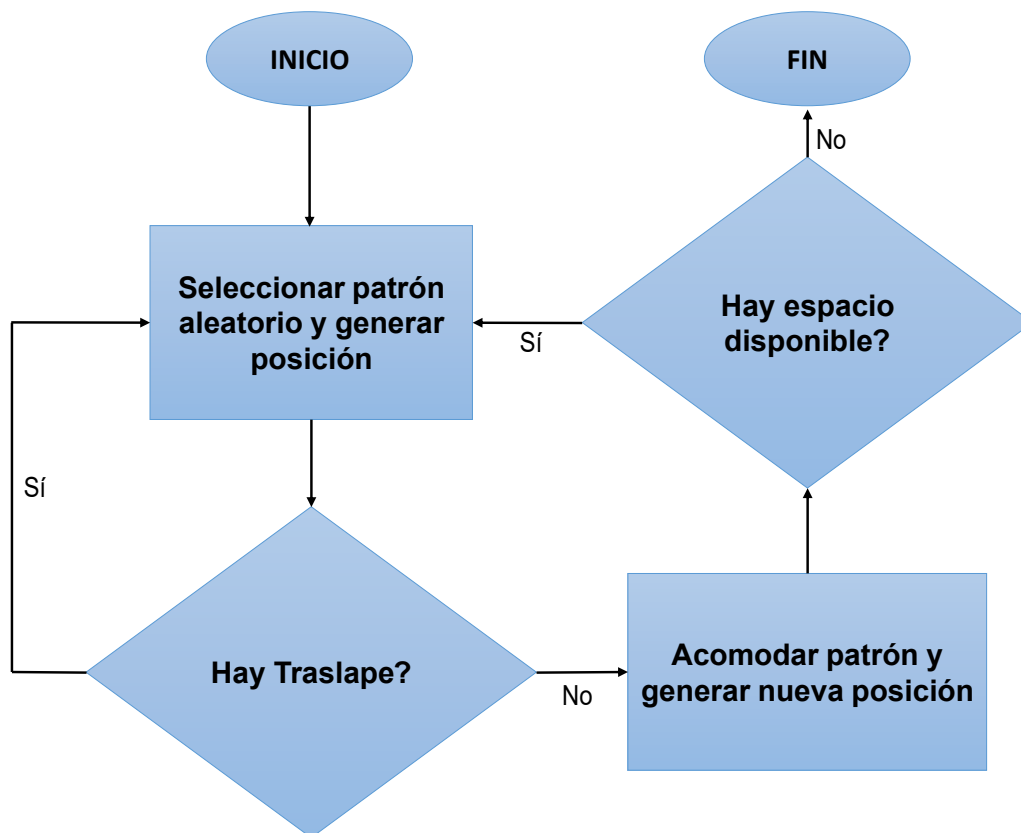


Figura 3.9: Diagrama de flujo del algoritmo goloso secuencial para rellenar el área del material no utilizado.

3.6. Algoritmo Goloso Secuencial

Una vez generados los *muérganos*, se implementa un algoritmo secuencial goloso o codicioso (Sequential Greedy Algorithm (SGA) por sus siglas en inglés) para rellenar el área del material que no se ha utilizado. La figura 3.9 muestra el diagrama de flujo del SGA. Los patrones y *muérganos* se seleccionan aleatoriamente y se genera una posición para acomodarlo, si no hay traslape se realiza el acomodo y el proceso se repite mientras haya espacio disponible en el material.

3.7. Conclusiones Parciales

En este capítulo se describió la metodología empleada para la resolución del problema. Se explicó el proceso de binarización, selección y calibración de las imágenes utilizadas. Se detalló la codificación del cromosoma utilizada en el algoritmo genético, así como las restricciones tomadas en cuenta para crear la población inicial. El proceso de reparación utilizado para evitar los traslapes entre los patrones acomodados. Se describió la estructura

de los *Muéganos* para controlar los patrones más cercanos y repetirlos en las siguientes iteraciones del algoritmo. Y finalmente se explica el algoritmo goloso secuencial que permite rellenar el área del material que no fue utilizado durante el proceso evolutivo.

El siguiente capítulo *Implementación 4* se describe cómo se implementó la interfaz gráfica de usuario, el lenguaje de programación utilizado, el algoritmo genético y sus parámetros para guiar la búsqueda. Y las herramientas empleadas para desarrollar el algoritmo de reparación.

CAPÍTULO 4

IMPLEMENTACIÓN

En el presente capítulo se describe la implementación del algoritmo genético para resolver el problema del acomodo de patrones de corte en materiales finitos. Se explica la codificación del cromosoma, que representan las combinaciones de los patrones en cada generación. Se analiza la función objetivo para evaluar la aptitud de cada cromosoma y los operadores genéticos que se utilizaron en la aplicación.

4.1. Lenguaje de Programación Implementado

Se trabajó en el aprendizaje autodidacta del lenguaje Python con el fin de migrar el software realizado en el trabajo de diploma de Maestría en Optomecatrónica, debido a que Python es un potente lenguaje de programación fácil de aprender. Tiene estructuras de datos eficientes de alto nivel y un enfoque simple pero efectivo para la programación orientada a objetos. La elegante sintaxis y la escritura dinámica de Python, junto con su naturaleza interpretada, lo convierten en un lenguaje ideal para la creación de scripts y el rápido desarrollo de aplicaciones en muchas áreas en la mayoría de las plataformas.

4.2. Interfaz Gráfica de Usuario

Se diseñó una interfaz gráfica de usuario (GUI, por sus siglas en inglés) para seleccionar las imágenes del material de acomodo y de los patrones de corte como muestra la Figura 4.1. Este proceso es el primer paso antes de comenzar el algoritmo genético.

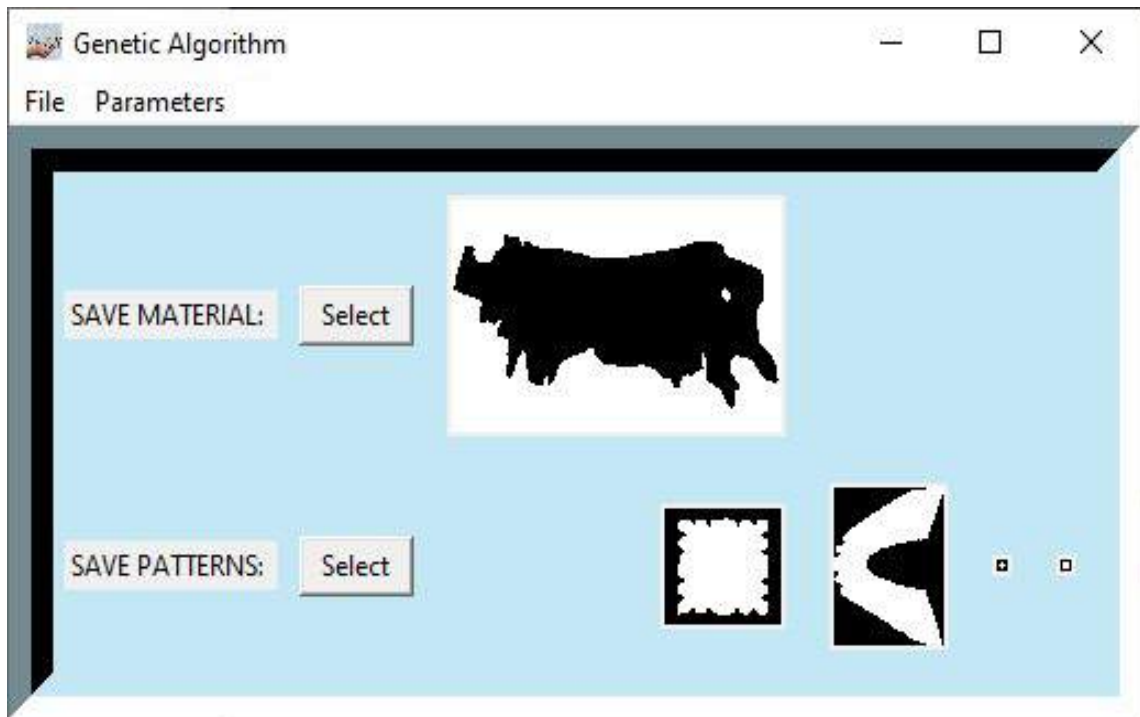
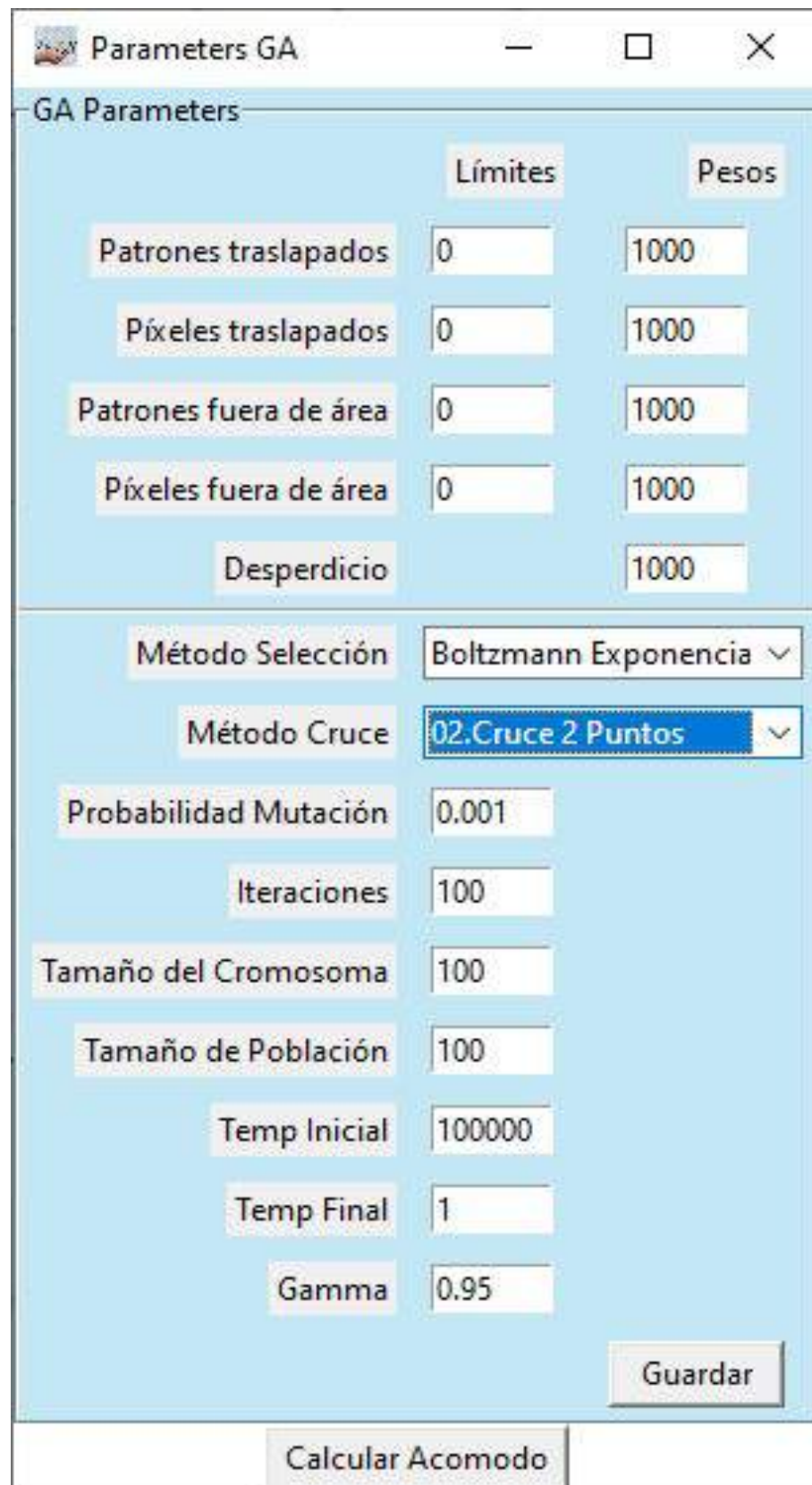


Figura 4.1: Interfaz gráfica de usuario para seleccionar el material de acomodación y los patrones de corte.

4.2.1. Parámetros en la Interfaz Gráfica de Usuario

Una vez que se seleccionan las imágenes, éstas se binarizan de acuerdo a cómo se van a utilizar para el acomodación y se activa en el menú principal de la ventana la entrada de parámetros del algoritmo genético, ver la Figura 4.2. Los parámetros utilizados en el AG se describen en la tabla 4.1. Cuando éstos son seleccionados por el usuario, se guardan y se activa el botón “*Calcular Acomodación*”.



The screenshot shows a window titled "Parameters GA" with standard window controls (minimize, maximize, close). The window is divided into two main sections. The top section, titled "GA Parameters", contains a table of parameters with two columns: "Límites" and "Pesos". The bottom section contains various algorithm parameters, each with a text input field. At the bottom of the window are two buttons: "Guardar" and "Calcular Acomodo".

Parameter	Límites	Pesos
Patrones traslapados	0	1000
Píxeles traslapados	0	1000
Patrones fuera de área	0	1000
Píxeles fuera de área	0	1000
Desperdicio		1000

Método Selección: Boltzmann Exponencia
 Método Cruce: 02.Cruce 2 Puntos
 Probabilidad Mutación: 0.001
 Iteraciones: 100
 Tamaño del Cromosoma: 100
 Tamaño de Población: 100
 Temp Inicial: 100000
 Temp Final: 1
 Gamma: 0.95

Buttons: Guardar, Calcular Acomodo

Figura 4.2: *Parámetros del AG en la GUI.*

Tabla 4.1: *Parámetros utilizados en el AG.*

Variables	Descripción	Valor recomendado
<i>Lim_pat_t</i>	Límite de patrones traslapados	0
<i>Lim_pix_t</i>	Límite de píxeles traslapados	0
<i>Lim_pat_fa</i>	Límite de patrones fuera de área	0
<i>Lim_pix_fa</i>	Límite de píxeles fuera de área	0
<i>P_pat_t</i>	Peso asociado a patrones traslapados	1000
<i>P_pix_t</i>	Peso asociado a píxeles traslapados	1000
<i>P_pat_fa</i>	Peso asociado a patrones fuera de área	1000
<i>P_pix_fa</i>	Peso asociado píxeles fuera de área	1000
<i>Prob_muta</i>	Probabilidad de Mutación	< 0,5
<i>Iteraciones</i>	Número de Iteraciones	> 100
<i>Tam_crom</i>	Tamaño del Cromosoma	> 200
<i>Tam_pob</i>	Tamaño de la Población	> 100
<i>Temp_ini</i>	Temperatura inicial selección Boltzmann	> 9000
<i>Temp_fin</i>	Temperatura final selección Boltzmann	< 1
α	Valor que permite moderar la temperatura en método de selección Boltzmann	$0,5 < \alpha < 0,99$

4.3. Algoritmo Genético

Se implementó un Algoritmo Genético para encontrar una solución global óptima para el problema de acomodo de patrones. El algoritmo consiste en generar soluciones aleatorias que componen una población inicial, a partir de los operadores genéticos hacer evolucionar dichas soluciones y guiar el proceso de búsqueda mediante la función objetivo

para llegar a una solución factible. Este procedimiento se describe en la figura 4.3.

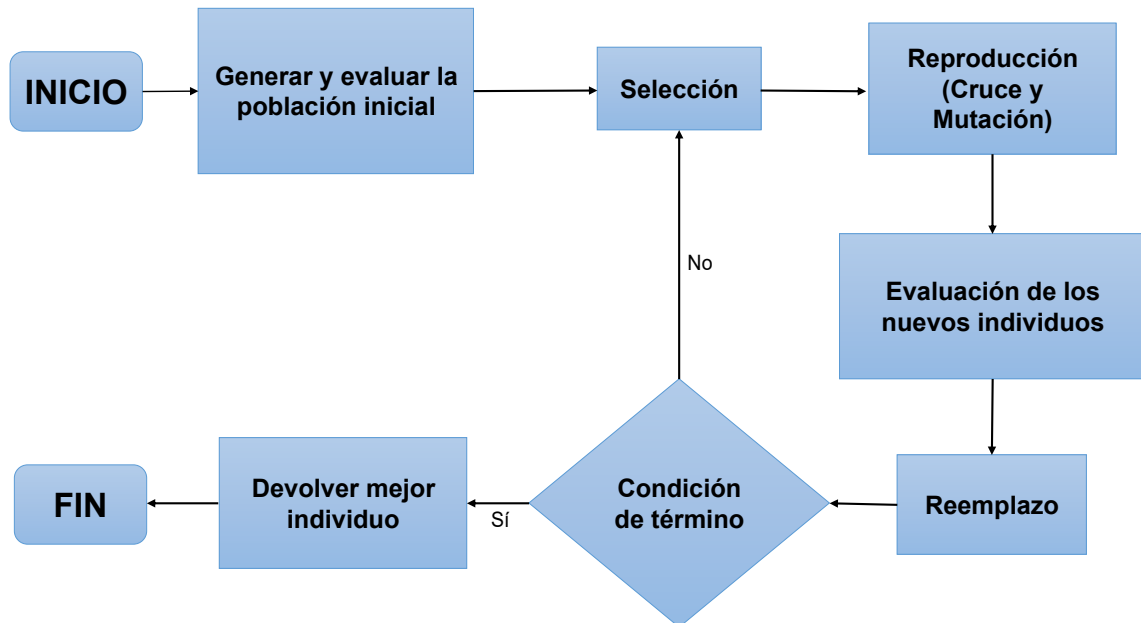


Figura 4.3: Diagrama del Flujo del Algoritmo Genético.

4.4. Población Inicial

La población inicial está compuesta por un conjunto de cromosomas, que son los individuos o soluciones iniciales que se generan aleatoriamente. El tamaño de la población es seleccionado por el usuario en los parámetros iniciales, es recomendable que el tamaño de la población sea mayor a 100.

4.4.1. Codificación del Cromosoma

El cromosoma está compuesto por los patrones que intervienen en el acomodo, como se mostró en la figura 3.5. Su implementación en Python se muestra en la Figura 4.4.

CROMOSOMA

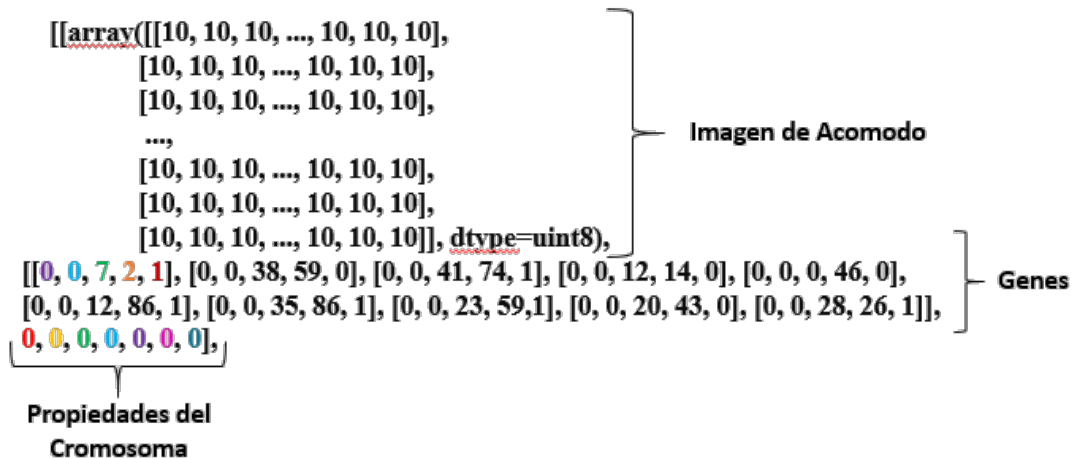


Figura 4.4: Población inicial implementada como una lista en Python.

El cromosoma almacena los valores en píxeles de la imagen binarizada del material. Los genes representan la codificación genética de todos los patrones que son acomodados, en este caso, cada gen contiene 5 alelos que se describen en la Figura 4.5 y los demás parámetros de la población inicial son parte de las propiedades del cromosoma, descritos en la Figura 4.6.

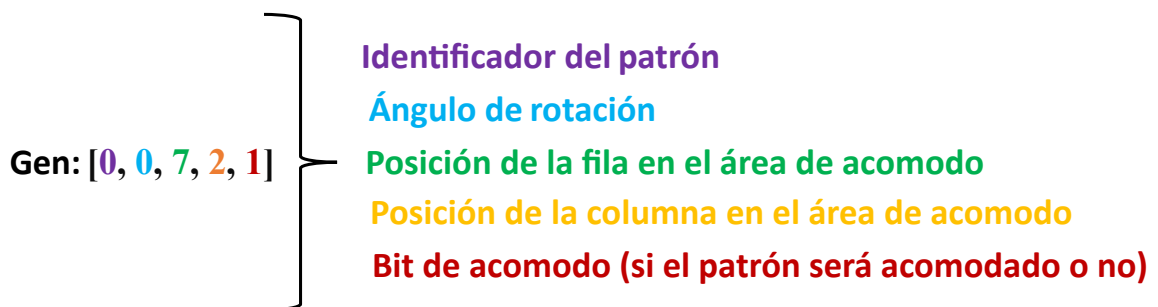


Figura 4.5: Un patrón se codifica como un gen en la codificación del cromosoma y está compuesto por cinco alelos.

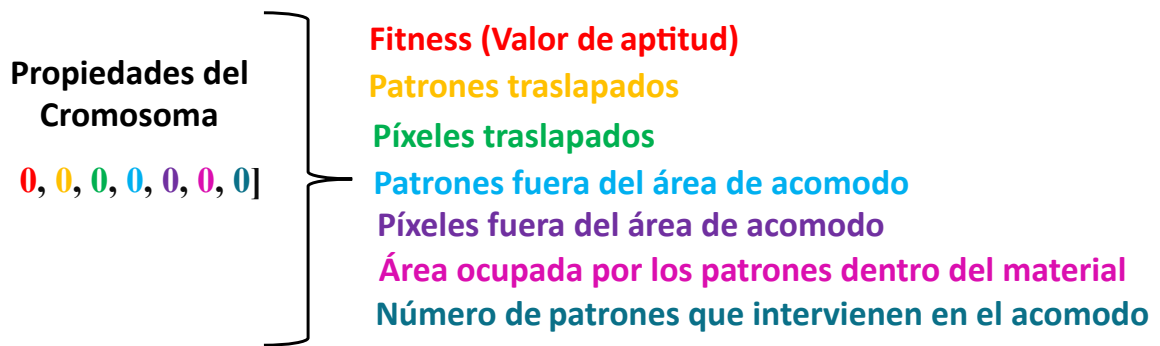


Figura 4.6: Valores que se toman en cuenta en la población inicial para cada solución aleatoria.

4.5. Función de Aptitud

La función o valor de aptitud (Fitness en el idioma Inglés) es de vital importancia para guiar el proceso de búsqueda de las mejores soluciones en cada iteración del AG. En la función de aptitud se manejan los pesos para cada uno de los parámetros de la población inicial y el objetivo es minimizar este valor para obtener el menor desperdicio en el acomodo y al mismo tiempo maximizar el espacio utilizado dentro del material. En el trabajo de Maestría de A. Rodríguez [42] que antecede a la presente investigación, se implementó la ecuación 4.1 como la función de aptitud:

$$F = W_P \cdot (P_T - P_A) + W_w \cdot (A_T - A_{OC}) + W_{NO} \cdot Num_{Ov} + W_{NOA} \cdot Num_{OA} + W_{OP} \cdot Overlaps + W_{OA} \cdot Outside_Area \quad (4.1)$$

donde cada variable se describe en la tabla 4.2

Tabla 4.2: Variables de la función de aptitud utilizadas en el trabajo de Maestría.

Variables	Descripción
W_P	peso asignado a la cantidad de patrones
$(P_T - P_A)$	cantidad de patrones que no son acomodados, se refiere a la diferencia entre la cantidad de patrones que almacena el cromosoma y la cantidad de patrones que son acomodados, mientras menor sea la diferencia significa que hay mayor cantidad de patrones en el acomodo
W_W	peso asignado al desperdicio
$(A_T - A_{OC})$	área de desperdicio, se refiere a la diferencia entre el área total del material y el área ocupada por los patrones
W_{NO}	peso asignado al número de patrones traslapados
Num_{OV}	número de patrones traslapados
W_{NOA}	peso asignado al número de patrones fuera del área válida de acomodo
Num_{OA}	número de patrones que están fuera del área válida de acomodo
W_{OP}	peso asignado al número de píxeles traslapados
$Overlaps$	número de píxeles traslapados
W_{OA}	peso asignado al número de píxeles que están fuera del área válida de acomodo
$Outside_Area$	número de píxeles que están fuera del área válida de acomodo

Con el objetivo de optimizar el código y agilizar la velocidad de convergencia del algoritmo se trabajó en la sensibilidad de los pesos de los parámetros en la función objetivo 4.1. A partir del análisis se llegó a la conclusión que se podía reducir a dos parámetros la función objetivo quedando el ecuación 4.2:

$$F = W_{OP} \cdot Overlaps + W_{OA} \cdot Outside_Area , \quad (4.2)$$

donde solo se tiene en cuenta la cantidad de píxeles traslapados *Overlaps* y los que están fuera del área *Outside Area*, así como sus respectivos pesos.

4.5.1. Sensibilidad de los pesos en la Función Objetivo

Los pesos en la función objetivo se encargan de penalizar cada solución de acuerdo a su adaptación en la población. Estos valores son seleccionados por el usuario en la interfaz gráfica que contiene los parámetros del AG. Un valor elevado en los pesos permite diferenciar entre una buena y una mala solución. Si sus valores son mayores entonces el valor Fitness será mayor, esto significa que es una solución peor, las cuales tiene menor probabilidad de ser seleccionadas en el proceso de selección del AG.

La Figura 4.7 muestra algunas corridas que se realizaron, donde podemos visualizar que a medida que disminuye el peso de los patrones traslapados, se alcanzan resultados no favorables. Para la sensibilidad de los parámetros se utilizó un material de 15x15 píxeles, con un área de acomodo de 169 píxeles. El patrón utilizado ocupa un área de 9 píxeles. Aproximadamente se pueden acomodar 18 patrones dentro del material. Si bien el tamaño del cromosoma es relativamente pequeño, recordemos que puede variar dependiendo del bit de acomodo, sería conveniente duplicar este tamaño para alcanzar mejores resultados. (Se hicieron pruebas con el tamaño del cromosoma en 18 para comprobar el funcionamiento de la función objetivo).

Corridas	Pesos pixeles Traslapados	Pesos Pixeles Fuera_Área	Cantidad de iteraciones	Desperdicio (%)	Patrones acomodados
1	1000	1000	100	73	5
2	1000	1000	100	62	7
3	1000	0	100	73	5
4	1000	1000	100	57	8
5	500	500	100	73	5
6	1000	0	100	68	6
7	600	400	100	78	4
8	1000	0	100	78	4
9	1000	500	100	73	5
10	1000	0	100	78	4
11	900	0	100	68	6
12	800	0	100	62	7
13	700	0	100	78	4
14	600	0	100	62	7
15	500	0	100	78	4
16	400	0	100	68	6
17	300	0	100	62	7
18	200	0	100	84	3
19	100	0	100	84	2

Figura 4.7: Pesos vs Patrones Acomodados utilizando dos parámetros en la función objetivo.
La fila amarilla muestra el mejor resultado, las rojas los peores.

La Figura 4.8 muestra las corridas variando el tamaño del cromosoma y variando los valores del peso que corresponde al desperdicio. Este comportamiento, nos permite observar que el parámetro más importante en la función objetivo es el que se refiere a los píxeles traslapados. Además al aumentar el tamaño del cromosoma se obtienen mejores resultados en comparación con los resultados de la Figura 4.7.

Corridas	Pesos Píxeles Traslapados	Pesos Patrones Fuera Área	Pesos Desperdicio	Iteraciones	Desperdicio (%)	Patrones Acomodados	Tamaño Cromosoma
1	1000	500	500	100	57	8	crom=18
2	1000	500	0	100	78	4	crom=36
3	1000	1000	0	100	73	5	crom=18
4	1000	1000	0	100	62	7	crom=36
5	1000	1000	500	100	57	8	crom=18
6	1000	1000	500	100	62	7	crom=36
7	1000	0	500	100	62	7	crom=18
8	1000	0	500	100	62	7	crom=36
9	1000	0	0	100	73	5	crom=18
10	1000	0	0	100	62	7	crom=36
11	1000	0	1000	100	62	7	crom=18
12	1000	0	1000	100	57	8	crom=36
13	1200	0	1000	100	68	6	crom=18
14	1000	0	1000	100	52	9	crom=36
15	1000	500	1000	100	68	6	crom=100
16	1000	0	1500	100	57	8	crom=18
17	1000	0	1500	100	57	8	crom=36

Figura 4.8: Pesos vs Patrones Acomodados utilizando tres parámetros en la función objetivo. La fila amarilla muestra el mejor resultado, las rojas los peores.

Una buena observación en el caso de este resultado es que sería conveniente ajustar la función de reparación, ya que si se realizan movimientos de los patrones en dependencia de la cantidad de píxeles que tiene traslapados, podríamos obtener mejores resultados después de la reparación. Es conveniente destacar que la variable de los píxeles traslapados es la de mayor importancia en la función objetivo.

4.6. Operadores Genéticos

Los operadores genéticos selección, cruce y mutación permiten generar nuevas poblaciones a lo largo del proceso evolutivo, y que éstas evolucionen al incrementar las iteraciones.

4.6.1. Selección de Boltzmann

La selección de Boltzmann es un método de enfriamiento basado en una función que varía la temperatura, permitiendo que este valor descienda mientras se incrementa el número de iteraciones. A continuación se describen algunas funciones que permite este comportamiento.

- a. **Enfriamiento Lineal:** la ecuación 4.3 permite un descenso de la temperatura de

manera lineal a medida que aumenta el número de iteraciones.

$$T_i = \left(\frac{T_{fin} - T_{ini}}{Iteraciones} \right) \cdot i + T_{ini}, \quad (4.3)$$

donde T_i es la temperatura en la iteración i , T_{ini} y T_{fin} son los valores que el usuario asigna en la configuración inicial a las temperaturas inicial y final respectivamente, $Iteraciones$ es el número de iteraciones.

- b. **Enfriamiento Geométrico:** la ecuación 4.4 permite que la temperatura disminuya de forma gradual a medida que aumenta el número de iteraciones.

$$T_i = \alpha \cdot T_{i-1}, \quad (4.4)$$

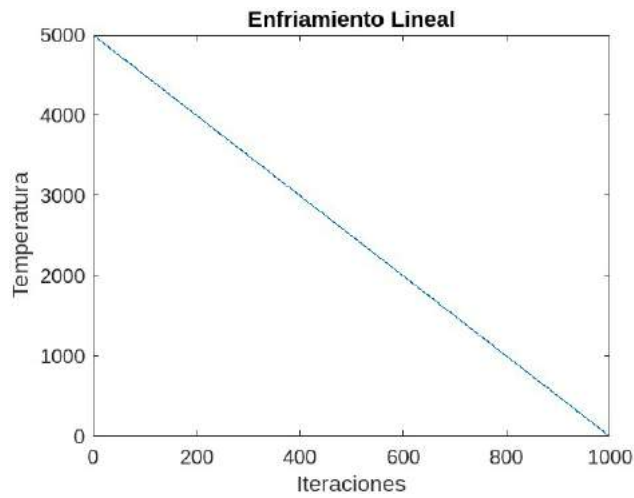
donde T_i es el valor de la temperatura en la iteración i , T_{i-1} es el valor de temperatura en la iteración anterior ($i - 1$) y el valor de α permite moderar la velocidad de descenso de la temperatura, los valores recomendados para α se encuentran entre 0,9 y 0,99.

- c. **Enfriamiento Exponencial:** la ecuación 4.5 permite variar la temperatura de una forma más suave y moderada que con la función geométrica.

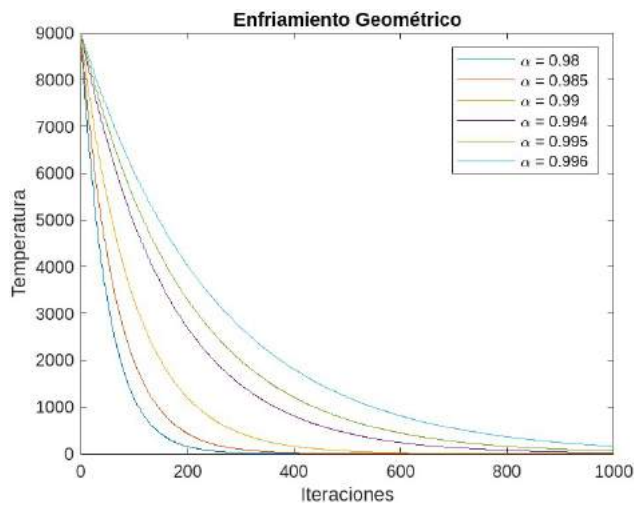
$$T_i = T_{ini} \cdot \left(\sqrt[N]{\frac{T_{fin}}{T_{ini}}} \right)^{i \cdot \alpha}, \quad (4.5)$$

donde T_i es el valor de la temperatura en la iteración i , T_{ini} y T_{fin} son las temperaturas inicial y final respectivamente, N es el número de iteraciones y α es un factor que varía entre 0,5 y 1 que modifica la velocidad de caída de la temperatura.

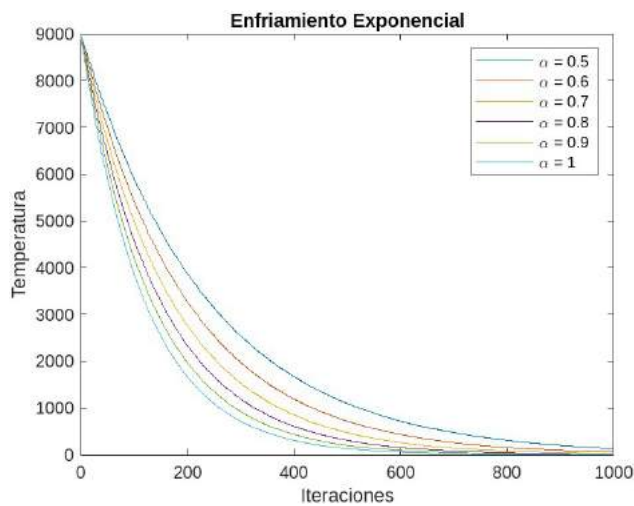
La Figura 4.9 muestra el comportamiento de las ecuaciones anteriores para cada enfriamiento respectivamente. Los valores de estas funciones son seleccionados por el usuario en la interfaz gráfica de los parámetros antes de iniciar el algoritmo.



(a)



(b)



(c)

Figura 4.9: Comportamiento de las funciones de selección de Boltzmann: (a) Enfriamiento Lineal. (b) Enfriamiento Geométrico. (c) Enfriamiento Exponencial.

4.6.2. Cruce en 2 Puntos

El método de cruce menos destructivo es el cruce en 2 puntos, mediante éste es posible heredar mayor información genética de los padres y obtener mejores resultados durante el proceso evolutivo. Para realizar este método se seleccionan de manera aleatoria dos puntos de cruce y dos individuos a los que llamamos padres, y se intercambia entre ellos la información genética entre los dos puntos de cruce para formar dos nuevos individuos llamados hijos, y de esta manera los hijos pasan a formar la población nueva en la siguiente iteración. La Figura 4.10 muestra un ejemplo de cruce entre dos padres.

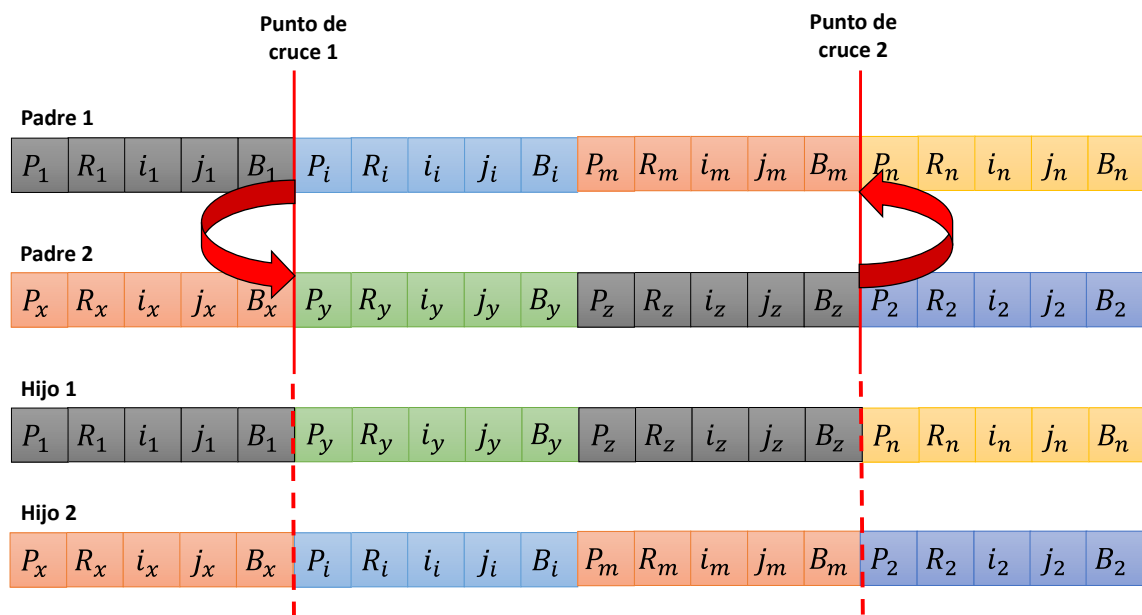


Figura 4.10: Ejemplo de la cruce en 2 puntos

4.6.3. Mutación

El operador genético mutación permite mantener la diversidad bajo una cierta probabilidad seleccionada por el usuario al inicio de la configuración de los parámetros. Este método tiene el propósito de realizar una mejor exploración del espacio de búsqueda. La ecuación logarítmica 4.6 se diseñó para realizar una mutación más efectiva dependiendo de la probabilidad de mutación.

$$t = 1 + \frac{\ln(1-u)}{\ln(1-P_m)} \quad , \quad (4.6)$$

$$\text{alelo_mutado} + = t$$

donde t es el incremento de la posición en el cromosoma que va a mutar, u es un número aleatorio entre 0 y 1, P_m es la probabilidad de mutación seleccionada por el usuario y alelo_mutado es la posición del alelo que va a ser mutado. Éste inicialmente se inicializa

con el valor de t y se va incrementando hasta que sea igual al tamaño del cromosoma, cuando es mayor entonces lo que sobra se utiliza para seleccionar el individuo que sigue.

Es importante destacar que la mutación se puede aplicar tanto al bit de acomodo como al ángulo de rotación de los patrones que conforman el cromosoma. De modo que, bajo la probabilidad de mutación cambiará un determinado porcentaje del cromosoma, o lo que es lo mismo, uno o varios patrones serán mutados, lo que permite una mejor exploración del espacio de búsqueda.

4.7. Reparación del Cromosoma

En la mayoría de los casos el AG genera una solución muy cercana a la óptima que contiene patrones traslapados, por lo que es necesario aplicar una función de reparación que elimine los traslapes. El pseudocódigo de este proceso se muestra en el Algoritmo 1.

4.8. Muérganos

En el capítulo anterior se definieron los muérganos. Para ello se definió una estructura que divide el cromosoma en varios muérganos según el límite de píxeles vecinos contiguos que tengan dos o más patrones. Esta estructura es aplicada durante todo el proceso evolutivo con el objetivo de encontrar los muérganos que ahorren la mayor cantidad de material para disminuir el desperdicio.

La idea de los muérganos es evitar que la cruce y la mutación no sean tan destructivos, ya que los operadores genéticos: cruce y mutación son disruptivos o destructivos, lo que provoca que los individuos en cada generación intercambien información genética. Sin embargo, se rompe el patrón de similaridad en un conjunto de individuos dentro de la población, según la teoría de esquemas. Los muérganos permiten que haya un patrón de similaridad y que éste se repita durante el proceso evolutivo.

El primer paso para encontrar los muérganos es aplicar un algoritmo de relleno por difusión (Flood Fill).

4.8.1. Algoritmo de Relleno por Difusión

El algoritmo de Relleno por difusión (Flood Fill) determina el área formada por elementos contiguos en una matriz. Aplicamos este concepto entre patrones para determinar la cantidad de píxeles contiguos entre dos patrones y formar los muérganos. Cuando el algoritmo encuentra dos o más patrones contiguos, éstos son etiquetados con un nivel de gris diferente. Al final del procedimiento se recorre la imagen que arroja el flood fill y se

Algorithm 1: Método de Reparación

```

Initialization;
Items ← Item_Coding
Repaired ← Items.clone()
for (n : 0 to Size(Items)) do
    if (Repaired.Alleles[n + 4] == 1) then
        overlapping, out_of_area ← false;
        overlapping_count, out_of_area_count ← 0;
        if (Repaired.Overlapping > 0) then
            overlapping_count ← overlapping_count + Repaired.Overlapping;
            overlapping ← true;
        end
    end
    overlaps_new ← 0;
    overlaps_old ← overlapping_count;
    while (overlapping > 0 and overlaps_new ≠ overlaps_old) do
        overlaps_new ← overlaps_old;
        if (overlapping || overlapping_count > 0) then
            Repaired_up ← move_item_up(Repaired);
            up ← calculate_overlaps(Repaired_up);
            Repaired_down ← move_item_down(Repaired);
            down ← calculate_overlaps(Repaired_down);
            Repaired_right ← move_item_right(Repaired);
            right ← calculate_overlaps(Repaired_right);
            Repaired_left ← move_item_left(Repaired);
            left ← calculate_overlaps(Repaired_left);
        end
        overlaps_new ← Calculate_Smallest(up, down, right, left);
        Repaired ← Smallest(overlaps_new);
        if overlaps_new == overlaps_old then
            Break;
        end
    end
end
Result: Repaired

```

forma la estructura de los muérganos, esto significa que se etiqueta en el cromosoma los patrones que pertenecen a los diferentes bloques según el nivel de gris. El procedimiento de relleno se muestra en el Algoritmo 2.

Algorithm 2: Algoritmo de Relleno (Flood Fill)

```

Initialization;
Im ← Image;
x ← PositionX;
y ← PositionY;
new_color ← newColor;
old_color ← oldColor;
pix ← grayLevel;
if (Im[x][y] == pix) then
    Im[x][y] = newColor;
    if (x + 1 < Im.rows()) then
        | FloodFill(Im, x+1, y, newColor, oldColor, grayLevel)
    end
    if (x - 1 > 0) then
        | FloodFill(Im, x-1, y, newColor, oldColor, grayLevel)
    end
    if (y + 1 < Im.cols()) then
        | FloodFill(Im, x, y+1, newColor, oldColor, grayLevel)
    end
    if (y - 1 > 0) then
        | FloodFill(Im, x, y-1, newColor, oldColor, grayLevel)
    end
end
Result: Im

```

4.9. Conclusiones Parciales

En este capítulo se abordó paso a paso el algoritmo implementado para encontrar una solución factible al problema de acomodo de patrones.

Se describió la interfaz gráfica de usuario para seleccionar las imágenes y configurar los parámetros del AG. Se explicó detalladamente los pasos del AG, comenzando con las variables que contiene la población inicial, la codificación del cromosoma, la mejora de la función objetivo a partir del estudio de sensibilidad de los parámetros con el fin de tener en cuenta las variables fundamentales para guiar el proceso de búsqueda de la solución con menor desperdicio. Se explicaron los operadores genéticos fundamentales que fueron

implementados para llevar a cabo el proceso evolutivo. Además se describió la función de reparación para evitar los traslapes entre patrones y el procedimiento para encontrar los muérganos.

En el capítulo *Resultados 5* se muestran los resultados de la investigación variando los parámetros y los operadores genéticos que han sido implementados para alcanzar los resultados satisfactorios.

CAPÍTULO 5

RESULTADOS

En el presente capítulo se muestran los resultados obtenidos en esta investigación, utilizando todos los operadores genéticos implementados. Se realizaron pruebas con imágenes reales y se comparan con resultados de investigaciones previas. El artículo científico de Rodríguez A. et. al. [43] se derivó de la presente investigación y los resultados del artículo son mostrados en esta tesis.

Los resultados que se expondrán fueron obtenidos mediante la aplicación de las técnicas explicadas en el capítulo *Metodología* 3. El término $waste_m$ de la ecuación (3.12) se utilizó para obtener los residuos en metros cuadrados (m^2) de las imágenes resultantes. Se calculó la desviación estándar para medir la precisión según el valor obtenido para los residuos en la misma imagen calculando repetidamente el término $waste_m$.

En esta investigación se utilizaron materiales obtenidos de una fábrica de cuero-calzado, incluyendo material de cuero de vaca y patrones de calzado tipo bota. Estos materiales se procesaron siguiendo la disposición descrita en la sección 3.2, para obtener sus tamaños reales en m^2 y en píxeles cuadrados. Posteriormente, se empleó el procedimiento descrito en el capítulo 3 para generar los resultados que se presentan en esta sección.

Los resultados fueron obtenidos usando 60 *muéganos* en las soluciones. La cantidad de patrones equivalente a dichos *muéganos* es de 120 patrones, que se utilizaron para formar las soluciones en los resultados obtenidos sin *muéganos*. La tabla 5.1 muestra el resultado de veinte ejecuciones del algoritmo sin *muéganos*, mientras que la tabla 5.2 muestra el resultado de veinte ejecuciones del algoritmo con *muéganos*. Cada ejecución representa una solución obtenida del acomodo de los patrones en el material. Las columnas especifican el porcentaje de desperdicio en píxeles y su equivalente en m^2 , el tiempo de ejecución en segundos (s) y las orientaciones admisibles para cada patrón en grados

(°).

Tabla 5.1: Resultados obtenidos sin Muérganos.

Ejecuciones	Desperdicio (%)	Desperdicio (m^2)	Tiempo (s)	Ángulos de rotación
1	31,2	0,4919	303,44	0°, 120°, 240°
2	30,77	0,4850	256,16	0°, 120°, 240°
3	30,58	0,4819	319,21	0°, 120°, 240°
4	30,58	0,4822	273,84	0°, 120°, 240°
5	30,98	0,4883	250,34	0°, 120°, 240°
6	31,29	0,4932	250,85	0°, 120°, 240°
7	30,62	0,4825	277,82	0°, 120°, 240°
8	30,87	0,4866	260,68	0°, 120°, 240°
9	31,1	0,4902	278,54	0°, 120°, 240°
10	30,27	0,4771	270,62	0°, 120°, 240°
11	31,14	0,4911	258,51	0°, 120°, 240°
12	30,64	0,4832	250,64	0°, 120°, 240°
13	30,94	0,4878	295,93	0°, 120°, 240°
14	30,4	0,4794	261,95	0°, 120°, 240°
15	30,65	0,4833	285,84	0°, 120°, 240°
16	31,58	0,4978	252,58	0°, 120°, 240°
17	30,74	0,4847	294,69	0°, 120°, 240°
18	30,93	0,4875	280,09	0°, 120°, 240°
19	31,1	0,4904	319,71	0°, 120°, 240°
20	30,69	0,4835	329,71	0°, 120°, 240°
<i>Average</i>	30,85	0,4863	278,55	

Tabla 5.2: Resultados obtenidos con Muérganos.

Ejecuciones	Desperdicio (%)	Desperdicio (m^2)	Tiempo (s)	Ángulos de rotación
1	28.64	0.4513	339.52	0°, 120°, 240°
2	28.73	0.4528	380.47	0°, 120°, 240°
3	29.36	0.4628	290	0°, 120°, 240°
4	29.52	0.4654	275.48	0°, 120°, 240°
5	29.21	0.4605	264.16	0°, 120°, 240°
6	28.82	0.4593	315.52	0°, 120°, 240°
7	28.4	0.4480	321.24	0°, 120°, 240°
8	28.76	0.4536	306.85	0°, 120°, 240°
9	28.63	0.4515	282.33	0°, 120°, 240°
10	28.49	0.4494	359.65	0°, 120°, 240°
11	28.89	0.4556	290.64	0°, 120°, 240°
12	28.9	0.4554	303.83	0°, 120°, 240°
13	28.99	0.4567	268.73	0°, 120°, 240°
14	29.23	0.4607	274.65	0°, 120°, 240°
15	29.2	0.4602	318.56	0°, 120°, 240°
16	28.5	0.4492	291.57	0°, 120°, 240°
17	28.79	0.4537	282.16	0°, 120°, 240°
18	29.11	0.4588	305.44	0°, 120°, 240°
19	29.21	0.4605	308.56	0°, 120°, 240°
20	28.8	0.4539	324.28	0°, 120°, 240°
<i>Average</i>	28,909	0,4559	305,82	

La figura 5.1 representa una comparación del valor de desperdicio entre las soluciones con y sin *muérganos*. Es importante destacar que la peor solución con *muérganos* consigue un desperdicio menor que la mejor solución sin ellos. Este método propuesto mejora la solución y reduce el desperdicio de material.

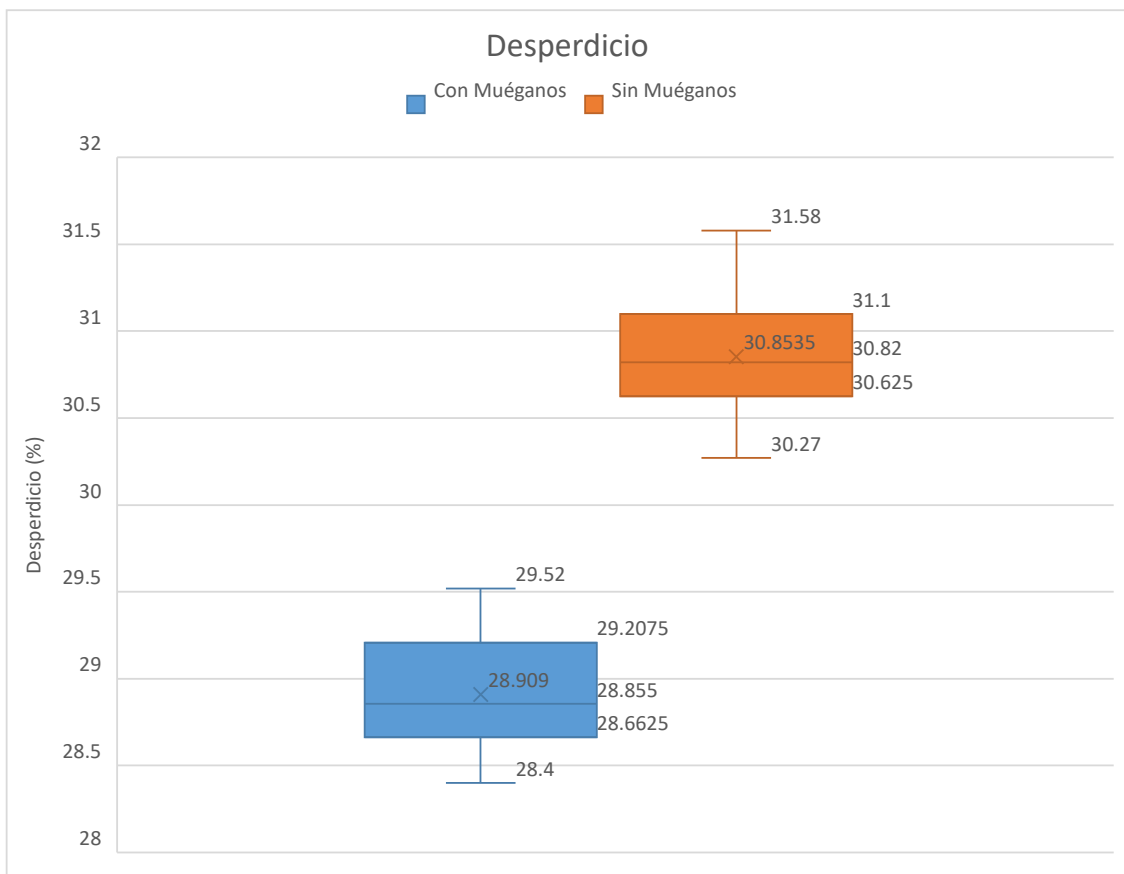


Figura 5.1: Comparación del desperdicio entre los resultados obtenidos con y sin muérganos. El diagrama de cajas y bigotes azules muestra una mejora del desperdicio obtenido con respecto al diagrama de cajas y bigotes naranja sin muérganos. La media de los resultados con muérganos es de 28,9%.

La figura 5.2 muestra el tiempo de ejecución, comparando las soluciones con y sin muérganos. Al estar formados por elementos más compactos, se tarda más en colocarlos en el material, ya que se requiere de un espacio mayor para realizar el acomodo de los muérganos. Sin embargo, el aumento del tiempo compensa el hecho de que se reduce el desperdicio.

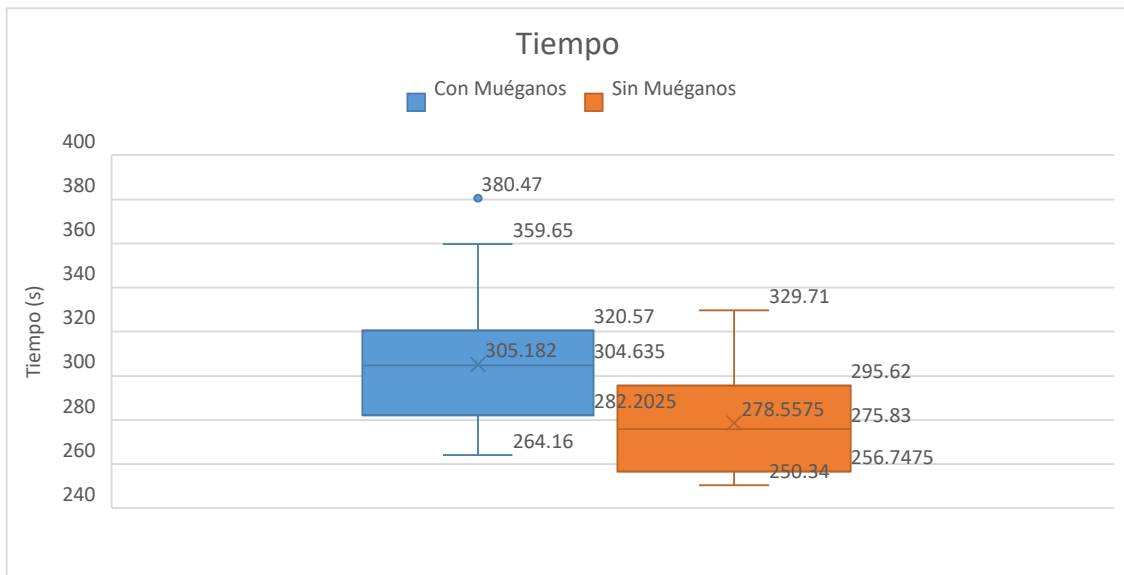
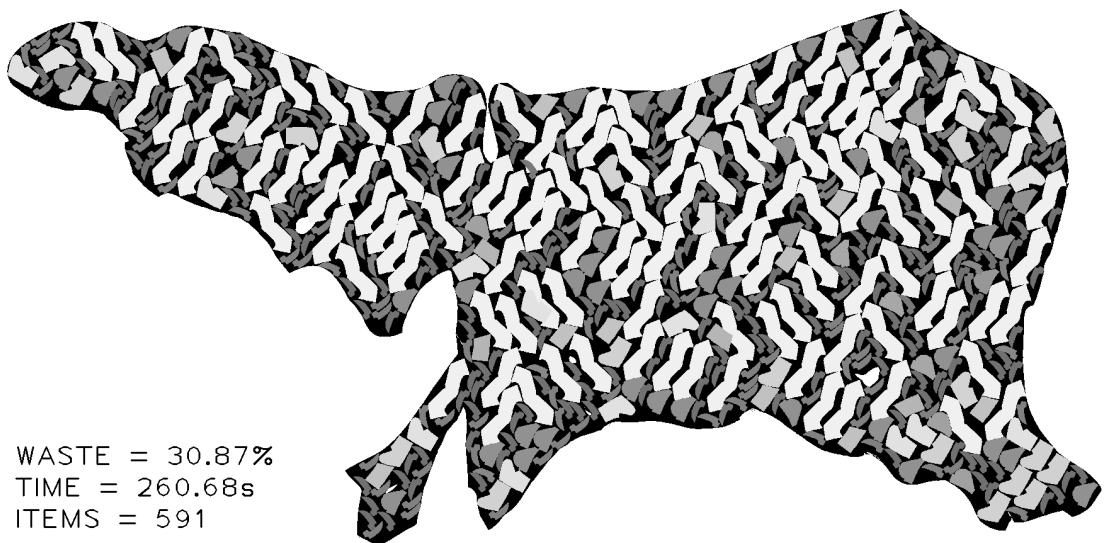
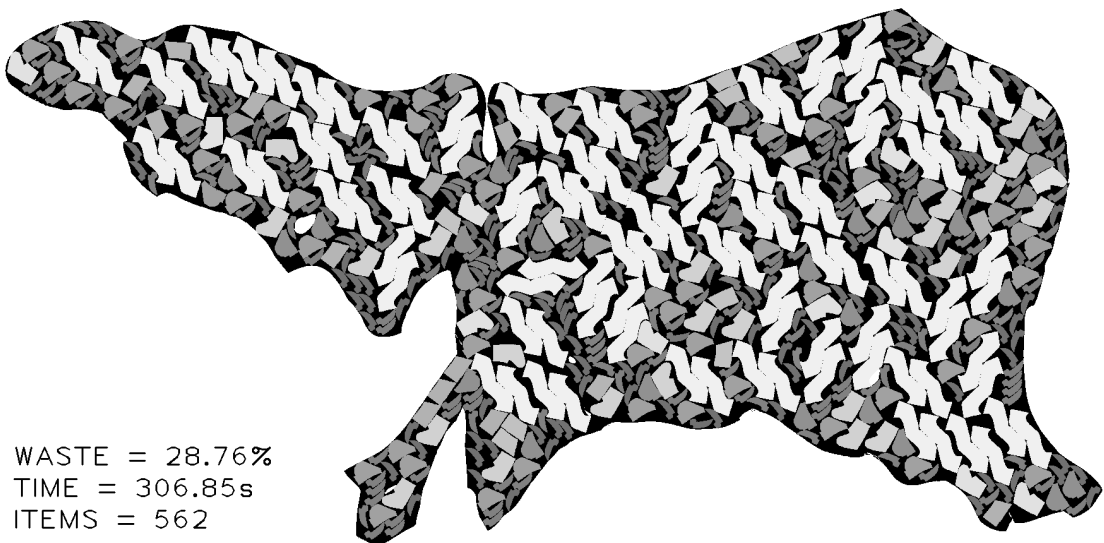


Figura 5.2: Comparación del tiempo entre los resultados obtenidos con y sin *muéganos*. El diagrama de cajas y bigotes azules muestra una mayor dispersión del tiempo de cálculo, y la media de los resultados es de 305 s debido al tamaño de la estructura *muéganos*. Sin embargo, se compensa con un desperdicio menor.

La figura 5.3 muestra los resultados obtenidos con y sin *muéganos*, respectivamente. La figura 5.3a muestra los patrones acomodados aleatoriamente, lo que provoca que al anidarlos haya una menor compactación entre ellos. El tiempo de ejecución de la solución fue de 260,68 s. La figura 5.3b ilustra una mayor compactación entre dos patrones contenidos en el *muécano*, con un tiempo de ejecución de 306,85s.



(a)



(b)

Figura 5.3: Resultado del acomodo (a) sin muérganos y (b) con muérganos. El resultado obtenido con muérganos genera menos desperdicio que la solución sin muérganos debido a la mínima separación entre patrones.

5.1. Comparaciones con otras investigaciones

Se realizaron algunas comparaciones con otros métodos propuestos en la literatura. En [6], utilizaron un algoritmo codicioso con una estrategia de abajo a la izquierda para colocar los elementos obteniendo una eficiencia de 81,54%. En nuestro trabajo se aplicó el concepto de *muérganos* para encontrar la solución obtenida con una eficiencia de

82,11 % como se muestra en la Figura 5.4a.

En [18], se desarrolló un algoritmo de separación y compactación utilizando un mapa de obstrucción rápido de búsqueda local (FOMLS) para resolver un problema de empaquetamiento rectangular (RPP) utilizando compactación de relación fija. Obtuvieron una eficiencia de 68,31 %, que, comparado con el resultado de esta investigación utilizando los *muéganos*, se obtuvo una mejora con una eficiencia de 69,09 %. La figura 5.4b muestra nuestro resultado.

En [30], calculando con BRKGA70 para las instancias con círculos se obtuvo una eficiencia de 73,32 %. En nuestro caso, realizamos algunas ejecuciones con los mismos patrones sobre un material finito obteniendo una eficiencia de 74,65 %, como se muestra en la Figura 5.4c.

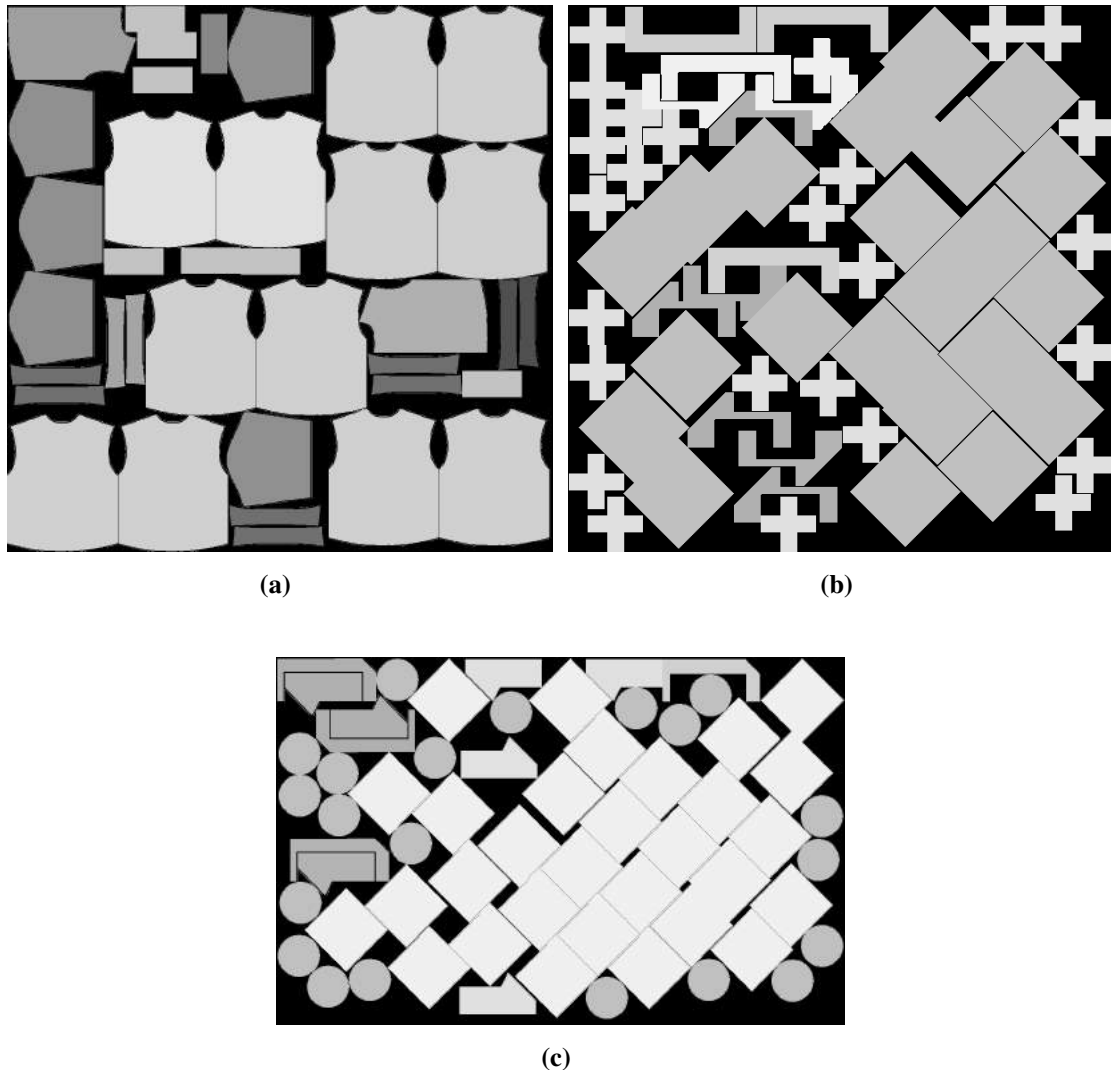


Figura 5.4: Resultados obtenidos con el método de reparación utilizando la forma de los patrones de otras investigaciones. Usamos la forma de los patrones (a) del artículo de Fischetti y Luzzi, (b) del artículo de Sato et al., y (c) del artículo de Mundim et al. para comparar el desperdicio y la eficiencia.

La tabla 5.3 muestra los resultados comparativos de nuestro trabajo frente a otros métodos. Las tres primeras columnas presentan referencias y los valores de eficiencia y residuos obtenidos a partir de los métodos citados. Las dos últimas columnas exponen los resultados del método propuesto en este estudio, obtenidos mediante la utilización de patrones de los métodos comparados.

Tabla 5.3: Resultados comparativos entre esta investigación y otras revisadas en el estado del arte.

Método y Referencia	Desperdicio (%)	Eficiencia (%)	Desperdicio (%) Método de reparación	Eficiencia (%) Método de reparación
MIP [6]	18,46	81,54	17,89	82.11
BRKGA70 [30]	26,68	73,32	25,35	74.65
FOMLS—RPP [18]	31,68	68,31	30,91	69.09

5.2. Conclusiones Parciales

En este capítulo se realizó una comparación entre los resultados obtenidos de la presente investigación con investigaciones anteriormente propuestas, demostrando que el método que se implementó es eficaz y arroja soluciones favorables para la solución del problema de acomodo de patrones de corte.

CAPÍTULO 6

CONCLUSIONES Y TRABAJOS A FUTURO

En esta investigación se resuelve el problema de acomodo de patrones en materiales finitos utilizando un algoritmo genético y una estrategia golosa para reparar soluciones obtenidas con traslapes, lo que proporciona una excelente alternativa para resolver el problema de acomodo. La estructura *muéganos* compacta los patrones para reducir el desperdicio, y se demuestra que se obtienen soluciones factibles que compiten con los resultados obtenidos en investigaciones anteriores. Es importante mencionar que el proceso de reparación y los *muéganos* se pueden aplicar a cualquier solución obtenido por alguna metaheurística utilizada para resolver el problema de anidamiento, siempre y cuando existan traslapes en cualquier solución.

El método propuesto en esta investigación se aplica a problemas reales en industrias como la del cuero-calzado, donde la reducción de desperdicio es crucial para una utilización óptima del material. Muchas empresas todavía llevan a cabo manualmente los procesos de anidamiento, dependiendo de la pericia de los operarios, lo que repercute en el desperdicio de tiempo y material. La automatización del proceso de anidado es vital para la eficiencia. Empresas especializadas nos han proporcionado los materiales necesarios (incluido material de cuero y patrones de calzado) para realizar pruebas a tamaño real. Además, establecimos un arreglo óptico dentro de la fábrica para capturar imágenes de los materiales, facilitando los procesos posteriores de calibración y digitalización de imágenes.

El método propuesto ofrece una ventaja significativa a la hora de automatizar el proceso de acomodo en estas industrias, lo que se traduce en una disminución del desperdicio

de material y un aumento de la eficiencia. A pesar de que este método presenta ventajas frente a otros reportados, tiene ciertas limitaciones, pero trabajar sobre ellas puede conducir a resultados aún mejores. Por ejemplo, el tiempo de producción es un factor importante en la industria cuando se trabaja en tiempo real, y esta mejora podría hacer que el algoritmo fuera más eficiente. En futuros trabajos, se mejorará el factor tiempo utilizando programación paralela con hilos en la implementación del algoritmo. Además, es conveniente llevar un recuento de todos los patrones necesarios para completar el proceso de fabricación dependiendo del tipo de industria en la que se aplique el algoritmo.

Para mejorar aún más los resultados, es necesario trabajar en la determinación de la calidad de la piel para dar un posicionamiento adecuado a cada patrón. La segmentación de la piel es por tipo de corte, ya que cada artículo debe tener una cierta elasticidad y, por lo tanto, debe colocarse en un ángulo de orientación adecuado. Otro trabajo futuro es calcular la eficiencia del desperdicio por *muéganos* para dar soluciones con el menor desperdicio posible. Por último, otra mejora será conseguir la formación de *muéganos* por unidad de producción; por ejemplo, si un zapato necesita cinco patrones para su fabricación, entonces los *muéganos* deberán contener estos cinco patrones.

APÉNDICE A

INVESTIGACIÓN EN EL PLANO COMERCIAL

A.1. Datos reales tomados de una fábrica de calzado.

La presente investigación se realizó teniendo en cuenta tanto el plano científico como el comercial, para ellos se visitaron fábricas de calzado con el objetivo de obtener experiencia y ajustar la implementación del algoritmo según datos reales.

A partir de un encuentro con el Jefe de Producción de una fábrica de Calzado: Calzado Gramado, el señor Luis Francisco Ramírez, explicó cómo realizan el proceso de acomodo, que lo realizan de forma manual, y calculan cuánto material ocupan para unos 24 pares de calzado de diferente numeración y luego calculan el desperdicio que les es permitido (5 a 10%).

La Ecuación A.1 calcula el consumo que se refiere a la cantidad de material que ocupa un patrón.

$$\text{Consumo} = \text{Ancho}_{\text{patron}} \cdot \text{Largo}_{\text{patron}} , \quad (\text{A.1})$$

donde $\text{Ancho}_{\text{patron}}$ y $\text{Largo}_{\text{patron}}$ representan el ancho y largo del patrón en píxeles respectivamente.

La Ecuación A.2 calcula el número de piezas que se pueden acomodar dentro del material.

$$\text{Piezas} = \frac{\text{Área}_{\text{material}}}{\text{Consumo}} , \quad (\text{A.2})$$

donde $\text{Área}_{\text{material}}$ es el área del material en píxeles y Consumo el área del patrón en píxeles.

Estos cálculos fueron implementados en la investigación para calcular el tamaño aproximado del cromosoma una vez que se seleccionan los patrones y el material de acomodo,

valor que se sugiere en la ventana de los parámetros en la interfaz gráfica de usuario.

Las Figuras A.1 y A.2 muestran los patrones que se utilizaron en los resultados experimentales, gracias a las fábricas de calzado **Tecnoboots** y **Lobosolo** por su aporte de material.

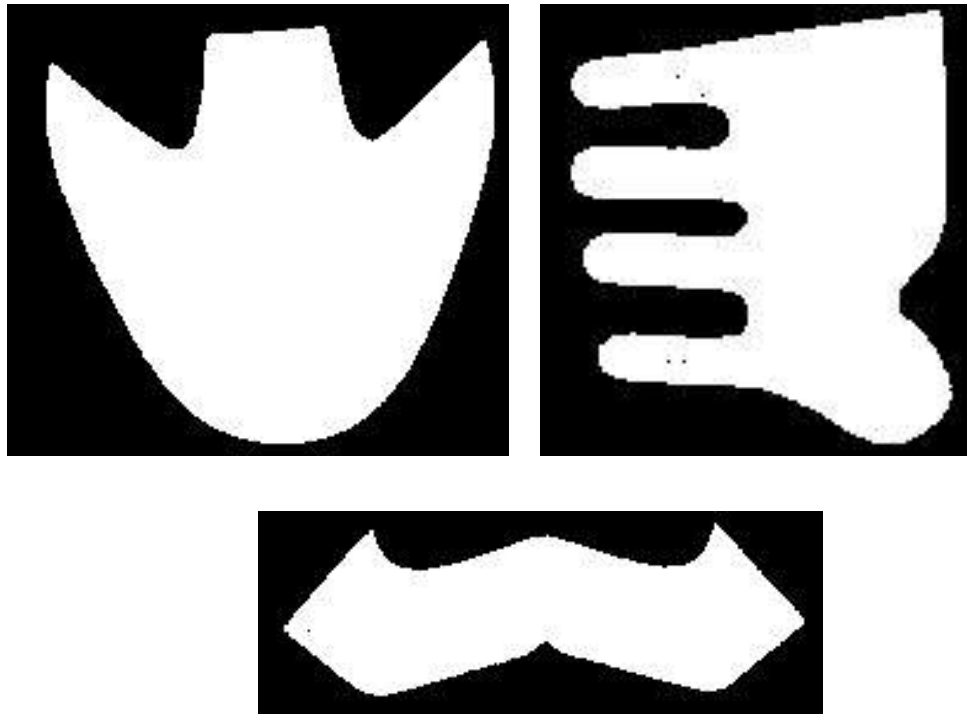


Figura A.1: Patrones reales de piel para botas de adultos después del proceso de binarización.



Figura A.2: Patrones reales de calzado de niño después del proceso de binarización.

APÉNDICE B

ALGORITMO DE RELLENO (FLOOD FILL)

```
import numpy as np

def FloodFill_Recursivo(img, x , y,new_color , old_color):
    rows, cols = img.shape
    if(img[x][y] == 245):
        img[x][y] = new_color
        if(x+1 < rows):
            img = FloodFill_Recursivo(img,x+1,y,new_color , old_color)
        if(x-1 > 0):
            img = FloodFill_Recursivo(img,x-1,y,new_color , old_color)
        if(y+1 < cols):
            img = FloodFill_Recursivo(img,x,y+1,new_color , old_color)
        if(y-1 > 0):
            img = FloodFill_Recursivo(img,x,y-1,new_color , old_color)

    return img
```

APÉNDICE C

ALGORITMO DE ETIQUETADO DEL CROMOSOMA

```
import numpy as np

def Etiquetar_Bloques(crom, img):
    etiquetas = []
    indices = []
    for i in range(Tam_cromosoma):
        if(crom[1][i][4] == 1):
            cx = lista_patrones[crom[1][i][0]][4]
            cy = lista_patrones[crom[1][i][0]][5]
            rows = (crom[1][i][2] + cx)
            cols = (crom[1][i][3] + cy)
            nivel_gris = img[rows][cols]
            indices.append(i)
            etiquetas.append(nivel_gris)

    return indices, etiquetas
```

APÉNDICE D

ALGORITMO DE MUÉGANOS

```
import numpy as np

def Crear_Muegano(genes , tam):
    im_mueg = np.zeros((50,50))
    for i in range(len(genes)):
        im = lista_patrones[genes[i][0]][0]
        print(im)
        rows,cols = im.shape
        k = 0
        l = 0
        for n in range(rows):
            for m in range(cols):
                im_mueg[k][l] = im[n][m]
                l += 1
            k+=1
    print("Muegano", im_mueg)
    return im_mueg
```

APÉNDICE E

ALGORITMO DE REPARACIÓN

```
import numpy as np

def Traslapes_FueraArea(pob):
    #print("POB_INI = ", pob[0])
    k = 0
    l = 0
    m = 0
    p = 0
    traslape = 0
    fuera_area = 0
    Area_patrones = 0
    tr = 0
    fa = 0
    pos = 0
    cant_pat = 0
    array = np.zeros(6, dtype=int)
    global strut_pat_traslapados
    global num_rotaciones
    global num_patrones

    for n in range(Tam_cromosoma):
        cant_ang = 0
        if pob[1][n][4] == 1:
            cant_pat += 1
            t = False
```

```
f = False
angulo = pob[1][n][1]
pos = int((angulo * num_rotaciones)/360 + (num_rotaciones
rows, cols = patrones_rotados[pos][0].shape
greyMaterial3 = pob[0]
k = pob[1][n][2]
for i in range(rows):
    l = pob[1][n][3]
    for j in range(cols):
        m = int(pob[0][k][1])
        p = int(patrones_rotados[pos][0][i][j])
        if m == 0 and p == 1:
            Area_patrones +=1
        elif m >= 1 and m < 10 and p == 1:
            traslape +=1
            t = True
        elif m >= 10 and p == 1:
            fuera_area +=1
            f = True
        pob[0][k][1] = int(m + p)
    l+=1
    k+=1
if t:
    tr +=1
if f:
    fa +=1

array[0] = traslape
array[1] = fuera_area
array[2] = tr
array[3] = fa
array[4] = Area_patrones
array[5] = cant_pat

return array
```

APÉNDICE F

APORTE CIENTÍFICO: CONGRESOS, EVENTOS Y PUBLICACIONES

Participación en el *Congreso Riao / Optilas / MOPM 2019, X Iberoamerican Optics Meeting, XIII Latinamerican Meeting on Optics, Lasers and Applications/ Mexican Optics and Photonics Meeting*, con el trabajo presentado en forma de póster: **“Genetic Algorithm applied to the arrangement of cutting patterns in finite materials”**. La constancia se muestra en la figura F.1.



Figura F.1: Certificado de Participación en el Congreso Riao / Optilas / MOPM 2019.

Derivado del trabajo de Maestría en Optomecatrónica de A. Rodríguez [42] se presentó el trabajo: “*A Genetic Algorithm Applied to the Nesting of Cutting Patterns in Finite Materials*”, en el VII Congreso Internacional de Robótica y Computación (CIRC 2020), cuyo certificado se muestra en la figura F.2.

Se publicó el artículo “*A Genetic Algorithm Applied to the Nesting of Cutting Patterns in Finite Materials*”, en el libro especializado del Congreso: “*Robótica y Computación. Nuevos Avances*”, Primera edición, Mayo de 2020. ISBN: 978-607-98174-6-6. Página 135.



Figura F.2: Certificado de Participación en el VII Congreso Internacional de Robótica y Computación (CIRC 2020).

Participación el Congreso “*Imaging and Applied Optics Congress & Optical Sensor and Sensing Congress 2020*”, este evento se realizó de manera virtual, la constancia se muestra en la figura F.3.



Figura F.3: *Certificado de Imaging and Applied Optics Congress & Optical Sensor and Sensing Congress 2020.*

Participación en el Foro de Inteligencia Artificial: “*Aplicaciones de Inteligencia Artificial para proyectos de impacto social*”, realizado en Abril de 2021 por la Universidad Nacional Autónoma de México, la constancia se muestra en la figura F.4.



Figura F.4: Constancia de participación en el Foro de Inteligencia Artificial 2021.

Participación en el Congreso “*Mexican Optics and Photonics Meeting (MOPM) 2021*”, con el trabajo: “*Repair Function applied to the single solution of genetic algorithm in the nesting of cutting patterns in finite materials*”, celebrado por la *Academia Mexicana de Óptica A.C.*, se muestra la constancia en la figura F.5.

DocuSign Envelope ID: EE93774E-7C9E-4113-AC2E-5A691D7B78ED



LA ACADEMIA MEXICANA DE ÓPTICA

otorga la presente

CONSTANCIA

a

A. Rodríguez Rodríguez, F.J. Cuevas de la Rosa

por su valiosa participación en la
Mexican Optics and Photonics Meeting (MOPM) 2021,
celebrada en modalidad virtual y síncrona
del 18 al 20 de noviembre de 2021,
con la presentación del cartel intitulado

**REPAIR FUNCTION APPLIED TO THE SINGLE SOLUTION
OF GENETIC ALGORITHM IN THE NESTING
OF CUTTING PATTERNS IN FINITE MATERIALS**

Tonanzintla , Pue., 25 de noviembre de 2021.

Atentamente

DocuSigned by:

645BE78512444EA
Dr. Eduardo Tepichin Rodríguez
Presidente de la AMO



Dr. Victor M. Coello Cárdenas
Vice-presidente de la AMO

Figura F.5: *Constancia de participación en el Congreso Mexican Optics and Photonics Meeting (MOPM) 2021.*

Participación en el evento “*Día Internacional de la Niña y la Mujer en la Ciencia 2023*”, mediante la entrevista “*Conociendo a mujeres de ciencia*” celebrado en el Centro de Investigaciones en Óptica A.C. La constancia se muestra en la figura F.6.



Figura F.6: Constancia de participación en el evento *Día Internacional de la Niña y la Mujer en la Ciencia 2023*.

Derivado de la presente investigación, se publicó un artículo científico titulado “*Repair Method and Muéganos Structure Applied to the Nesting Problem in Finite Materials*” en la revista Applied Sciences, año 2023. El certificado de publicación se muestra en la figura F.7.



Figura F.7: Certificado de la Publicación en la revista Applied Sciences.

BIBLIOGRAFÍA

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, New Jersey 07458: Pearson Education, Inc., 2008.
- [2] L. V. Kantorovich, “Mathematical methods of organizing and planning production,” *Management Science*, vol. 6, no. 4, pp. 366–422, Jul. 1960. [Online]. Available: <http://dx.doi.org/10.1287/mnsc.6.4.366>
- [3] R. C. E. Gilmore and R. Gomory, “A linear programming approach to the cutting stock problem,” *Operations Research*, vol. 9, 01 1961.
- [4] R. Martí, *Multi-Start Methods*. Boston, MA: Springer US, 2003, pp. 355–368. [Online]. Available: https://doi.org/10.1007/0-306-48056-5_12
- [5] C. R. Reeves, *Modern Heuristic Techniques for Combinatorial Problems*. New York, NY, USA: John Wiley & Sons, Inc., 1993. [Online]. Available: <http://dl.acm.org/citation.cfm?id=166648.166657>
- [6] M. Fischetti and I. Luzzi, “Mixed-integer programming models for nesting problems,” *Journal of Heuristics*, vol. 15, pp. 201–226, 06 2009.
- [7] S. C. Leung, D. Zhang, C. Zhou, and T. Wu, “A hybrid simulated annealing metaheuristic algorithm for the two-dimensional knapsack packing problem,” *Computers & Operations Research*, vol. 39, no. 1, pp. 64 – 73, 2012, special Issue on Knapsack Problems and Applications. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0305054810002510>
- [8] L. Wei, Q. Hu, A. Lim, and Q. Liu, “A best-fit branch-and-bound heuristic for the unconstrained two-dimensional non-guillotine cutting problem,” *European Journal*

- of *Operational Research*, vol. 270, no. 2, pp. 448–474, 2018. [Online]. Available: <https://ideas.repec.org/a/eee/ejores/v270y2018i2p448-474.html>
- [9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983. [Online]. Available: <https://science.sciencemag.org/content/220/4598/671>
- [10] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization,” *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, Jun 2007. [Online]. Available: <https://doi.org/10.1007/s11721-007-0002-0>
- [11] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Oct 1995, pp. 39–43.
- [12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [13] R. E. Leakey., *Charles Darwin, El origen de las especies. Versión Abreviada*, 1st ed. DF, México.: Martín Casillas Editores, S. A., 1980.
- [14] F. Cuevas, O. Gonzalez, Y. Suzuki, D. Hernández, M. Roche, and N. Alcalá, “Genetic algorithms applied to optics and engineering,” in *Symposium on Optics in Industry*, vol. 6046, 2006. [Online]. Available: <https://doi.org/10.1117/12.674556>
- [15] J. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975. [Online]. Available: <https://books.google.com.mx/books?id=JE5RAAAAMAAJ>
- [16] M. Gen, R. Cheng, and L. Lin, *Network Models and Optimization: Multiobjective Genetic Algorithm Approach*, 1st ed. Springer Publishing Company, Incorporated, Sept 2008. [Online]. Available: <https://api.semanticscholar.org/CorpusID:60406355>
- [17] Y. Suzuki, “Simulación del acomodo de patrones de corte en materiales,” Instituto Tecnológico de León. León, Guanajuato, México, 2003, tesis de Maestría.
- [18] A. K. Sato, L. R. Mundim, T. C. Martins, and M. S. G. Tsuzuki, “A separation and compaction algorithm for the two-open dimension nesting problem using penetration-fit raster and obstruction map,” *Expert Systems with Applications*, vol. 220, p. 119716, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417423002178>

- [19] Y.-P. Cui, Y. Cui, and T. Tang, “Sequential heuristic for the two-dimensional bin-packing problem,” *European Journal of Operational Research*, vol. 240, no. 1, pp. 43–53, 2015. [Online]. Available: <https://ideas.repec.org/a/eee/ejores/v240y2015i1p43-53.html>
- [20] D. Zhang, L. Shi, S. C. Leung, and T. Wu, “A priority heuristic for the guillotine rectangular packing problem,” *Information Processing Letters*, vol. 116, no. 1, pp. 15–21, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020019015001519>
- [21] J. C. Gomez and H. Terashima-Marín, “Evolutionary hyper-heuristics for tackling bi-objective 2d bin packing problems,” *Genetic Programming and Evolvable Machines*, vol. 19, no. 1–2, p. 151–181, Jun 2018. [Online]. Available: <https://doi.org/10.1007/s10710-017-9301-4>
- [22] S. C. Leung, D. Zhang, and K. M. Sim, “A two-stage intelligent search algorithm for the two-dimensional strip packing problem,” *European Journal of Operational Research*, vol. 215, no. 1, pp. 57–69, November 2011. [Online]. Available: <https://ideas.repec.org/a/eee/ejores/v215y2011i1p57-69.html>
- [23] A. Neuenfeldt Júnior, E. Silva, A. M. Gomes, C. Soares, and J. Oliveira, “Data mining based framework to assess solution quality for the rectangular 2d strip-packing problem,” *Expert Systems with Applications*, vol. 118, Oct 2018.
- [24] A. Yin, J. Huang, D. Hu, and C. Chen, “A quick heuristic-dynamic programming for two-dimensional cutting problem,” *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, pp. 164–169, 2019.
- [25] P. R. Pinheiro, B. A. Júnior, and R. D. Saraiva, “A random-key genetic algorithm for solving the nesting problem,” *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 11, pp. 1159–1165, 2016. [Online]. Available: <https://doi.org/10.1080/0951192X.2015.1036522>
- [26] D. N. Hama Rashid, T. A. Rashid, and S. Mirjalili, “Ana: Ant nesting algorithm for optimizing real-world problems,” *Mathematics*, vol. 9, no. 23, 2021. [Online]. Available: <https://www.mdpi.com/2227-7390/9/23/3111>
- [27] B. Ding, R. Zhang, L. Xu, G. Liu, S. Yang, Y. Liu, and Q. Zhang, “U2d2net: Un-supervised unified image dehazing and denoising network for single hazy image enhancement,” *IEEE Transactions on Multimedia*, pp. 1–16, 2023.

- [28] R. Zhang, S. Yang, Q. Zhang, L. Xu, Y. He, and F. Zhang, “Graph-based few-shot learning with transformed feature propagation and optimal class allocation,” *Neurocomputing*, vol. 470, pp. 247–256, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231221016672>
- [29] P. Poshyanonda and C. H. Dagli, “Genetic neuro-nester,” *Journal of Intelligent Manufacturing*, vol. 15, no. 2, pp. 201–218, Apr 2004. [Online]. Available: <https://doi.org/10.1023/B:JIMS.0000018033.05556.65>
- [30] L. R. Mundim, M. Andretta, and T. A. de Queiroz, “A biased random key genetic algorithm for open dimension nesting problems using no-fit raster,” *Expert Systems with Applications*, vol. 81, pp. 358–371, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417417302233>
- [31] I. Kierkosz and M. Luczak, “A one-pass heuristic for nesting problems,” *Operations Research and Decisions*, vol. 1, pp. 37–60, 2019. [Online]. Available: <https://ideas.repec.org/a/wut/journal/v1y2019p37-60id1401.html>
- [32] J. A. Bennell and X. Song, “A comprehensive and robust procedure for obtaining the nofit polygon using minkowski sums,” *Computers & Operations Research*, vol. 35, no. 1, pp. 267–281, 2008, part Special Issue: Applications of OR in Finance. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054806000669>
- [33] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Evolutionary Computation 2 : Advanced Algorithms and Operators*, 1st ed. CRC Press Boca Raton ISBN: 9780367806378, Nov 2000. [Online]. Available: <http://api.semanticscholar.org/CorpusID:63512819>
- [34] L. D. Whitley, V. S. Gordon, and K. E. Mathias, “Lamarckian evolution, the baldwin effect and function optimization,” in *Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature — serie PPSN III*. Berlin, Heidelberg: Springer-Verlag, 1994, p. 6–15.
- [35] F. Glover and M. Laguna, *Tabu Search*. USA: John Wiley & Sons, Inc., 1993, p. 70–150.
- [36] J. Kennedy, *Particle Swarm Optimization*. Boston, MA: Springer US, 2010, pp. 760–766. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_630

- [37] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, May 1998, pp. 69–73.
- [38] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [39] O. Astrachan, “Bubble sort: An archaeological algorithmic analysis,” in *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 1–5. [Online]. Available: <https://doi.org/10.1145/611892.611918>
- [40] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [41] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
- [42] A. Rodríguez, “Algoritmos de visión e inteligencia computacional para el acomodo de patrones de corte en materiales,” Centro de Investigaciones en Óptica A.C. León, Guanajuato, México, pp. 1–89, 2019, tesis de Maestría en Optomecatrónica. [Online]. Available: <https://cio.repositorioinstitucional.mx/jspui/bitstream/1002/1081/3/17617.pdf>
- [43] A. Rodríguez, F. Cuevas, and D. Esparza, “Repair method and muéganos structure applied to the nesting problem in finite materials,” *Applied Sciences*, vol. 13, no. 18, 2023. [Online]. Available: <https://www.mdpi.com/2076-3417/13/18/10117>