

**INSTITUTO TECNOLOGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY**  
**CAMPUS MONTERREY**  
**DIVISION DE GRADUADOS E INVESTIGACION**  
**PROGRAMA DE GRADUADOS EN ELECTRONICA,  
COMPUTACION, INFORMACION Y COMUNICACIONES**



**OPTIMIZACION DEL PROBLEMA DE EMPACADO  
DE MATERIAL EN 3-D MEDIANTE  
ALGORITMOS GENETICOS**

**T E S I S**

**PRESENTADA COMO REQUISITO PARCIAL  
PARA OBTENER EL GRADO ACADEMICO DE  
MAESTRO EN CIENCIAS  
CON ESPECIALIDAD EN SISTEMAS INTELIGENTES**

**VICTOR EMANUEL DE ATOCHA UC CETINA**

**MAYO 2002**

# INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY

DIVISION DE GRADUADOS E INVESTIGACION  
PROGRAMA DE GRADUADOS EN ELECTRONICA,  
COMPUTACION, INFORMACION Y COMUNICACIONES



OPTIMIZACION DEL PROBLEMA DE EMPACADO  
DE MATERIAL EN 3-D MEDIANTE  
ALGORITMOS GENETICOS

T E S I S

PRESENTADA COMO REQUISITO PARCIAL  
PARA OBTENER EL GRADO ACADEMICO DE  
MAESTRO EN CIENCIAS  
CON ESPECIALIDAD EN SISTEMAS INTELIGENTES

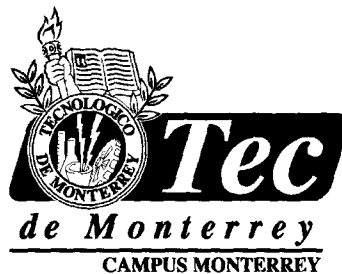
VICTOR EMANUEL DE ATOCHA UC CETINA

MAYO 2002

# Optimización del Problema de Empacado de Material en 3-D mediante Algoritmos Genéticos

por

Víctor Emanuel de Atocha Uc Cetina



Tesis

Presentada al Programa de Graduados en Electrónica, Computación, Información y

Comunicaciones

del

Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey

como requisito parcial para obtener el grado académico de

**Maestro en Ciencias en Sistemas Inteligentes**

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Campus Monterrey**

Monterrey, N.L. Mayo de 2002

A las personas que más quiero:  
Mi padre, Víctor, quien siempre ha sido mi inspiración.  
Mi madre, Wilma, por ser mi fuente de valor.  
Mi hermana y mis hermanos, Flor, Moisés y Rodrigo, quienes son también mis  
mejores amigos.  
Los amo.

## Reconocimientos

Quiero agradecer:

A Dios, por regalarme vida y salud.

Al Dr. Hugo Terashima, por su supervisión y profesionalismo.

A mis sinodales, los doctores Manuel Valenzuela y José Luis González, por sus valiosos consejos y puntos de vista.

Al Dr. Horacio Martínez, por facilitarme el uso de su equipo de cómputo para mis experimentos finales.

Al Centro de Inteligencia Artificial, por prestarme sus instalaciones.

A mi familia por su amor y apoyo incondicional.

Y finalmente, a mis amigos en Monterrey y en Mérida, por ser eso, mis amigos.

VÍCTOR EMANUEL DE ATOCHA UC CETINA

*Instituto Tecnológico y de Estudios Superiores de Monterrey*  
*Mayo 2002*

# Optimización del Problema de Empacado de Material en 3-D mediante Algoritmos Genéticos

M.C. Víctor Emanuel de Atocha Uc Cetina  
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2002

Asesor de la tesis: Dr. Hugo Terashima Marín

Este documento presenta la tesis para obtener el grado de Maestro en Ciencias en Sistemas Inteligentes. La tesis trata sobre la optimización de la carga de contenedores, lo cual se considera como un problema de empaçado en tres dimensiones. El siguiente trabajo propone el uso de una técnica de la Inteligencia Artificial para encontrar soluciones eficientes al problema mencionado, mediante el uso de Algoritmos Genéticos (AGs), los cuales son algoritmos de búsqueda basados en la mecánica de la selección natural y en la genética natural, un método que ha sido aplicado exitosamente en diversos problemas de optimización. El problema de la carga de contenedores presenta dos grupos de datos básicos, por una parte, existe una lista de objetos grandes consistente de uno o más contenedores, y por otra parte, una lista de objetos más pequeños que deben ser empaçados dentro de los contenedores. Aparte de ciertos objetivos específicos y restricciones dadas, el principal aspecto de la carga de contenedores concierne a la combinación geométrica de objetos pequeños formando patrones de empaçado que pueden ser asignados a los contenedores de la lista, dependiendo de sus formas y dimensiones. Como resultado de esta investigación, se diseñó y se implementó un AG para optimización en este dominio. El AG fue denominado Algoritmo Genético para Empacado (AGE) y fue probado con problemas de un contenedor, usando tres funciones objetivo distintas y dos operadores de cruce diferentes. Además, se realizaron experimentos con problemas de empaçado de más de un contenedor. Finalmente, se utilizó el AGE para resolver dos problemas reales de carga de contenedores de una compañía de transportación, obteniendo resultados satisfactorios.

# Índice General

<b>Reconocimientos</b>	<b>v</b>
<b>Resumen</b>	<b>vi</b>
<b>Índice de Tablas</b>	<b>x</b>
<b>Índice de Figuras</b>	<b>xi</b>
<b>Capítulo 1 Introducción</b>	<b>1</b>
1.1 Contexto de la Investigación . . . . .	2
1.2 Definición del Problema . . . . .	3
1.3 Objetivo . . . . .	5
1.4 Hipótesis de Investigación . . . . .	5
1.5 Alcances . . . . .	6
1.6 Contribuciones de la Investigación . . . . .	6
1.7 Estructura de la Tesis . . . . .	7
<b>Capítulo 2 Antecedentes</b>	<b>8</b>
2.1 Problema de Empacado en Contenedores en una Dimensión . . . . .	8
2.2 Problema de Empacado del Rectángulo . . . . .	9
2.3 Un Método Eficiente para el Problema de Carga de Palets Múltiples . .	10
2.4 Uso del Algoritmo Genético Co-operativo y Co-evolutivo para resolver el Problema de Carga de Contenedores en Tres Dimensiones . . . . .	12
2.5 Resumen . . . . .	12
<b>Capítulo 3 Aspectos Generales del Problema de Corte y Empacado</b>	<b>13</b>
3.1 Estructura Básica de los Problemas de C&P . . . . .	13
3.2 Características Principales de los Problemas de C&P . . . . .	14
3.2.1 Dimensionalidad . . . . .	15
3.2.2 Medición de la Cantidad . . . . .	15
3.2.3 Forma de las Figuras . . . . .	16
3.2.4 Disponibilidad . . . . .	17

3.2.5	Restricciones de Patrones . . . . .	18
3.2.6	Restricciones de Asignación . . . . .	18
3.2.7	Objetivos de los Problemas de C&P . . . . .	20
3.3	Resumen . . . . .	20
<b>Capítulo 4 Algoritmos Genéticos para Optimización</b>		<b>21</b>
4.1	Fundamentos de los Algoritmos Genéticos . . . . .	21
4.2	Cómo Funcionan los Algoritmos Genéticos . . . . .	22
4.3	Fundamentos Matemáticos . . . . .	25
4.4	Modelo de Solución del Problema de Empacado . . . . .	26
4.5	Diseño e Implementación del Algoritmo Genético para Empacado . . . . .	26
4.5.1	Función Objetivo . . . . .	26
4.5.2	Representación del Cromosoma . . . . .	28
4.5.3	Operador de Selección . . . . .	29
4.5.4	Operadores de Cruce . . . . .	30
4.5.5	Operador de Mutación . . . . .	31
4.5.6	Criterio de Terminación . . . . .	31
4.6	Resumen . . . . .	31
<b>Capítulo 5 Resultados Experimentales</b>		<b>33</b>
5.1	Ajuste del Algoritmo Genético para Empacado . . . . .	33
5.2	Problemas de Empacado con Un Contenedor . . . . .	41
5.2.1	Función Objetivo de Volumen . . . . .	41
5.2.2	Función Objetivo de Adyacencia . . . . .	43
5.2.3	Función Objetivo de Espacios Vacíos . . . . .	47
5.2.4	Comparación de las Funciones Objetivo . . . . .	51
5.3	Resolviendo Diversos Problemas de Empacado . . . . .	52
5.3.1	Incrementando el Número de Objetos . . . . .	53
5.3.2	Incrementando el Tamaño de los Objetos . . . . .	54
5.4	Problemas de Empacado con $n$ Contenedores . . . . .	55
5.4.1	Trabajando sin Heurística . . . . .	56
5.4.2	Trabajando con la Heurística de Prioridad . . . . .	57
5.4.3	Puntaje de Empacado para Cargar 2 Contenedores . . . . .	60
5.5	Resumen . . . . .	61
<b>Capítulo 6 Resolviendo Problemas Reales de Empacado</b>		<b>63</b>
6.1	Experimentando con Problemas Reales . . . . .	63
6.2	Primer Problema Real de Empacado . . . . .	67
6.3	Segundo Problema Real de Empacado . . . . .	70
6.4	Resumen . . . . .	74



<b>Capítulo 7 Conclusiones y Trabajo Futuro</b>	<b>75</b>
<b>Bibliografía</b>	<b>78</b>
<b>Vita</b>	<b>80</b>

## Índice de Tablas

5.1	Dimensiones de los objetos utilizados con fines experimentales. . . . .	35
5.2	Resultados experimentales empleando el operador <i>PMX</i> . . . . .	36
5.3	Resultados experimentales empleando el operador <i>ERX</i> . . . . .	37
5.4	Puntajes de empaçado obtenidos con la función objetivo de volumen. . .	44
5.5	Puntajes de empaçado obtenidos con la función objetivo de adyacencia.	47
5.6	Puntajes de empaçado obtenidos utilizando la función objetivo de espa- cios vacíos. . . . .	50
5.7	Puntajes obtenidos con diferentes funciones objetivo. . . . .	51
5.8	Características de las diferentes funciones objetivo. . . . .	52
5.9	Puntajes de empaçado e incrementos en el tiempo de ejecución para problemas que involucran diferentes números de objetos. . . . .	53
5.10	Puntajes de empaçado e incrementos en el tiempo de ejecución para problemas que involucran diferentes tamaños de objetos. . . . .	54
5.11	Catálogo de objetos utilizados para experimentos con problemas de $n$ contenedores. . . . .	56
5.12	Puntajes obtenidos por el AGE con un problema de 2 contenedores y sin heurística. . . . .	61
5.13	Puntajes obtenidos por el AGE con un problema de 2 contenedores y con la heurística de prioridad. . . . .	61
6.1	Dimensiones de los objetos utilizados para resolver el primer problema real de empaçado. . . . .	67
6.2	Puntajes de empaçado del primer problema. . . . .	68
6.3	Puntajes de empaçado del primer problema con 2 secuencias de carga. .	69
6.4	Dimensiones de los objetos del primer subproblema. . . . .	71
6.5	Dimensiones de los objetos del segundo subproblema. . . . .	72
6.6	Puntajes del primer subproblema. . . . .	72
6.7	Puntajes del segundo subproblema. . . . .	72
6.8	Puntajes del primer subproblema con 4 secuencias de carga. . . . .	73
6.9	Puntajes del segundo subproblema con 2 secuencias de carga. . . . .	73

## Índice de Figuras

4.1	Diagrama de flujo del algoritmo genético. . . . .	23
4.2	Modelo de solución del problema de empaçado en tres dimensiones. . .	27
4.3	Estructura del cromosoma. . . . .	29
4.4	Posiciones posibles de un objeto cuando $p = 6$ . . . . .	29
4.5	Un ejemplo de mutación. . . . .	31
5.1	Catálogo de objetos utilizados con fines experimentales. . . . .	34
5.2	Posibles posiciones de los objetos tipo $D$ . . . . .	35
5.3	Curva de aptitud usando el operador $PMX$ (Evaluación máxima = 5070.9). . . . .	38
5.4	Curva de aptitud usando el operador $ERX$ (Evaluación máxima = 6465). . . . .	38
5.5	Curvas de aptitud utilizando los operadores $PMX$ y $ERX$ durante 70 generaciones. . . . .	39
5.6	Una solución buena utilizando el operador $PMX$ (Evaluación máxima = 5070.9). . . . .	40
5.7	Una solución buena utilizando el operador $ERX$ (Evaluación máxima = 6465). . . . .	41
5.8	Curva de aptitud usando la función objetivo de volumen. . . . .	43
5.9	Una solución buena utilizando la función objetivo de volumen. . . . .	44
5.10	Curva de aptitud usando la función objetivo de adyacencia. . . . .	46
5.11	Una solución buena utilizando la función objetivo de adyacencia. . . . .	48
5.12	Curva de aptitud usando la función objetivo de espacios vacíos. . . . .	49
5.13	Una solución buena utilizando la función objetivo de espacios vacíos. . .	50
5.14	Tipos de objetos utilizados en experimentos con el tiempo requerido por el AGE para proporcionar una solución. . . . .	53
5.15	Los tipos de objetos $A$ , $C$ y $D$ fueron utilizados en experimentos con $n$ contenedores. . . . .	56
5.16	Curva de aptitud al resolver un problema con 2 contenedores y sin heurística. . . . .	57
5.17	Una solución buena sin heurística: Contenedor 1. . . . .	58
5.18	Una solución buena sin heurística: Contenedor 2. . . . .	58

5.19	Curva de aptitud al resolver un problema con 2 contenedores usando la heurística de prioridad. . . . .	59
5.20	Una solución buena con la heurística de prioridad: Contenedor 1. . . .	60
5.21	Una solución buena con la heurística de prioridad: Contenedor 2. . . .	60
6.1	Clasificación de las cajas dependiendo de sus dimensiones. . . . .	65
6.2	Posiciones posibles de un objeto cuando $p = 6$ . . . . .	66
6.3	a) Forma del objeto tipo 15. b) Paquete de objetos tipo 15. . . . .	66
6.4	Comportamiento de la curva de aptitud del AGE al resolver el primer problema real de empaçado. . . . .	68
6.5	Solución del primer problema real. . . . .	69
6.6	División del contenedor para empaçar las diferentes ordenes de mercancía. . . . .	71

## Capítulo 1

### Introducción

Considere a un distribuidor de muebles para oficina que necesita satisfacer la demanda de productos por parte de varios clientes en otros puntos de venta. Requiere hacer el transporte del total de la mercancía solicitada para una ciudad, utilizando el menor número de camiones y reduciendo de esta forma los costos de envío. Todos los productos que deben ser transportados están en cajas, las cuales varían de tamaño y dimensiones dependiendo de su contenido. Entre la mercancía pueden contarse sillas, escritorios, mesas, libreros, pizarrones, etc. En otras palabras, existe una gran diversidad en los tamaños y cantidades de las cajas que se requiere transportar. Este problema ocurre generalmente en situaciones industriales y se le conoce como “problema de empaqueo en contenedores” y “problema de carga”. Para determinar el número de camiones necesarios, el comerciante podría primero examinar el número total de patrones en los que puede cargar los camiones con las cajas de mercancía requeridas. Después, estos patrones deben ser combinados de forma eficiente para que se utilice el menor número de camiones. La dificultad radica en el número muy grande de patrones que se pueden generar.

Este tipo de problemas de empaqueo presenta una dualidad que permite verlos también como problemas de corte. Tomemos por ejemplo el problema del empaqueo de rectángulos en dos dimensiones. Al acomodar un determinado número de figuras geométricas dentro del área limitada por un rectángulo, es claro que el problema a resolver es un problema de empaqueo. Sin embargo, si esas figuras que acomodamos o empacamos dentro del rectángulo, son piezas que deben ser cortadas del rectángulo para elaborar quizás un producto más complejo, entonces, nuestro problema de empaqueo se convierte en un problema de corte de material. Esta dualidad existente permite definir y utilizar una misma clasificación para ambos tipos de problemas. Por otra parte, la tarea de corte y empaqueo de material es considerada evidentemente como un problema de optimización combinatoria y cae dentro de los problemas denominados NP completos, los cuales serán explicados a continuación.

## 1.1 Contexto de la Investigación

Resultados reportados en la teoría de la complejidad indican que muchos problemas combinatorios son de hecho muy difíciles de resolver [6]. Resolver un problema en tiempo polinomial significa que hay un algoritmo que puede sugerir una solución y luego verificar si esta solución es correcta en un tiempo polinomial [12]. Este es el tipo de problema P. Otro tipo de problemas, el NP, es el que contiene problemas, que aunque son resolvibles, aún no se ha comprobado que existe un algoritmo que pueda sugerir soluciones en un tiempo polinomial para todas las instancias del problema. Los problemas NP completos son los problemas más difíciles entre los problemas de tipo NP [2]. El problema de corte y empaado de material pertenece a este tipo de problemas [6] y es por eso que resulta interesante encontrar diferentes alternativas de solución.

El problema de corte de material está entre los primeros problemas mencionados en la literatura de investigación de operaciones con aplicaciones obvias en todas las industrias cuyos productos están en forma de hojas planas [7]. Genéricamente, el problema de corte de material involucra órdenes de clientes para cortar material almacenado en bodegas, tratando de minimizar el desperdicio y otros costos de producción relacionados [1]. Debido a la dualidad que presentan los problemas de corte y de empaado, algunos autores hacen referencia a ambos como un mismo tipo de problemas, y para este trabajo de investigación es importante la forma en la cual son definidos.

Los problemas de corte y empaado en tres dimensiones (3D C&P, por sus siglas en inglés), pueden definirse como sigue:

- Existen dos grupos básicos de datos cuyos elementos definen formas geométricas en un espacio de tres dimensiones: un conjunto de contenedores y un conjunto de objetos.
- El proceso de corte y empaado encuentra patrones los cuales son combinaciones geométricas de objetos asignados a contenedores. Todos o algunos de los objetos son cortados de o empaados en los contenedores para optimizar una función objetivo.

Los problemas de corte y empaado en 3 dimensiones incluyen la carga de contenedores, la carga de palets (Paletización en 3D), y problemas de anidamiento en 3D. El problema de corte y empaado libre en 3 dimensiones es una variación del problema original de corte y empaado en 3 dimensiones, cuya diferencia radica en que tanto los contenedores como los objetos no están restringidos a formas de paralelepípedo y pueden tener formas complejas [14]. Los primeros trabajos realizados en corte y empaado en 3D fueron llevados a cabo alrededor de 1980, cuando George y Robinson [4] trabajaron por

primera vez en el problema de determinar cómo puede ser acomodado un conjunto de cajas dentro de un contenedor. Años antes, en 1978, se realizaron algunos trabajos para determinar el número mínimo de contenedores necesarios para empacar un conjunto de cajas. La presente tesis se enfoca en resolver el problema de cómo empacar el mayor número de objetos dentro de un contenedor. Los objetos y los contenedores tienen formas de paralelepípedo y sus dimensiones son conocidas. Este problema se presenta con mucha frecuencia en la industria cuando determinada mercancía es empacada en contenedores para ser transportada de un lugar a otro.

El problema de empaqueo es considerado como un problema combinatorio y un método para resolver tal tipo de problema ya comprobado es el Algoritmo Genético (AG). Los AGs son algoritmos de búsqueda basados en la mecánica de la selección natural y la genética natural. Combinan la supervivencia del más apto entre estructuras de cadenas con un intercambio de información estructurada para formar un algoritmo de búsqueda. En cada generación, se crea un nuevo conjunto de criaturas artificiales (cadenas), usando bits y partes de los individuos viejos más aptos; ocasionalmente se prueban partes nuevas para tratar de mejorar la medición. Aunque funcionan de manera aleatoria, los AGs no siguen un camino aleatorio. Ellos explotan eficientemente información histórica para especular sobre nuevos puntos de búsqueda que proporcionen un mejor desempeño. Los algoritmos genéticos han demostrado teóricamente y empíricamente que proporcionan una búsqueda robusta en espacios complejos. Estos algoritmos son simples computacionalmente hablando y muy poderosos en la búsqueda de mejoras [5].

Un AG comienza con una población inicial de individuos representados por cadenas de caracteres tomados generalmente del alfabeto binario, sin embargo, hay ocasiones en las que es mejor utilizar cromosomas hechos de números enteros o reales; la población es inicializada normalmente de manera aleatoria. El AG combina información entre los individuos de la población usando operadores análogos a la selección del más apto y a la reproducción de las especies. Además se incluye un operador de mutación que simula el proceso de mutación en los individuos naturales. En este trabajo de tesis se utiliza un AG para resolver el problema de empaqueo.

## 1.2 Definición del Problema

Hay muchos problemas interesantes relacionados a la optimización del corte y empaqueo de material, problemas para resolver en uno, dos, tres o aún más dimensiones. Sin embargo, el presente trabajo de investigación se enfoca en el problema de empaqueo en tres dimensiones y está orientado al diseño e implementación computacional de una

herramienta para optimizar la forma en la que se deben empacar  $m$  objetos dentro de  $n$  contenedores considerando una serie de restricciones.

En general, hablar de un problema de optimización es hablar de minimizar o maximizar una función objetivo. En este caso, la función objetivo será maximizada y consistirá básicamente de una función que represente el volumen total de los objetos empacados dentro de los contenedores. En otras palabras, se requiere empacar el mayor volumen de objetos. Otra característica que será considerada en la función objetivo para decidir si una solución es mejor o peor que otra es qué tan cerca son empacados unos de otros, los objetos pertenecientes a un mismo tipo. Las diversas funciones objetivos generadas en el proceso de encontrar la mejor serán explicadas posteriormente en este documento.

Dado que el problema de empacado es un problema de optimización combinatoria muy difícil, es necesario utilizar una herramienta que garantice la obtención de buenas soluciones cuando se aplica a este tipo específico de problemas complejos. Una herramienta que ha demostrado proporcionar excelentes resultados, cuando se aplica para resolver problemas combinatorios, son los Algoritmos Genéticos. En especial, los Algoritmos Genéticos con cromosomas hechos de cadenas de enteros dan buenos resultados cuando se usan para resolver este tipo de problemas [3].

Por lo tanto, el problema a resolver puede ser definido como sigue: Es un problema de

- empacado,
- en tres dimensiones,
- con  $m$  objetos pequeños (artículos a empacar), y
- $n$  objetos más grandes (contenedores).

Donde  $m$  y  $n$  son números naturales.

Las restricciones básicas consideradas son:

- Dos o más objetos no pueden ocupar el mismo lugar al mismo tiempo.
- Un objeto no puede quedar suspendido en el aire, sino que debe ser colocado sobre una superficie que puede ser el fondo del contenedor o la cima de otro objeto.
- Los objetos y los contenedores tienen formas de paralelepípedos.

Y las suposiciones básicas son:



- Los objetos no tienen peso.
- Los contenedores tienen forma de cajas grandes.
- Los objetos son colocados dentro de cada contenedor desde arriba.

### 1.3 Objetivo

Desarrollar un software que sirva como asistente en la tarea de cargar contenedores con diversos objetos de formas rectangulares, mediante un Algoritmo Genético que servirá para encontrar las posiciones en las que deben ser cargados los objetos dentro de los contenedores. Los objetivos particulares que se persiguen en este estudio son los siguientes:

- Diseñar una estructura eficiente de cromosoma que permita representar correctamente los patrones de empaçado.
- Definir una función objetivo que represente correctamente el problema de empaçado y que permita la correcta evaluación de los individuos de la población del Algoritmo Genético.
- Codificar un Algoritmo Genético que optimice el empaçado de  $m$  objetos dentro de  $n$  contenedores.
- Desarrollar una interfaz gráfica en C y OpenGL para mostrar las soluciones proporcionadas por el Algoritmo Genético.
- Probar el sistema con diferentes configuraciones de problema, usando diferentes funciones objetivo y diferentes conjuntos de parámetros de entrada para el AG.
- Probar el Algoritmo Genético desarrollado para resolver dos problemas reales de carga de contenedores.

### 1.4 Hipótesis de Investigación

En el problema de empaçado, un Algoritmo Genético permite encontrar combinaciones de patrones para empaçar objetos dentro de contenedores, de forma tal que los espacios vacíos en el contenedor pueden ser minimizados al mismo tiempo que el número de objetos empaçados es maximizado.

Las siguientes preguntas serán respondidas:

¿Es posible diseñar un Algoritmo Genético para resolver el problema de empaçado?

¿Es posible encontrar una función objetivo que permita evaluar correctamente los individuos de la población?

¿Cuál es el conjunto de parámetros de entrada del Algoritmo Genético que permiten encontrar soluciones satisfactorias?

¿Cuáles son las características de un problema de empaçado que son difíciles de modelar?

¿Puede este Algoritmo Genético proporcionar soluciones satisfactorias a problemas reales de empaçado?

## 1.5 Alcances

El presente trabajo de investigación considera el desarrollo de un software que indique a través de una interfaz gráfica, la forma de empaçar  $m$  objetos dentro de  $n$  contenedores de manera eficiente. Se asume que tanto los objetos como los contenedores tienen forma de paralelepípedos. La técnica de Inteligencia Artificial empleada para resolver este problema es un Algoritmo Genético, el cual permite especificar tanto el número como las dimensiones de los objetos y contenedores. El software fue desarrollado en C y OpenGL.

## 1.6 Contribuciones de la Investigación

El Algoritmo Genético para Empaçado (AGE), como se ha denominado, es el resultado de este trabajo de investigación y contribuye al área de la computación evolutiva demostrando una vez más la robustez de los Algoritmos Genéticos para resolver problemas difíciles de optimización combinatoria.

Esta tesis presenta una estructura de cromosoma para representar las posibles soluciones del problema de empaçado, usando cadenas de números enteros, donde cada gene representa un tipo de objeto, su posición dentro del contenedor, y el número específico del contenedor en el cual debe ser empaçado, cuando se trata de un problema de más de un contenedor.

Además, se presenta una función objetivo que toma en cuenta el número total de unidades cúbicas empaçadas, el número de objetos del mismo tipo colocados en

posiciones adyacentes dentro del contenedor y además minimiza el número de unidades cúbicas desperdiciadas al momento de empaclar los objetos.

Se propone un método para resolver problemas de empaclado cuando intervienen más de un contenedor. Se le llama “heurística de prioridad” y consiste en indicar al AG que objetos deben ser empacados primeramente.

Algunos resultados experimentales muestran que *ERX*, un algoritmo especial de cruce, proporciona mejores soluciones que el operador *PMX*, cuando ambos son aplicados a problemas de permutaciones.

El AGE proporciona una manera automatizada de hacer optimización de carga de contenedores en problemas reales y proporciona mejores soluciones que la realización de esta actividad de manera empírica.

## 1.7 Estructura de la Tesis

A continuación se presenta una descripción del contenido de los capítulos 2, 3, 4, 5, 6 y 7. El capítulo 2 contiene los antecedentes necesarios relacionados con el área de investigación. El capítulo 3 proporciona una descripción detallada de los problemas de corte y empaclado en tres dimensiones. El capítulo 4 presenta los conceptos básicos de los Algoritmos Genéticos, una explicación de cómo funcionan y de cómo fue implementado el Algoritmo Genético propuesto. El capítulo 5 presenta los resultados experimentales obtenidos con el AGE, las diferentes funciones objetivo implementadas y algunos métodos para resolver el problema de empaclado con más de un contenedor. En el capítulo 6 se utiliza el AGE para resolver dos problemas reales de carga de contenedores. Finalmente, en el capítulo 7 se presentan algunas conclusiones del trabajo de investigación y recomendaciones para trabajos futuros.

## Capítulo 2

### Antecedentes

Este capítulo proporciona los antecedentes necesarios para entender el tópico principal de esta investigación que se enfoca al problema de empaçado en tres dimensiones. Se presentan algunos ejemplos tomados de la literatura relacionada al problema de empaçado de material en una, dos y tres dimensiones. También se explica la manera en la cual fue solucionado cada uno de los problemas presentados. El primer problema presentado se refiere al comúnmente conocido como problema de empaçado de *bins*, que consiste en minimizar el número total de contenedores necesarios para empaçar  $n$  pesos. Después se menciona el problema del empaçado de rectángulos, un problema en dos dimensiones y que consiste como su nombre lo dice en empaçar varios rectángulos dentro de un rectángulo más grande. Posteriormente se presenta una heurística que se propone para solucionar el problema de carga de palets múltiples, el cual es un problema en tres dimensiones. Finalmente se presenta un trabajo en el cual se utilizó un Algoritmo Genético Co-operativo y Co-evolutivo para resolver el problema de carga de contenedores en tres dimensiones.

### 2.1 Problema de Empaçado en Contenedores en una Dimensión

El problema de empaçado en contenedores en una dimensión es definido por Cheng-Yan Kao y Feng-Tse Lin [10] como sigue. Se tiene una lista de  $n$  pesos y un conjunto de contenedores, cada uno con la misma capacidad. El objetivo es encontrar una asignación posible de todos los pesos a los contenedores para minimizar el número total de contenedores utilizados. El problema de empaçado de contenedores modela varios problemas prácticos en diversas áreas como control industrial, sistemas computacionales, asignación de horarios de máquina, diseño de chips VLSI, etc. Algunos ejemplos típicos son los camiones con límite de peso dado, el corte de material en una dimensión, optimización de la longitud de cable, la asignación de pistas en discos, paginado, el empaçado de cadenas de longitud variable dentro de palabras de longitud establecida y la minimización del número de procesadores necesarios para completar todas las tareas en

la fecha límite establecida. El problema de empaqueo de contenedores es NP completo, así que es muy improbable que exista un algoritmo de tiempo polinomial para encontrar soluciones óptimas.

La solución propuesta es un método estocástico basado en la naturaleza, como un algoritmo nuevo de aproximación para resolver el problema de empaqueo de contenedores. Esta solución propuesta combina el recocido simulado con el algoritmo genético y realiza algunas modificaciones a estos algoritmos para obtener uno nuevo que trabaje más rápido.

Las reglas básicas son las siguientes. Empezando con el punto de menor costo en la generación anterior, un punto siguiente se genera a partir del punto actual por la estrategia de movimiento de la generación. Si es aceptado por el criterio, el punto siguiente no sólo se convierte en el punto actual sino que también se convierte en miembro de la siguiente generación. Este proceso se continúa hasta que se genera la siguiente población. El punto de menor costo en cada generación se copia a un vector de solución independiente conforme continúa la búsqueda y se imprime al final de la corrida. Finalmente, cuando el 80% de la población en una determinada generación tienen el mismo costo del vector de solución, se cumple la condición de enfriamiento. Después de realizar un procedimiento de búsqueda local, el algoritmo se detiene.

## 2.2 Problema de Empaqueo del Rectángulo

El problema de determinar si un conjunto de rectángulos pueden ser empaquados dentro de otro rectángulo mayor puede relacionarse con algunas aplicaciones prácticas en planeación de piso VLSI, ubicación, o en problemas de asignación de tiempos. Se ha demostrado que este problema es de tipo NP completo y fue estudiado por Tam, Wu, Huang y Wong [15] como se muestra a continuación. Varios esquemas de aproximación o heurísticas se han propuesto para atacar estos problemas. Buscando heurísticas buenas que sean rápidas (aunque no completas), la gente se ha tratado de inspirar con la naturaleza. Basándose en observaciones de fenómenos de procesos físicos, se han explorado la clase de heurísticas denominadas casi-físicas (simulando la naturaleza).

Para algunos problemas, existen algunas heurísticas sencillas y efectivas cuyas estrategias fueron desarrolladas mediante la experiencia acumulada de los humanos al resolver problemas semejantes día tras día. Las heurísticas de este tipo se definen como las heurísticas basadas en comportamiento casi-humano. De hecho, el problema del empaqueo del rectángulo es simplemente un caso simplificado del problema de empaqueo del plato de piedra de forma poligonal que los albañiles de la antigüedad tenían que

resolver con frecuencia en su trabajo. Este método simplemente adopta esas estrategias de miles de años de antigüedad desarrolladas por esos trabajadores profesionales de la antigüedad (granjeros, albañiles, vendedores, etc.) y las mejora para que se ajusten de mejor forma a este problema de empaçado. Probablemente debido a su gran efectividad, estas técnicas han sobrevivido durante generaciones.

Por miles de años, los albañiles chinos han pasado por la denominada regla del dedo pulgar, en su trabajo de empaçado, la cual sugiere que las esquinas vacías dentro del área limitada deben ser llenadas primeramente, después las líneas que limitan los espacios vacíos y finalmente el resto de los espacios vacíos. Se cree que esta forma de empaçado ha sido la más razonable basándose en la gran experiencia de esos profesionales. Se propone entonces, la denominada heurística de la ocupación de la esquina, inspirándose en esta vieja estrategia. Parece natural considerar que si existe una solución de “empaçado completo” (por ejemplo, cuando no quedan espacios vacíos en el empaçado final) de un determinado problema, entonces con seguridad debe existir también alguna secuencia de empaçado tal que cada rectángulo se empaque siempre en una posición ocupando una esquina vacía formada por otro rectángulo previamente empaçado o por los límites de la caja. Esta noción está de acuerdo de alguna manera con las lecciones dadas por las reglas mencionadas anteriormente, las cuales evitan empaçar en posiciones del centro que no están pegadas a ningún objeto. Las reglas consideran ese tipo de empaçado como no razonable, ya que los rectángulos empaçados de esta manera solamente crean obstáculos para empaçados posteriores y podrían necesitar ser movidos una y otra vez hasta poder encontrar una solución definitiva.

Dada una caja vacía de forma rectangular y un conjunto de rectángulos de tamaños arbitrarios, se debe determinar si todos estos rectángulos pueden ser empaçados dentro de la caja vacía si ningún traslape.

## **2.3 Un Método Eficiente para el Problema de Carga de Palets Múltiples**

Este problema consiste en cargar un conjunto de productos diferentes con cantidades dadas, en palets (o en contenedores) y minimizar el número de palets requeridos. El objetivo teórico de encontrar el mejor espacio está restringido por una lista de aspectos prácticos (restricciones tecnológicas, distribución del peso sobre el palet, aspectos de estabilidad, etc.). J. Terno, G. Scheithauer y Sommerweib [16] proponen una heurística eficiente para este problema. El método de solución en tres dimensiones usa en el núcleo de su estrategia de carga, patrones en dos dimensiones que son óptimos.

En el problema de carga de palets múltiples se tiene una orden de varios tipos de cajas y el objetivo es encontrar un número mínimo de palets (y sus patrones de carga correspondientes) necesarios para cargar toda la orden. Adicionalmente, a diferencia de la carga de contenedores, en el problema de palets múltiples los aspectos de estabilidad de los patrones de carga deben tomarse muy en cuenta debido a que en este tipo de problema se carece de las paredes del contenedor.

Dado que es muy improbable encontrar un algoritmo de tiempo polinomial para resolver el problema de carga de múltiples palets de manera óptima, dos aspectos deben tomarse muy en cuenta para solucionar este problema. Por una parte, para poder calcular una solución óptima, se debe usar una estrategia como a lo ancho y a los límites. Por otra parte, en situaciones prácticas se deben cumplir con un tiempo límite para obtener la solución de empacado. Por lo tanto se requiere de un método de heurística que nos proporcione soluciones posibles rápidamente, que maneje una instancia del problema de tamaño mediano, y que proporcione soluciones óptimas probadas para un rango amplio de diferentes instancias.

El método de solución surge a partir de las siguientes observaciones que pueden ser hechas cuando una instancia del problema de carga de palets es resuelto secuencialmente, por ejemplo, para la instancia  $E$  se calcula el mejor patrón de carga, después se reduce  $E$ , y se calcula el próximo patrón de carga, etc., hasta que todos los objetos son empacados. Cuando se utiliza este método de solución, aparecen ciertas dificultades computacionales (con respecto al tiempo que se consume), dado que todos los objetos son utilizados en la carga, y el aspecto más complicado es que se debe calcular un criterio de aceptación de patrones de carga con respecto al volumen que debe ser aceptado o mejorado. Por lo tanto se utiliza otro método de solución.

El concepto del algoritmo del método de solución propuesto por Scheithauer, Terno y Sommerweib [16], se basa en un concepto complejo de ancho y límite. Tomando en cuenta un tiempo límite dado en el cual debe ser encontrada una solución posible, se calcula la primera solución posible que puede obtenerse. Después se realizan algunos intentos para mejorar la solución actual. Estos intentos se detienen al momento de encontrar una solución óptima o bien, si se excede el límite de tiempo. El método de solución consiste en una estructura jerárquica. Algunos algoritmos para resolver subproblemas son utilizados en el procedimiento principal de búsqueda de soluciones.

El método propuesto por Scheithauer, Terno y Sommerweib [16] puede usarse como una base para mayores investigaciones en conexión con el problema de la carga de palets. Surge una situación más general cuando se tiene una orden de un tipo de objeto con límites inferiores y superiores.

## 2.4 Uso del Algoritmo Genético Co-operativo y Co-evolutivo para resolver el Problema de Carga de Contenedores en Tres Dimensiones

C. Pimpawat y N. Chaiyaratana [11] proponen un procedimiento para resolver la carga de contenedores en 3D. El procedimiento involucra la utilización de una regla de heurística y la búsqueda de un Algoritmo Genético. La regla de heurística usada emplea un esquema una clasificación de los paquetes en tres grupos distintos: grupos de paquetes grandes, de paquetes medianos y de paquetes chicos. La introducción de esta regla de heurística resulta en la partición de la secuencia de carga completa, en tres sub-secuencias, donde cada sub-secuencia consiste en cargar ordenes de paquetes pertenecientes al mismo grupo. Con el uso de este esquema de partición de secuencias, se realiza una búsqueda de la mejor combinación de sub-secuencias mediante la aplicación del Algoritmo Genético Co-operativo Co-evolutivo (CCGA) propuesto primeramente por Potter y De Jong [13]. En este caso la población de búsqueda está compuesta de tres sub-poblaciones, donde un individuo de cada sub-población representa una secuencia de carga de paquetes pertenecientes al mismo grupo. Los resultados obtenidos indican que el desempeño del CCGA en cuanto a la búsqueda es mejor que el realizado por el Algoritmo Genético estándar, en términos del menor número de contenedores requeridos para almacenar todos los paquetes, lo cual sirve para confirmar la idoneidad y efectividad de la regla de heurística propuesta cuando se utiliza en conjunto con la técnica de búsqueda del Algoritmo Genético Co-operativo y Co-evolutivo.

## 2.5 Resumen

Los problemas de empaqueo son problemas NP completos que han sido resueltos para muchas de sus instancias usando diversas técnicas. En este capítulo se explicaron cuatro casos de problemas de empaqueo y sus métodos de solución. Estos problemas son de una, dos y tres dimensiones. Todos los problemas presentados en este capítulo son del tipo NP completo.



## Capítulo 3

# Aspectos Generales del Problema de Corte y Empacado

En este capítulo se explican los aspectos generales del problema de corte y empacado (C&P) junto con la topología que integra los diferentes tipos de problemas. El tópico de corte y empacado se caracteriza por el hecho de que problemas con la misma estructura lógica esencial aparecen con diferentes nombres en la literatura, por ejemplo:

- Problemas de corte de material y pérdida de sobrantes,
- Problemas de empacado de *bins*, empacado de *bins* duales, empacado de franjas, empacado de vectores, y de la mochila,
- Problemas de carga de vehículos, carga de palets, carga de contenedores y carga de carros,
- Problemas de surtido, agotamiento, diseño, división, disposición, anidamiento y particionamiento,
- Problemas de presupuesto capital, realización de cambios, balanceo de línea, asignación de memoria y programación de procesadores.

### 3.1 Estructura Básica de los Problemas de C&P

Para entender la estructura básica de los problemas de corte y empacado, algunos ejemplos son de utilidad.

Primeramente, existe un problema de corte de tubos para radiadores. Por una parte hay un número ilimitado de tubos de longitudes de 98, usados para producir tubos más pequeños. Por otra parte, hay una lista de tubos de longitudes que van de 5 a 46, los cuales deben ser producidos para satisfacer la demanda semanal. Estos dos grupos, la lista de objetos grandes y la lista de objetos más pequeños, constituyen los datos básicos de un problema de corte de material. Las ordenes de tubos pequeños se

combinan entre sí para formar patrones de corte que son asignados luego a los tubos más grandes. Este proceso debe de obedecer ciertos objetivos y restricciones específicos para el problema [8].

Como segundo ejemplo, tenemos un problema de carga de contenedores. Este tipo de problema también tiene dos grupos básicos de datos. Por una parte se tiene una lista de objetos grandes que puede ser uno o más contenedores. Y por otra parte se tiene una lista de objetos pequeños que deben ser empacados en los contenedores. Además de ciertos objetivos y restricciones dados, el aspecto principal de la carga de contenedores concierne a la combinación geométrica de objetos pequeños en patrones de empaado, los cuales son asignados a los contenedores de la lista [8].

Por lo tanto la estructura lógica común de los problemas de C&P puede ser determinada como sigue:

1. Hay dos grupos básicos de datos cuyos elementos definen cuerpos geométricos o formas definidas en una o más dimensiones del espacio de los números reales:
  - Por una parte, la lista de objetos grandes, y
  - por otra parte, la lista de objetos pequeños.
2. El proceso de corte y empaado encuentra patrones los cuales son combinaciones geométricas de objetos pequeños asignados a objetos grandes. Las partes sobrantes, por ejemplo, las figuras resultantes en patrones que no pertenecen a objetos pequeños, se tratan comúnmente como “desperdicio” [8].

La estructura lógica de los problemas de C&P proporciona un esquema para una sistematización más profunda la cual indica que estos problemas pueden verse como:

- La lista de objetos grandes
- La lista de objetos pequeños
- Los patrones como combinaciones geométricas de figuras pequeñas para una figura grande, o
- La asignación de objetos pequeños a patrones así como patrones a objetos grandes.

### **3.2 Características Principales de los Problemas de C&P**

A continuación se presentan la características que definen un problema de C&P como lo propone Dyckhoff [8]. Estas características son: Dimensionalidad, Medición

de la Cantidad, Forma de las Figuras, Disponibilidad, Restricciones de Patrones, Restricciones de Asignación y Objetivos de los Problemas de C&P.

### 3.2.1 Dimensionalidad

La característica más importante de los problemas de corte y empackado es la dimensionalidad. En lugar de considerarla por separado de los objetos grandes por una parte y los pequeños por otra, se atribuye al problema o más precisamente a los patrones. La dimensionalidad es el número mínimo de dimensiones de números reales necesarios para describir la geometría de los patrones. Los tipos elementales son:

- una dimensión,
- dos dimensiones,
- tres dimensiones, y
- problemas de múltiples dimensiones (más de tres dimensiones).

Se pueden obtener problemas de cuatro dimensiones cuando los problemas de empackado en tres dimensiones tienen al tiempo como una cuarta dimensión, por ejemplo, cuando las cajas deben empackarse en el contenedor durante un período de tiempo sin interrupciones (o cuando un pan está en el horno) [8].

Aunque pareciera sin importancia clasificar un problema con respecto a sus dimensiones, en este caso no lo es. Por ejemplo, la carga de palets se considera como un problema de dos dimensiones. Sin embargo, la altura del palet podría considerarse como una tercera dimensión. De manera similar, los contenedores con frecuencia son cargados primeramente, construyendo pilas verticales y luego colocando las pilas horizontalmente en la base de los contenedores. En cualquiera de los casos podríamos hablar de dimensionalidad '2+1' en lugar de 3. Entonces, los problemas de corte de láminas de vidrio usando guillotina podrían caracterizarse como un problema de '1+1' dimensiones [8].

### 3.2.2 Medición de la Cantidad

Otra característica principal de los problemas de corte y empackado es la forma de medir la cantidad de los objetos grandes y de los objetos pequeños, respectivamente. Se pueden distinguir dos casos:

- Medición discreta, por ejemplo, números naturales, y
- Medición continua, por ejemplo, números reales.

**614196**

El primer caso se refiere a la frecuencia o número de objetos de cierta forma. Por otra parte, las cantidades fraccionarias miden la longitud total de varios objetos que tienen la misma forma con respecto a las dimensiones relevantes, se suman las longitudes de los objetos para formar otra dimensión que no es esencial para la geometría de los patrones. En lugar de la longitud puede ser también el peso o el diámetro, por ejemplo, de rollos. La combinación de problemas de una dimensión con mediciones continuas se denomina “problema de una y media dimensión” [8].

### 3.2.3 Forma de las Figuras

Otra característica principal de los problemas de C&P relacionada con la dimensionalidad, es la forma de las figuras de los objetos grandes y pequeños. La figura de un objeto se define como la representación geométrica en el espacio, de las dimensiones relevantes. Por lo tanto una figura se determina por:

- forma,
- tamaño, y
- orientación.

Las figuras de la misma forma difieren a lo mucho en su tamaño y orientación (o posición) en el espacio relevante. Para problemas con más dimensiones una de las características importantes es si las figuras tienen formas regulares o irregulares. La gran mayoría de los problemas existentes en la literatura, consideran objetos de formas regulares, especialmente en forma de rectángulos o bloques. Sin embargo algunas industrias trabajan con objetos de formas no convexas y asimétricas, por ejemplo, la industria metalúrgica, textil, del calzado y la de madera [8].

Las figuras que difieren a lo mucho en su tamaño, pueden hacerse idénticas cambiando la escala de medición a una misma para todas las dimensiones relevantes. Dependiendo de la dimensionalidad, el tamaño de una figura puede medirse por su longitud, área, o volumen. Un aspecto importante que determina la facilidad o dificultad para resolver un problema específico, es el tamaño relativo de los objetos.

Las figuras de la misma forma y tamaño difieren a lo mucho con respecto a su orientación (y posición), por ejemplo, si son congruentes. Se pueden distinguir tres casos importantes con respecto a la orientación de los objetos:

- Si se permite alguna orientación, entonces no se deben distinguir del problema los objetos con figuras congruentes.

- Si sólo se permiten rotaciones de 90 grados, entonces sólo se consideran como idénticas las figuras respectivas.
- Si la orientación es establecida, entonces los objetos con figuras congruentes son diferenciados, excepto aquellos que pueden hacerse idénticos con traslaciones.

Con estas definiciones se puede decir que todas las figuras de una dimensión tienen las mismas formas y orientaciones: Sólo difieren en sus tamaños, por ejemplo, sus longitudes. Los círculos y cuadrados son ejemplos de formas en dos dimensiones definidas, mientras que los rectángulos que tienen anchos y largos distintos se consideran formas distintas [8].

### 3.2.4 Disponibilidad

La disponibilidad de los objetos de figuras permitidas se refiere a:

- Límites inferior y superior de cantidad,
- su secuencia u orden, y
- la fecha cuando un objeto puede ser o tiene que ser cortado o empacado.

Se puede distinguir entre un número infinito o finito de objetos. En el segundo caso podrían haber muchos o pocos objetos. Los problemas clásicos de empacado de *bins* y el corte de material consideran típicamente un número infinito de objetos que son todos de la misma figura grande; sin embargo, existen pocos objetos de cada figura pequeña (o muchos pero diferentes) para el caso del empacado de bins mientras que hay muchos de los objetos de cada una de las figuras pequeñas en el problema de corte de material (o relativamente pocas diferentes). Los problemas de carga de palets y de la mochila se caracterizan por tener un solo objeto grande.

El número de objetos dados se ha interpretado como un límite superior de disponibilidad para el proceso de corte y empacado. En la mayoría de los casos existen límites inferiores, los cuales pueden ser iguales a los límites superiores. Por ejemplo, se debe empacar un objeto grande de un problema de empacado de palets o del problema de la mochila. Inversamente, todos los objetos dados de un problema clásico de empacado de bins o de corte de material tienen que ser empacados (dentro de bins) o cortados (de algún material). Los límites inferiores o superiores distintos pueden representar niveles de tolerancia de exceso y deficiencia con respecto a la orden de objetos.

Otras propiedades con respecto a la disponibilidad se obtienen cuando existe un cierto orden para objetos, lo cual ocurre cuando el tiempo juega un rol importante. Un orden parcial de objetos pequeños tiene que tomarse en cuenta para el balance de

una línea de ensamble. Las barras de acero caliente deben cortarse no sólo en una determinada secuencia, sino que también sin ningún retraso. Los objetos pequeños que deben ser empacados podrían tener fechas distintas de entrega que son importantes para enviar los contenedores [8].

### 3.2.5 Restricciones de Patrones

Las características discutidas hasta el momento, se refieren a las propiedades de los objetos grandes y de objetos análogos pequeños, respectivamente. La mayoría de ellos tienen impactos inmediatos en los patrones como combinaciones geométricas, y en la asignación de objetos pequeños a objetos grandes. En el primer caso, la construcción de patrones individuales es especificado, resultando en restricciones geométricas y combinatorias. En el caso de la segunda restricción relativa al número, el orden, o la combinación de patrones tienen que ser tomados en cuenta.

Se pueden distinguir cuatro grupos importantes de restricciones, las cuales pueden ser consideradas para la construcción de combinaciones geométricas de figuras pequeñas y figuras grandes:

- Las distancias mínima y máxima entre las figuras pequeñas o entre cortes que dividen figuras grandes son con frecuencia importantes, por ejemplo, el corte de vidrio o la carga de contenedores.
- La orientación de las figuras pequeñas relativas a cada una y/o a figuras grandes deben ser tomadas en cuenta, por ejemplo, en fábricas de patrones de corte o cuando se cargan objetos delicados en palets.
- Pueden haber restricciones con respecto a la frecuencia de objetos pequeños o grandes en un patrón, especialmente relacionado a la combinación o al número de figuras pequeñas diferentes o al número de objetos pequeños, en total o con relación a ciertas figuras. Esto incluye, por ejemplo, cantidades máximas de desperdicio así como los números máximos de rangos de tamaños u ordenes.
- Es esencial el tipo y el número de cortes permitidos, particularmente si los objetos son de formas rectangulares o de bloques. Siguiendo esta misma línea, se distinguen los patrones no-ortogonales y ortogonales. Cuando se trata de cortes con guillotina la complejidad depende del número de cambios en las direcciones de corte, así como del número de cortes paralelos por etapa.

### 3.2.6 Restricciones de Asignación

Se puede considerar que la asignación de objetos pequeños a objetos grandes consiste en dos pasos sucesivos, el primero que es arreglar objetos pequeños dentro de

patrones convenientes, y el segundo que es la asignación de patrones a objetos grandes apropiados. Pueden haber restricciones relacionadas a:

- el tipo de asignación,
- el número de etapas,
- número, frecuencia, o secuencia de patrones, y
- dinámicas de asignación.

Una propiedad importante es el tipo de asignación. Tanto para los objetos grandes como para los objetos pequeños, tiene que ser claro si sólo una selección o todas ellas tienen que ser asignadas a patrones correspondientes. En principio, esta diferenciación nos lleva a cuatro posibles categorías. Dos categorías juegan un rol importante en la literatura. El problema de la mochila o *napsack* es un ejemplo de la categoría donde una selección de los objetos dados tiene que ser combinada con los patrones de tal forma que un patrón no trivial sea asignado a cada uno de los objetos dados. En la categoría opuesta, que es típica para los problemas de corte de material, todos los objetos tienen que ser asignados a una selección apropiada de los objetos [8].

El número de etapas se refiere a la cuestión de si los objetos pequeños deben o no ser cortados de o empacados dentro de objetos grandes simultáneamente en un paso o en varios pasos sucesivos. En el primer caso sólo los objetos grandes de los datos del problema original son admisibles, en el segundo caso, las figuras residuales de algunos patrones se convierten en figuras grandes de otros patrones. Este proceso de asignación de múltiples etapas puede tener *a priori* un número de etapas limitado o ilimitado [8].

El orden o la secuencia de patrones en el proceso de asignación puede restringirse, por ejemplo, inducido por el orden o la conexión de los objetos, o por la tecnología de los procesos de C&P. Además, las restricciones se refieren a la frecuencia de patrones, siendo esta el número de tipos de patrones diferentes o el número de patrones del mismo tipo [8].

La asignación de objetos pequeños a objetos más grandes debe ser de naturaleza estática o dinámica. Las asignaciones estáticas podrían ser sin embargo, procesos en línea, donde a diferencia de los procesos fuera de línea, los objetos se asignan sucesivamente sin una posible recolocación y sin conocer el siguiente objeto. En procesos dinámicos, juegan un rol importante los diferentes puntos o períodos de tiempo [8].

### 3.2.7 Objetivos de los Problemas de C&P

No siempre es posible decidir definitivamente si una característica es geométrica o combinatoria. Algunas incluyen ambos aspectos, algunas no. Los objetivos de los problemas de C&P con frecuencia tienen tanto características geométricas como combinatorias. Además, pueden ser atribuidas a los objetos, patrones o a los procesos de asignación. Un objetivo es un criterio a ser maximizado o minimizado. Los criterios por satisfacer con cierto nivel son tratados como restricciones del proceso de corte y empaçado, por ejemplo, la restricción de que sólo los objetos pertenecientes a la misma orden deben estar contenidos en un patrón [8].

Existen diferentes tipos de objetivos dependiendo de si se refieren a

- las cantidades de objetos grandes y pequeños, y las piezas residuales asignados a patrones,
- la geometría de los patrones (optimización de disposición), o
- la secuencia, combinación, o número de patrones.

Las funciones objetivo que son lineales con respecto a las cantidades, deben valorar los objetos por sus tamaños o más generalmente por sus precios. Los objetivos no lineales, por ejemplo, aparecen si el desperdicio de material es minimizado o si hay cargos asignados a los patrones. Típicamente para muchos problemas de C&P deben ser considerados muchos objetivos [8].

## 3.3 Resumen

Por lo tanto la estructura lógica común de los problemas de C&P pueden ser determinada como sigue:

1. Hay dos grupos básicos de datos cuyos elementos definen cuerpos geométricos o formas definidas en una o más dimensiones del espacio de los números reales:
  - Por una parte, la lista de objetos grandes, y
  - por otra parte, la lista de objetos pequeños.
2. El proceso de corte y empaçado encuentra patrones los cuales son combinaciones geométricas de objetos pequeños asignados a objetos grandes.

Además, existen otras características que deben ser consideradas cuando estamos definiendo un problema de C&P, tales como: su dimensionalidad, cantidad de medición, formas de las figuras, disponibilidad, restricciones de patrones, restricciones de asignamiento y objetivos específicos.



## Capítulo 4

# Algoritmos Genéticos para Optimización

En este capítulo se presenta una introducción a los Algoritmos Genéticos (AGs), dando una explicación de qué son, cómo funcionan y sus fundamentos matemáticos. Además, se describe el diseño e implementación del Algoritmo Genético de Empacado (AGE). El AGE es un método propuesto para resolver el problema de la carga de contenedores.

## 4.1 Fundamentos de los Algoritmos Genéticos

Un Algoritmo Genético es un subconjunto de la computación evolutiva y modela procesos biológicos para optimizar funciones de costo altamente complejas. Un Algoritmo Genético permite que una población compuesta de muchos individuos pueda evolucionar bajo reglas de selección específicas hasta un estado en el cual se maximice la “aptitud” (por ejemplo, la minimización de la función de costo). El método fue desarrollado por John Holland (1975) en el curso de los años 1960s y 1970s y una de las personas que ayudó a popularizarlo fue uno de sus estudiantes, David Goldberg [9].

Algunas de las ventajas de un Algoritmo Genético, como mencionan Randy y Sue Haupt [9] incluyen que:

- optimiza con parámetros continuos o discretos,
- no requieren información de derivadas,
- buscan simultáneamente en un amplio rango de la superficie de costo,
- maneja un número grande de parámetros,
- funciona muy bien con computadoras paralelas,
- optimiza parámetros de superficies de costo extremadamente complejas; pueden saltar de un mínimo local,

- proveen una lista de parámetros óptimos y no sólo una solución,
- pueden codificar los parámetros de tal forma que la optimización se realice con los parámetros codificados, y
- trabaja con datos generados numéricamente, datos experimentales, o funciones analíticas.

## 4.2 Cómo Funcionan los Algoritmos Genéticos

El Algoritmo Genético comienza como cualquier otro algoritmo de optimización, definiendo los parámetros de optimización y la función objetivo. Termina también como cualquier otro algoritmo, probando la convergencia. Sin embargo, por dentro este algoritmo es muy diferente de los otros algoritmos de optimización.

Básicamente, un Algoritmo Genético evoluciona un conjunto de soluciones hasta que se cumple con un criterio de terminación. Al conjunto de soluciones se le denomina población y a cada una de las posibles soluciones se les llama individuos. Cada individuo es representado por una cadena de elementos tomados de un alfabeto definido que generalmente resulta ser el alfabeto binario. Sin embargo, este alfabeto puede ser también el conjunto de los números enteros o de los números reales, dependiendo del problema que se quiera resolver. Además, el criterio de terminación debe ser suficientemente bueno para asegurar que una solución óptima sea generada antes de que el algoritmo detenga su ejecución.

Los pasos principales del Algoritmo Genético se ilustran en la Figura 4.1 y se describen a continuación:

1. Selección del tipo de representación

Primeramente se escoge un tipo de representación de los individuos de la población, la cual puede ser cadenas binarias, cadenas de números enteros o cadenas de números reales.

2. Generación de la población inicial

Como se mencionó anteriormente, una población es un conjunto de soluciones posibles y cada solución es codificada como una cadena. La población inicial debe ser generada escogiendo aleatoriamente un número finito de puntos pertenecientes al espacio de soluciones. A los individuos también se les conoce como cromosomas. El número de individuos que determinan el tamaño de la población, varía dependiendo de diversos factores que pueden ser por ejemplo: la complejidad del problema, la estructura de los cromosomas, el comportamiento del AG, etc.

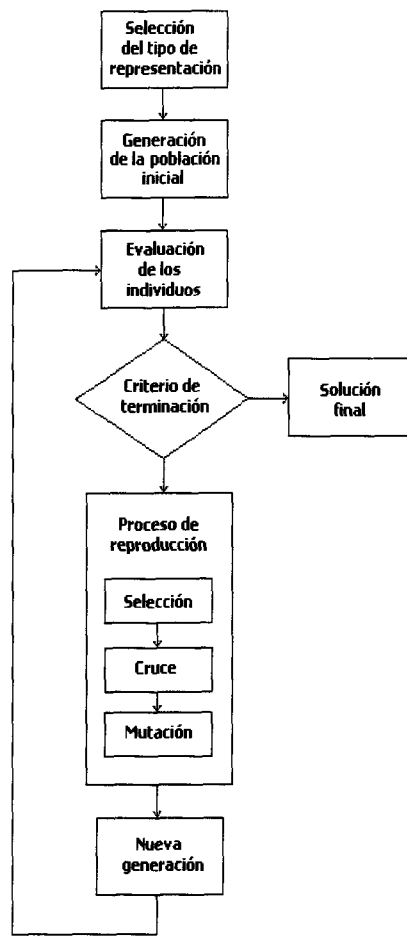


Figura 4.1: Diagrama de flujo del algoritmo genético.

### 3. Evaluación de los individuos

Una vez que la función objetivo ha sido definida y que la población inicial ha sido creada, el próximo paso es evaluar a cada individuo usando la función objetivo. Haciendo esto, se le asigna un valor de aptitud a cada individuo, el cual será utilizado para determinar si un individuo es bueno o malo, dependiendo del tipo de problema. Puede tratarse de maximización o minimización.

### 4. Proceso de reproducción

Hasta este punto del algoritmo, se ha generado una población de individuos con sus respectivas evaluaciones. Ahora, es tiempo de evolucionar esta población para obtener una nueva población de individuos mejorados. Para generar la siguiente población, se requieren tres acciones, las cuales son: selección, cruce y mutación.

#### (a) Selección

Existen unas cuantas funciones de selección que pueden ser usadas y las

más comunes son: rueda de ruleta, ranqueo y selección por torneo. Todas ellas se usan para seleccionar parejas de individuos con buenas evaluaciones, para posteriormente cruzarlos y crear nuevos individuos a partir de ellos, tratando de preservar de esta forma, las características que los hacen buenos individuos.

(b) Cruce

Una vez que todos los individuos se han puesto en parejas, se aplica el operador de cruce. Haciendo esto, se generan nuevos individuos llamados hijos. El cruce consiste en intercambiar elementos entre parejas de cromosomas para crear nuevos cromosomas. Este operador asegura que las siguientes generaciones serán una rica mezcla de las características de sus ancestros, hasta el momento en que uno de sus individuos se convierte en una solución óptima, después de algunas o muchas generaciones. El operador de cruce se aplica por lo general con una alta probabilidad de ocurrencia.

(c) Mutación

Después de cierto tiempo, la diversidad de la población comienza a disminuir hasta llegar un momento donde los cromosomas y sus valores de aptitud se convierten en el mismo, ocasionando que los mejores individuos obtenidos hasta el momento ya no mejoren. A esto se le llama convergencia. La convergencia prematura ocurre cuando el AG converge en una etapa temprana de su ejecución sin haber generado aún una solución buena. Por lo tanto, el operador de mutación se aplica para inyectar diversidad a la población, evitando de esta manera que ocurra una convergencia prematura. La mutación consiste en realizar un pequeño cambio en algunos cromosomas. Por ejemplo, en un cromosoma de estructura binaria, un buen operador de mutación podría consistir en cambiar un 0 por un 1 o viceversa, un 1 por un 0. El operador de mutación, en contraste con el operador de cruce, se aplica por lo general con una baja probabilidad de ocurrencia.

5. Repetir los pasos 3 y 4 hasta que se cumple con el criterio de terminación

Los pasos de evaluación del individuo y reproducción deben ejecutarse iterativamente a través de varias generaciones, permitiendo al algoritmo evolucionar los individuos, y producir una solución óptima. Algunos de estos criterios de terminación pueden ser:

- Un número determinado de generaciones
- Tiempo de ejecución del AG
- Convergencia

### 4.3 Fundamentos Matemáticos

Se puede entender la forma en la que trabaja el Algoritmo Genético analizando esquemas. Un esquema es un patrón de bits definido usualmente en el alfabeto  $\{0,1,*\}$ . El asterisco es un símbolo que puede representar un 0 o un 1 en una posición particular. El comportamiento del número esperado de esquemas representativos en una población a través de las generaciones es analizado. Un esquema es un conjunto de puntos en el espacio de búsqueda que tienen una característica en común definida como un patrón de bits, por ejemplo en el conjunto  $\{10101, 11111, 11101\}$  todos los elementos tienen en común que en las posiciones 1, 2 y 3 existe un 1, se puede decir entonces que el patrón en común es  $1*1*1$ , es decir el esquema que los contiene. Si el espacio de búsqueda es representado como un hipercubo de una dimensión, los esquemas son hiperplanos en el espacio de *hamming*, esta es la razón por la que los esquemas son llamados también hiperplanos. Si la relación de los puntos en el espacio es representada usando mapas de Karnaugh, los esquemas son grupos [17]. La evaluación de un esquema que se denota por  $f(H)$  es la evaluación promedio de todos los puntos en el espacio que pertenecen a ese esquema  $H$ . La varianza de un esquema, la cual se denota por  $\sigma^2(H)$ , es la varianza de la evaluación de todos los puntos que pertenecen a ese esquema. Los individuos de una población de un Algoritmo Genético que representa un esquema  $H$  constituye un ejemplo de este esquema. La evaluación promedio de estos individuos es una estimación de la evaluación verdadera del esquema  $H$ . Mientras menor es la varianza de un esquema y el tamaño de su muestra es mayor, mejor será la estimación de la evaluación verdadera del esquema. En un esquema, las posiciones que tienen 0 ó 1 se llaman posiciones definidas. El orden de un esquema  $H$ , denotado por  $o(H)$  es el número de posiciones definidas. La longitud de definición de un esquema  $H$ , denotada por  $\delta(H)$ , es el número de puntos de cruce entre la primera y la última posiciones definidas de un esquema. El teorema fundamental de los algoritmos genéticos, estima el número de individuos que pertenecen a un esquema dado, en la próxima generación, comenzando de una población base de individuos; se define como sigue [5]:

$$m(H, t + 1) \geq m(H, t) \frac{f(H)}{\sum \frac{f_i}{n}} \left[ 1 - p_c \frac{\delta(H)}{l - 1} - o(H)p_m \right]$$

Un esquema particular  $H$  recibe un número esperado de copias en la generación siguiente dependiendo principalmente de la evaluación de su función objetivo (el término en la sumatoria es la relación entre la aptitud del esquema y la aptitud promedio de la población), su longitud de definición, y la probabilidad de cruce, su orden de esquema y su probabilidad de mutación.

Un Algoritmo Genético asigna un número exponencialmente creciente de esquemas representativos a los que tienen las siguientes características [5]:

- Evaluación alta
- Orden pequeño
- Longitud de definición corto

## 4.4 Modelo de Solución del Problema de Empacado

El modelo propuesto para resolver el problema de empaado en tres dimensiones se ilustra en la Figura 4.2 y se describe a continuación:

Primeramente, tenemos un problema de empaado definido en un espacio de trabajo de tres dimensiones. Se tiene un conjunto de  $n$  contenedores  $C = \{c_1, c_2, \dots, c_n\}$  y un conjunto de  $m$  objetos  $I = \{i_1, i_2, \dots, i_m\}$  que deben ser empaados dentro de los contenedores. Los  $m$  objetos están agrupados en  $r$  conjuntos de objetos del mismo tamaño y forma. Esto significa que hay  $r$  conjuntos de objetos de mismo tipo  $T = \{t_1, t_2, \dots, t_r\}$ .

Cuando el problema de empaado involucra a más de un contenedor, es conveniente seleccionar una heurística para guiar el proceso de empaado, como se ilustra en el modelo de la Figura 4.2.

Después de definir el conjunto de parámetros del Algoritmo Genético de empaado (AGE), el algoritmo se ejecuta varias veces usando distintas semillas. Las semillas son números reales en el intervalo  $[0, 1]$  y se utilizan para inicializar la función aleatoria que usa el AG para generar su población inicial. Se calcula un porcentaje de las ejecuciones y se presentan finalmente como la solución óptima. La solución se imprime en un archivo y sobre la pantalla como una gráfico por computadora que muestra el resultado.

## 4.5 Diseño e Implementación del Algoritmo Genético para Empacado

Para resolver el problema, se diseñó un AG con características seleccionadas cuidadosamente siguiendo los principios de la teoría de Algoritmos Genéticos. A continuación se explican las principales características del AGE, tales como la función objetivo, la representación, y los operadores de recombinación.

### 4.5.1 Función Objetivo

Uno de los principales componentes de un Algoritmo Genético es la función objetivo. En esta investigación se implementaron y probaron tres funciones objetivos

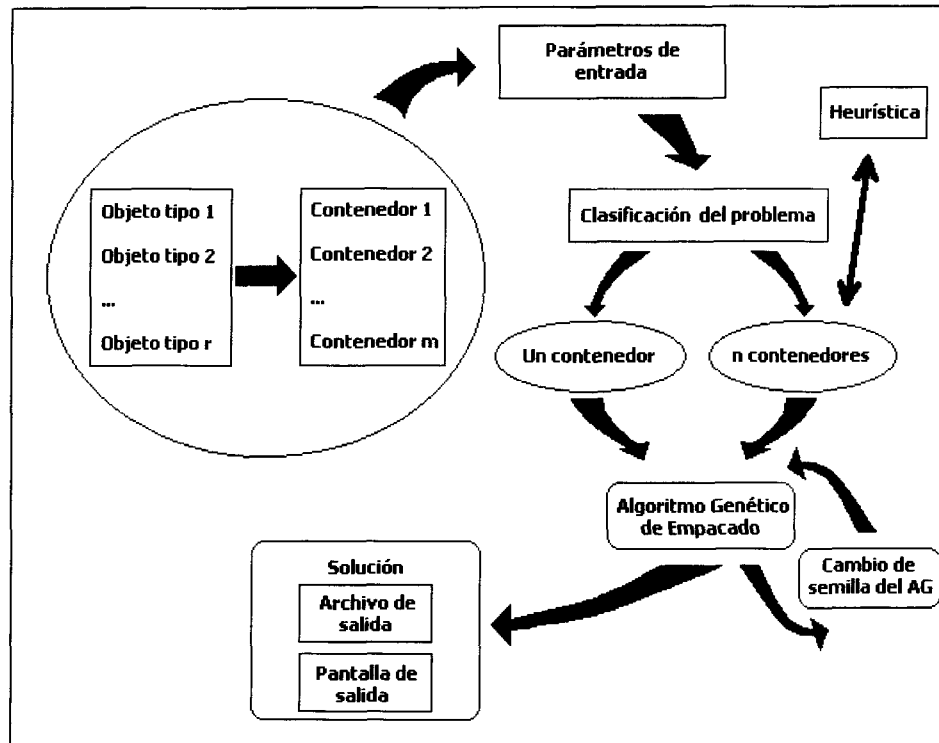


Figura 4.2: Modelo de solución del problema de empaquetado en tres dimensiones.

diferentes para encontrar la que mejor funciona con el problema de empaquetado definido previamente. Todas estas funciones objetivo son de maximización.

### **Función objetivo de volumen**

Esta función objetivo sólo toma en cuenta el número total de unidades cúbicas empaquetadas, y se define como sigue:

$$f(\Delta_k, s_k, r) = \omega \sum_{k=1}^r \Delta_k s_k \quad \text{donde,}$$

$$\Delta_k = l_k h_k w_k$$

$\Delta_k$  : Volumen del objeto tipo  $k$

$l_k$  : Largo del objeto tipo  $k$

$h_k$  : Alto del objeto tipo  $k$

$w_k$  : Ancho del objeto tipo  $k$

$s_k$  : Número de objetos empaquetados del tipo  $k$

$\omega$  : Factor de peso  
 $r$  : Número de tipos de objetos

### **Función objetivo de adyacencia**

Esta función objetivo toma en cuenta el volumen empacado tal como la función anterior lo hace, y además toma en cuenta la adyacencias existentes entre objetos del mismo tipo, para asegurar que sean empacados en lugares cercanos. Se define como sigue:

$$f(\Delta_k, s_k, r, p, q) = \omega \sum_{k=1}^r \Delta_k s_k + \beta p - \gamma q \text{ donde,}$$

$p$  : Número de adyacencias entre objetos del mismo tipo (se le suma uno a esta variable por cada par de objetos del mismo tipo que son empacados en posiciones adyacentes)  
 $q$  : Número de adyacencias entre objetos de distinto tipo (se le resta uno a esta variable por cada par de objetos de tipo distinto que son empacados en posiciones adyacentes)  
 $\beta$  : Constante de premio  
 $\gamma$  : Constante de castigo

### **Función objetivo de espacios vacíos**

Esta función objetivo considera tanto el volumen como la adyacencia y además toma en cuenta los espacios vacíos existentes entre cada par de objetos dentro del contenedor. Se define como sigue:

$$f(\Delta_k, s_k, r, p, q, e) = \omega \sum_{k=1}^r \Delta_k s_k + \beta p - \gamma q - \epsilon e \text{ donde,}$$

$e$  : Número de espacios vacíos entre objetos  
 $\epsilon$  : Constante de espacios vacíos

## **4.5.2 Representación del Cromosoma**

Una solución posible al problema de empacado consiste en un orden en el cual  $m$  elementos deben ser empacados dentro del contenedor. Esto permite trabajar con cromosomas de números enteros donde cada solución puede ser codificada como una permutación de  $m$  números. Por cada objeto que debe ser empacado existen tres elementos (Ver Figura 4.3). Esto significa que para un problema con  $m$  objetos, la



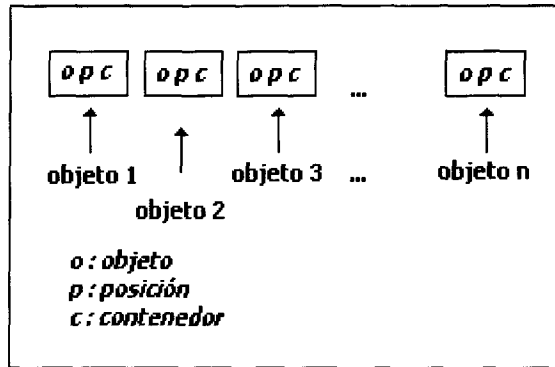


Figura 4.3: Estructura del cromosoma.

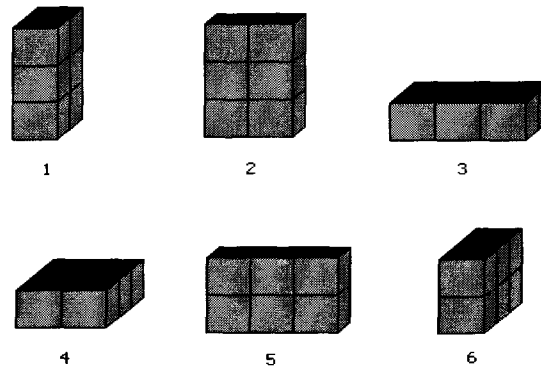


Figura 4.4: Posiciones posibles de un objeto cuando  $p = 6$ .

longitud del cromosoma utilizado es  $3m$ . El primer elemento es un número entre 1 y  $m$  y corresponde a uno de los  $m$  objetos. El segundo elemento es un número entre 1 y  $p$ , donde  $p$  es el número total de posiciones en las cuales puede ser colocado un objeto (Ver Figura 4.4). Finalmente, el tercer elemento es un número entre 1 y  $n$  e indica el contenedor en el cual debe ser empacado el objeto.

### 4.5.3 Operador de Selección

El operador de selección que se usa en esta investigación es el de selección por torneo de tamaño  $l$  el cual consiste de los siguientes pasos:

1. Revolver la población para que cada individuo tome una posición aleatoria.
2. Realizar tantos torneos como sea el tamaño de la población. Los torneos se forman con individuos colocados en posiciones adyacentes. Los individuos que participan en estos torneos se obtienen moviendo una ventana de tamaño  $l$  sobre la población, asumiendo que el último elemento del arreglo es adyacente al elemento de la primera posición.

3. Se selecciona un ganador para cada torneo, el cual es el mejor individuo. Entonces se asigna una copia de este individuo al grupo de apareamiento.

#### 4.5.4 Operadores de Cruce

Se implementaron dos operadores de cruce para determinar cuál funciona mejor con el problema, los cuales son el *PMX* y el *ERX*. Estos tipos de cruce se utilizan cuando se trabaja con cromosomas de números enteros

##### Operador *PMX*

En *PMX* (cruce parcialmente mapeado), se alinean dos cadenas (permutaciones y sus respectivos alelos), y se escogen dos puntos de cruce aleatoriamente a lo largo de las cadenas. Estos dos puntos definen una sección de unión que es usada para efectuar una operación de intercambio de posición por posición [5]. Para ver esto considere las siguientes dos cadenas:

$$A = 9\ 8\ 4\ .\ 5\ 6\ 7\ .\ 1\ 3\ 2\ 10$$

$$B = 8\ 7\ 1\ .\ 2\ 3\ 10\ .\ 9\ 5\ 4\ 6$$

Las cadenas se encuentran alineadas y los dos puntos de cruce aleatorio se encuentran ya establecidos. Se hace entonces un mapeo de *B* a *A*, donde los puntos de intercambio correspondientes son 5 con 2, 3 con 6, y 10 con 7. Al hacer el intercambio las cadenas hijo *A'* y *B'* quedan:

$$A' = 9\ 8\ 4\ .\ 2\ 3\ 10\ .\ 1\ 6\ 5\ 7$$

$$B' = 8\ 10\ 1\ .\ 5\ 6\ 7\ .\ 9\ 2\ 4\ 3$$

##### Cruce *ERX*

El operador de recombinación de aristas (*ERX*) utiliza un mapa de adyacencias para construir un hijo que herede tanta información como sea posible de las estructuras de sus padres. Este mapa de adyacencias almacena todas las conexiones de los dos padres. Considere las siguientes cadenas:  $[ABCDEF]$  y  $[BDCAEF]$ . El mapa de adyacencias de estas dos cadenas es el siguiente:

*A* tiene adyacencias con: *BFCE*  
*B* tiene adyacencias con: *ACDF*  
*C* tiene adyacencias con: *BDA*  
*D* tiene adyacencias con: *CEB*  
*E* tiene adyacencias con: *DFA*  
*F* tiene adyacencias con: *AEB*

*ERX* procede escogiendo una letra inicial. En el caso de ejemplo, *A* y *B* tienen 4 adyacencias, así que el algoritmo hará una selección aleatoria. Supóngase que se selecciona a *B*. Los candidatos para la siguiente letra son *A*, *C*, *D* y *F*. *C*, *D* y *F* tienen 2 adyacencias (Las tres iniciales menos la *B* - *A* tiene ahora 3 adyacencias y por lo tanto ya no se considera) así que se escoge aleatoriamente entre *C*, *D* y *F*. Supóngase que se escoge *C*. *C* tiene ahora adyacencia con *A* y *F*, las cuales ya no tienen adyacencias. Se escoge aleatoriamente a *A*. Ahora *A* debe irse a *F*. El hijo resultante es *[BCDEAF]* y se compone completamente por adyacencias tomadas de sus dos padres [18].

#### 4.5.5 Operador de Mutación

El operador de mutación que se diseñó especialmente para este algoritmo, consiste en seleccionar aleatoriamente dos puntos en un individuo de la población y posteriormente intercambiar sus posiciones en el cromosoma. Además, el segundo elemento, correspondiente a la posición del objeto, se cambia aleatoriamente en ambas posiciones. Esta simple operación es necesaria para inyectar diversidad a la población. Dado que el operador de cruce provee suficiente combinación de esquemas, el operador de mutación debe ser muy sencillo. La Figura 4.5 ilustra un ejemplo de mutación.

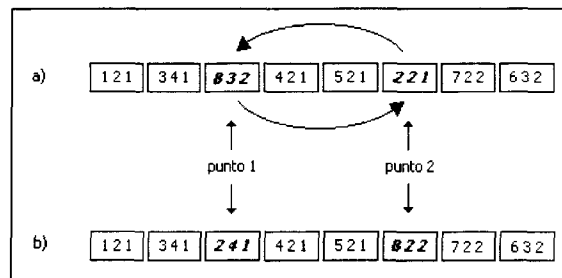


Figura 4.5: Un ejemplo de mutación.

#### 4.5.6 Criterio de Terminación

El criterio de terminación es el número de evaluaciones. La población es tan rica en diversidad que trabajar con los parámetros apropiados asegura que no ocurra convergencia prematura y que el tiempo necesario para realizar todas las evaluaciones necesarias antes de que se detenga al algoritmo sea apropiado.

### 4.6 Resumen

En este capítulo, se explicaron los aspectos básicos de los Algoritmos Genéticos, así como sus fundamentos matemáticos. Después, se describió un modelo de solución para

el problema de empaado. El AGE es un Algoritmo Genético diseado para proveer soluciones a problemas de empaado donde los objetos tienen forma de paralelepedros. Este algoritmo utiliza cromosomas de numeros enteros, con operadores de cruce especializados en problemas de permutaciones. El AGE puede aplicarse para solucionar problemas de empaado que involucren mas de un contenedor.

## Capítulo 5

# Resultados Experimentales

El Algoritmo Genético para Empacado (AGE) es un Algoritmo Genético diseñado e implementado para resolver el problema de carga de contenedores. En la primera sección de este capítulo se explican los experimentos realizados para determinar el conjunto más adecuado de parámetros de entrada del AGE. El conjunto de parámetros de entrada incluye el tamaño de la población, el número de generaciones, las probabilidades de cruce y mutación, así como también el tamaño del torneo para el operador de selección. También se muestra algunas comparaciones estadísticas entre los comportamientos de los operadores de cruce PMX y ERX, las cuales permiten determinar cuál de estos operadores proporciona mejores resultados cuando se aplican para resolver el problema de empacado. Posteriormente, se presentan tres funciones objetivo distintas y se explica cómo la primera de ellas se fue modificando hasta obtener la función objetivo final y que proporciona mejores resultados. También, se presentan algunos resultados experimentales relacionados con el tiempo que necesita el AGE para proporcionar sus soluciones. Finalmente, se explican dos métodos distintos para resolver el problema de empacado con más de un contenedor, mostrando su utilidad en diferentes situaciones reales.

### 5.1 Ajuste del Algoritmo Genético para Empacado

Una vez implementados todos los componentes del AGE, el siguiente paso consistió en encontrar el mejor conjunto de parámetros básicos de entrada, incluyendo:

- tamaño de la población,
- número de generaciones,
- operador y probabilidad de cruce,
- probabilidad de mutación,
- y tamaño del torneo del operador de selección.

Dado que no existe regla alguna para determinar el conjunto correcto de parámetros de entrada de un AG, esto debe hacerse mediante experimentación, usando distintos valores para esos parámetros, hasta que se obtiene un conjunto aceptable de parámetros. En este caso, fue necesario definir una instancia representativa del problema de empaclado y usarla para correr el AGE muchas veces. El problema de empaclado utilizado para ajustar el AGE se define a continuación.

### Problema de empaclado utilizado con fines experimentales

El problema consiste en empaclar 55 objetos pequeños dentro de un contenedor. Existen 11 tipos de objetos, todos ellos con formas de paralelepípedo y dimensiones diferentes, como se ilustra en la Figura 5.1 y se indica en la Tabla 5.1. El contenedor tiene las siguientes dimensiones: *ancho* = 9, *largo* = 9 y *alto* = 9. Además, cada objeto puede ser empaclado en dos posiciones posibles, las cuales son: su posición original o rotada 90° sobre su propia base. El total de las unidades cúbicas a empaclar es de 560.

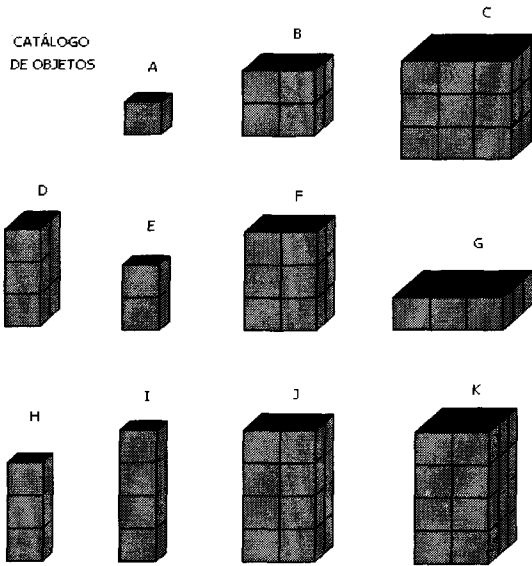


Figura 5.1: Catálogo de objetos utilizados con fines experimentales.

La Figura 5.2 ilustra las dos únicas posiciones en la que un objeto tipo *D* puede ser empaclado. La misma restricción se aplica a los demás tipos de objetos.

La función objetivo utilizada para ajustar el AGE es la llamada función objetivo de adyacencias. Básicamente, esta función objetivo toma en cuenta el número total de unidades cúbicas empacladas, así como también, el número de puntos adyacentes en el

Tipo	Ancho	Largo	Alto	Cantidad
A	1	1	1	5
B	2	2	2	5
C	3	3	3	5
D	1	2	3	5
E	1	1	2	5
F	2	2	3	5
G	3	3	1	5
H	1	1	3	5
I	1	1	4	5
J	2	2	4	5
K	2	3	4	5

Tabla 5.1: Dimensiones de los objetos utilizados con fines experimentales.

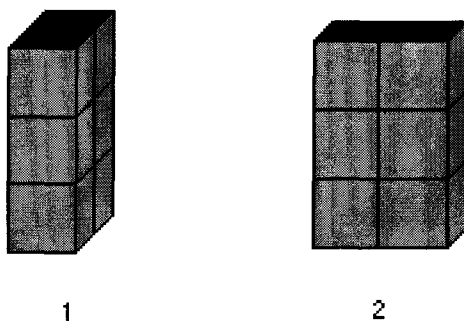


Figura 5.2: Posibles posiciones de los objetos tipo  $D$ .

cromosoma, que indican que dos objetos del mismo tipo deben ser colocados uno al lado del otro. Esto significa que el algoritmo cuenta cuantos objetos del mismo tipo son empacados juntos (de acuerdo a cada cromosoma) y luego asigna copias de los cromosomas con valores de adyacencia mayores al grupo de apareamiento. En otras palabras, se le dice al algoritmo genético que son preferidas las soluciones donde los objetos del mismo tipo son empacados juntos. La función objetivo de adyacencia es de la siguiente forma:

$$f(\Delta_k, s_k, r, p, q) = \omega \sum_{k=1}^r \Delta_k s_k + \beta p - \gamma q \text{ donde,}$$

$$\Delta_k = l_k h_k w_k$$

$\Delta_k$  : Volumen del objeto tipo  $k$

$l_k$  : Largo del objeto tipo  $k$

- $h_k$  : Alto del objeto tipo  $k$
- $w_k$  : Ancho del objeto tipo  $k$
- $s_k$  : Número de objetos empacados del tipo  $k$
- $p$  : Número de puntos adyacentes entre objetos del mismo tipo
- $q$  : Número de puntos adyacentes entre objetos de diferente tipo
- $\omega$  : Factor de peso
- $\beta$  : Constante de premio
- $\gamma$  : Constante de castigo
- $r$  : Número de tipos de objetos

Se realizaron experimentos con distintas semillas generadoras de números aleatorios, con 3 probabilidades de cruce diferentes, 3 probabilidades de mutación y 2 operadores de cruce distintos. Los resultados se muestran en las Tablas 5.2 y 5.3. Todos estos experimentos se realizaron usando los siguientes parámetros que demostraron proporcionar los mejores resultados:

- un tamaño de población de 200 individuos,
- durante 25 generaciones y
- un tamaño de torneo de 4.

Operador de cruce		PMX								
		0.8			0.89			0.98		
Probabilidad de cruce		0.8			0.89			0.98		
Probabilidad de mutación		0.003	0.0098	0.02	0.003	0.0098	0.02	0.003	0.0098	0.02
Semillas	0.02	4743	5052	4953	4746	4701	5214	4980	5043	4893
	0.16	5160	4797	4953	4962	5079	4746	5364	4776	4926
	0.23	5001	5367	5442	5292	5115	5334	5415	5160	5091
	0.39	4593	4884	4917	5025	5283	4923	5160	4518	4785
	0.45	5703	5679	5121	4863	5346	5202	5166	5469	4809
	0.52	5214	4524	4548	5364	4935	4770	5019	4971	4500
	0.68	4935	5088	5325	4845	5220	4914	4749	5142	4626
	0.74	4767	4548	4833	4857	4626	4440	4692	4902	5043
	0.88	5361	4839	5364	4845	5199	4899	4899	4716	4893
0.93	4752	4596	4725	5208	5205	4698	4776	4797	4569	
Promedios		5022.9	4937.4	5018.1	5000.7	5070.9	4914	5022	4949.4	4813.5

Tabla 5.2: Resultados experimentales empleando el operador  $PMX$ .



Operador de cruce		ERX								
Probabilidad de cruce		0.8			0.89			0.98		
Probabilidad de mutación		0.003	0.0098	0.02	0.003	0.0098	0.02	0.003	0.0098	0.02
Semillas	0.02	6495	5976	5871	6810	5973	6216	6204	6585	6522
	0.16	6495	6165	5958	6189	6249	6090	6495	6201	6720
	0.23	5472	6078	6507	6495	6036	5961	6432	5772	6018
	0.39	6480	5847	5778	6234	6045	5964	6126	6765	6720
	0.45	5790	6168	5856	5940	6294	6684	6126	6630	6333
	0.52	6360	6099	6495	6675	6204	6399	6396	6510	6135
	0.68	6210	5667	6366	5952	6285	6339	6765	5991	6648
	0.74	6180	6201	6099	6405	6033	6390	6990	6135	6420
	0.88	6129	6048	5958	6372	6297	5991	6450	6426	6459
0.93	5841	6225	6198	5964	6351	5997	5871	6855	6675	
Promedios		6145.2	6047.4	6108.6	6303.6	6176.7	6203.1	6385.5	6387	6465

Tabla 5.3: Resultados experimentales empleando el operador *ERX*.

Los resultados experimentales en la Tablas 5.2 y 5.3 indican que el operador *ERX* es mucho mejor que *PMX*. Es claro que aún el mejor resultado proporcionado por *PMX*, 5070.9, es peor que el peor resultado que proporciona el *ERX* de 6145.2, según los experimentos realizados. Finalmente se puede decir de Tablas 5.2 y 5.3 que el mejor conjunto de parámetros para cada operador de cruce implementado, de acuerdo a los experimentos realizados, es:

1. Para *ERX*:

- Tamaño de la población = 200
- Número de generaciones = 70
- Probabilidad de cruce = 0.98
- Probabilidad de mutación = 0.02
- Tamaño del torneo = 4

2. Para *PMX*:

- Tamaño de la población = 200
- Número de generaciones = 70
- Probabilidad de cruce = 0.89
- Probabilidad de mutación = 0.0098
- Tamaño del torneo = 4

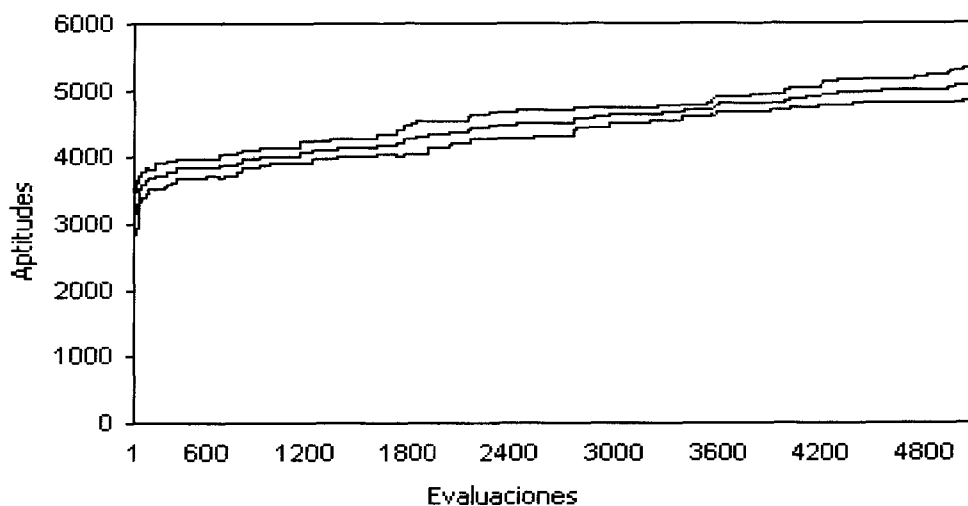


Figura 5.3: Curva de aptitud usando el operador *PMX* (Evaluación máxima = 5070.9).

En la Figuras 5.3 y 5.4 se graficaron las mejores columnas de resultados de las Tablas 5.2 y 5.3 respectivamente. La curva en el centro corresponde al promedio de la aptitud y las curvas inferior y superior indican la desviación estándar.

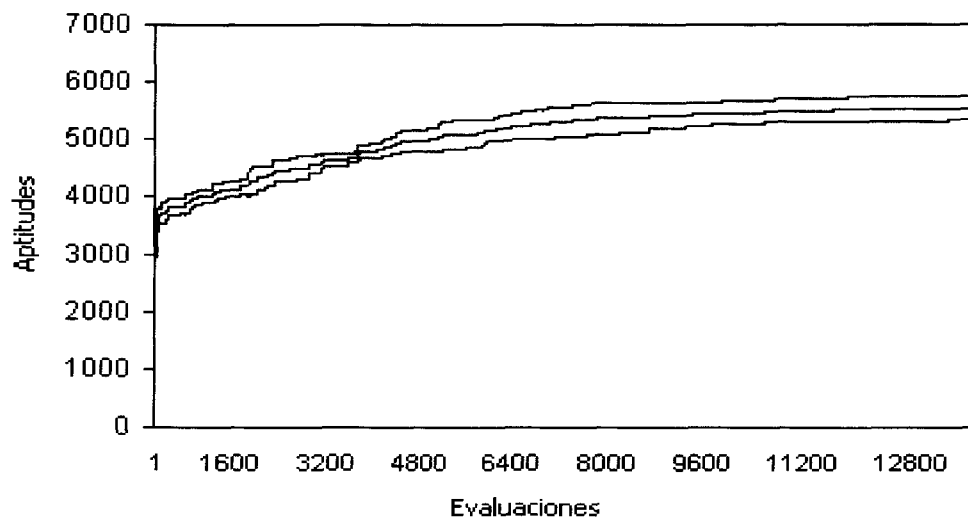


Figura 5.4: Curva de aptitud usando el operador *ERX* (Evaluación máxima = 6465).

Estas dos gráficas proporcionan una muy buena idea de lo importante que es escoger un buen operador de cruce. Aunque el comportamiento de la desviación estándar es buena en ambos casos, es obvio que el operador *ERX* proporciona mejores soluciones, con evaluaciones mayores que las proporcionadas por el operador *PMX*.

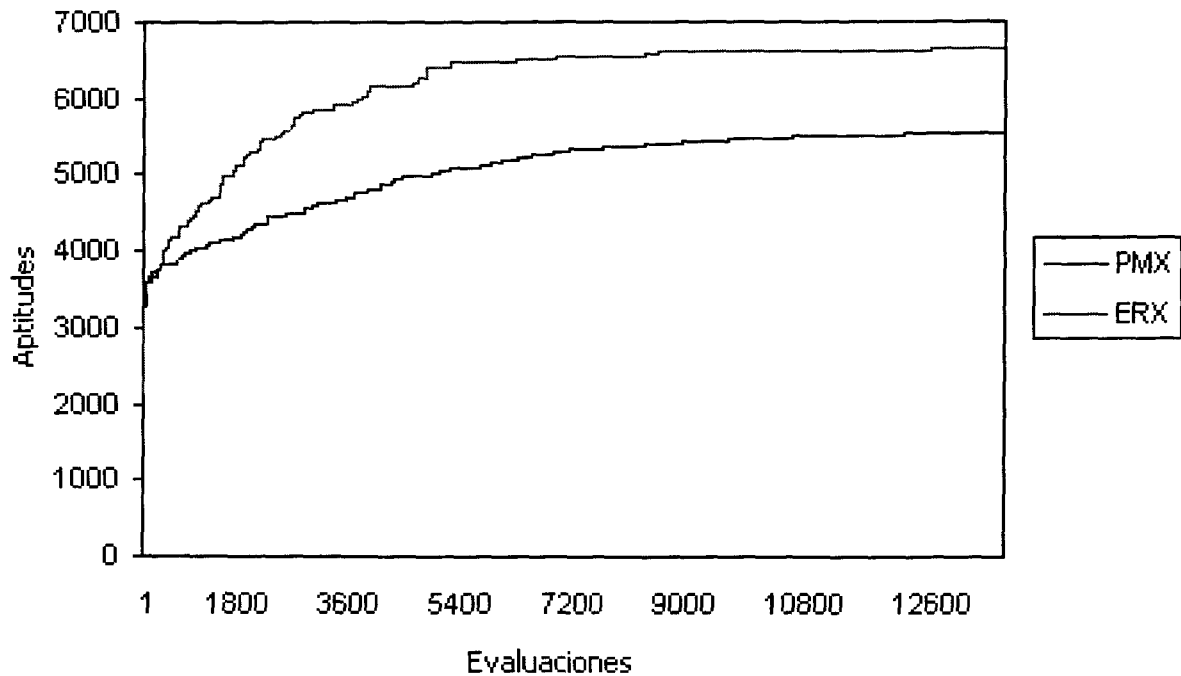


Figura 5.5: Curvas de aptitud utilizando los operadores *PMX* y *ERX* durante 70 generaciones.

La Figura 5.5 presenta una comparación entre las curvas de aptitud generadas por *PMX* y *ERX*, cuando se aplican al mismo problema con los siguientes parámetros:

1. Curva superior

- Tamaño de la población = 200
- Número de generaciones = 70
- Operador de cruce = *ERX*
- Probabilidad de cruce = 0.98
- Probabilidad de mutación = 0.02
- Tamaño del torneo = 4

2. Curva inferior

- Tamaño de la población = 200
- Número de generaciones = 70
- Operador de cruce = *PMX*
- Probabilidad de cruce = 0.89

- Probabilidad de mutación = 0.0098
- Tamaño del torneo = 4

Las mejores soluciones obtenidas durante esta parte inicial de la experimentación, usando *PMX* y *ERX*, se ilustran en las Figuras 5.6 y 5.7 respectivamente. Estos resultados se obtuvieron utilizando los mismos parámetros mencionados anteriormente.

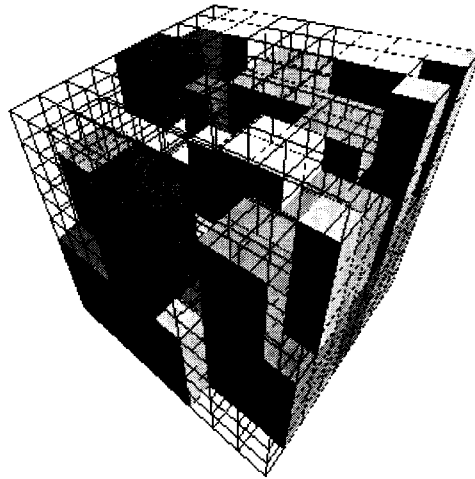


Figura 5.6: Una solución buena utilizando el operador *PMX* (Evaluación máxima = 5070.9).

Todos los objetos fueron empacados por completo en ambas soluciones, sin embargo, la solución en la Figura 5.7 tiene una evaluación mayor debido a que los objetos del mismo tipo fueron empacados en lugares cercanos unos a otros. En la figura, los objetos del mismo tipo aparecen del mismo color.

Las soluciones que proporciona el operador *ERX* lucen más compactas que las proporcionadas por el *PMX*.

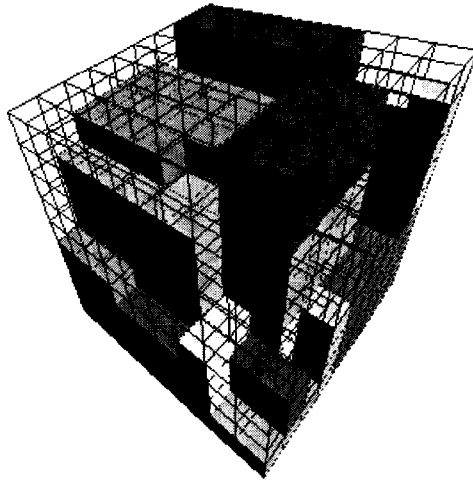


Figura 5.7: Una solución buena utilizando el operador *ERX* (Evaluación máxima = 6465).

## 5.2 Problemas de Empacado con Un Contenedor

La función objetivo es una de las partes más importantes de un AG, dado que se utiliza para evaluar a cada individuo y asignarle un valor de aptitud. El valor de aptitud que se le da a cada individuo, indica si es un buen individuo o no, y si sus características deben sobrevivir al paso de las generaciones.

En esta sección se presentan las modificaciones que sufrió la primera función objetivo implementada para resolver el problema de empaçado, hasta que se convirtió en la función objetivo que finalmente resultó proporcionar mejores resultados. Las tres funciones objetivos probadas son las denominadas: “función objetivo de volumen”, “función objetivo de adyacencias” y “función objetivo de espacios vacíos”. Los experimentos realizados con estas tres funciones objetivos, se hicieron utilizando el problema definido previamente, el cual consiste en empaçado 55 objetos dentro de un contenedor. En términos más específicos el problema a resolver consiste en empaçado 560 unidades cúbicas, con un máximo posible de 44 puntos de adyacencia entre objetos del mismo tipo, lo cual equivale a empaçado toda la mercancía de forma ordenada, y un máximo de 169 unidades cúbicas que pueden ser desperdiciadas entre las cajas, originando soluciones no compactas.

### 5.2.1 Función Objetivo de Volumen

La función objetivo de volumen toma en cuenta únicamente el volumen total de mercancía empaçada para calcular el valor de aptitud. Esta fue la primera función objetivo implementada para evaluar a la población del AGE y fue muy útil para verificar

que cada uno de los componentes del algoritmo funcionen adecuadamente. La función objetivo de volumen se define como sigue:

$$f(\Delta_k, s_k, r) = \omega \sum_{k=1}^r \Delta_k s_k$$

$$\Delta_k = l_k h_k w_k$$

- $\Delta_k$  : Volumen del objeto tipo  $k$
- $l_k$  : Largo del objeto tipo  $k$
- $h_k$  : Alto del objeto tipo  $k$
- $w_k$  : Ancho del objeto tipo  $k$
- $s_k$  : Número de objetos empacados tipo  $k$
- $\omega$  : Factor de peso
- $r$  : Número de tipos de objeto

Sin embargo, después de realizar varios experimentos con esta función objetivo, los resultados obtenidos no fueron tan buenos como se esperaba que fueran. Es una buena función objetivo dado que en los experimentos realizados todos los objetos fueron empacados en el contenedor. Sin embargo, los objetos son empacados en cualquier lugar del contenedor sin importar a que tipo de objeto pertenezca cada uno. Esto hace que los objetos del mismo tipo no sean empacados juntos, como se desearía que fuera, para facilitar el proceso de descarga. En otras palabras se puede decir que esta función objetivo proporciona una forma de empaado desordenada.

El valor de  $\omega$  es el peso que tendrá cada unidad cúbica durante la evaluación de la función objetivo. Si aumentamos el valor de  $\omega$ , estamos aumentando la importancia que tiene cada unidad cúbica al ser empacada, lo cual tiene verdadera relevancia al momento de incluir en la función objetivo otras características que necesitamos maximizar. Con la función objetivo de volumen el peso utilizado fue de  $\omega = 12$ , el cual permite obtener soluciones con mayor número de unidades empacadas . En la Figura 5.8 se grafica la curva de aptitud y la desviación estándar, usando 10 semillas generadoras de números aleatorios diferentes. La evaluación mínima obtenida es 6,468 y la máxima es 6,720.

Parámetros utilizados:

- Tamaño de la población = 200
- Número de generaciones = 70

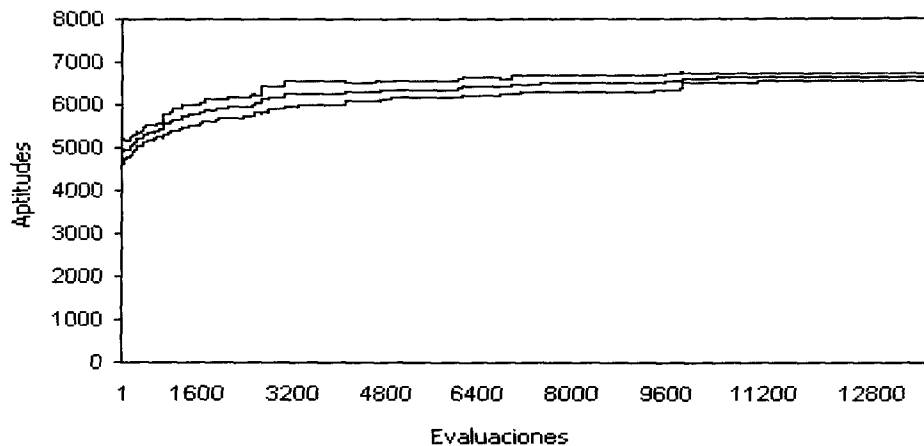


Figura 5.8: Curva de aptitud usando la función objetivo de volumen.

- Operador de cruce =  $ERX$
- Probabilidad de cruce = 0.98
- Probabilidad de mutación = 0.02
- Tamaño del torneo = 4

Se consideran tres valores para determinar si una solución es buena o no, al analizar una solución del AGE. El primer valor es el número total de unidades cúbicas empacadas en el contenedor. El segundo es el número total de adyacencias entre objetos del mismo tipo. Finalmente, el tercer valor considerado es el número total de unidades cúbicas vacías existentes entre los objetos empacados. En la Tabla 5.4 se muestran los puntajes obtenidos por el AGE cuando fue corrido para resolver el problema definido anteriormente, usando la función objetivo de volumen y 10 semillas distintas de generación de números aleatorios. En este caso los valores promedios son:  $volumen = 556$ ,  $puntos\ de\ adyacencia = 31$  y  $espacios\ vacíos = 154.5$ .

En la Figura 5.9 se muestra una de las mejores soluciones proporcionadas por el AGE con la función objetivo de volumen. Los objetos del mismo tipo aparecen del mismo color.

## 5.2.2 Función Objetivo de Adyacencia

La función objetivo de adyacencia fue implementada para generar soluciones de empaquetado ordenadas, es decir, soluciones donde se procure que los objetos pertenecientes

Función objetivo de volumen				
Operador de cruce		ERX		
Probabilidad de cruce		0.98		
Probabilidad de mutación		0.02		
Puntajes		Unidades cúbicas empacadas	Puntos de adyacencia	Espacios vacíos
Semillas	0.02	560	33	113
	0.16	560	32	150
	0.23	539	31	190
	0.39	547	35	173
	0.45	560	24	155
	0.52	560	30	159
	0.68	560	37	150
	0.74	554	26	156
	0.88	560	31	139
	0.93	560	31	157
Promedios		556	31	154.2

Tabla 5.4: Puntajes de empaqueo obtenidos con la función objetivo de volumen.

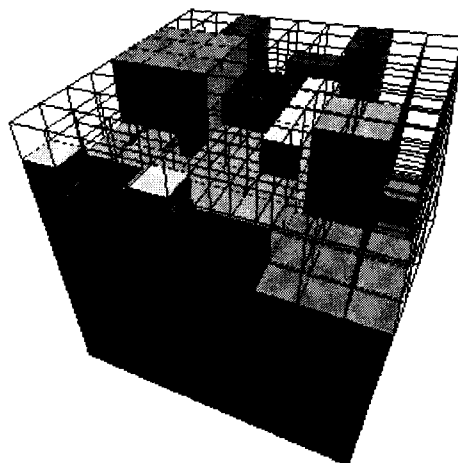


Figura 5.9: Una solución buena utilizando la función objetivo de volumen.



al mismo tipo sean empacados en posiciones cercanas. Esta función es básicamente la función de volumen más unas funciones de premio y castigo para lograr el propósito anteriormente mencionado. La función objetivo de adyacencia se define como sigue:

$$f(\Delta_k, s_k, r, p, q) = \omega \sum_{k=1}^r \Delta_k s_k + \beta p - \gamma q \text{ donde,}$$

$$\Delta_k = l_k h_k w_k$$

- $\Delta_k$  : Volumen del objeto tipo  $k$
- $l_k$  : Largo del objeto tipo  $k$
- $h_k$  : Alto del objeto tipo  $k$
- $w_k$  : Ancho del objeto tipo  $k$
- $s_k$  : Número de objetos empacados del tipo  $k$
- $p$  : Número de adyacencias entre objetos del mismo tipo
- $q$  : Número de adyacencias entre objetos de diferente tipo
- $\omega$  : Factor de peso
- $\beta$  : Constante de premio
- $\gamma$  : Constante de castigo
- $r$  : Número de objetos del mismo tipo

Primeramente, se cuenta el número total de unidades empacadas, después una rutina especial del algoritmo se encarga de recorrer el cromosoma de la primera posición a la última y contar los puntos de adyacencia existentes entre objetos del mismo tipo, así como también los puntos de adyacencia entre objetos de tipos diferentes. Por ejemplo, en un cromosoma indicando el siguiente orden de objetos:

A B\_B\_B A\_A B A

los puntos de adyacencia entre objetos del mismo tipo son aquellos indicados por un guión bajo y los puntos de adyacencia entre objetos de distinto tipo están indicados por espacios. Por lo tanto, para este ejemplo, los valores de  $p$  y  $q$  son:  $p = 3$  y  $q = 4$ . Además, los valores de  $p$  y  $q$  son multiplicados en la función objetivo por las constantes  $\beta$  y  $\gamma$  respectivamente, los cuales son números reales que fueron ajustados experimentalmente al igual que la  $\omega$  de acuerdo al comportamiento del algoritmo genético.

La Figura 5.10 muestra el comportamiento de la curva de aptitud y la desviación estándar generada por el AGE cuando fue corrido 10 veces, usando diferentes semillas para los números aleatorios y la función objetivo de adyacencia.

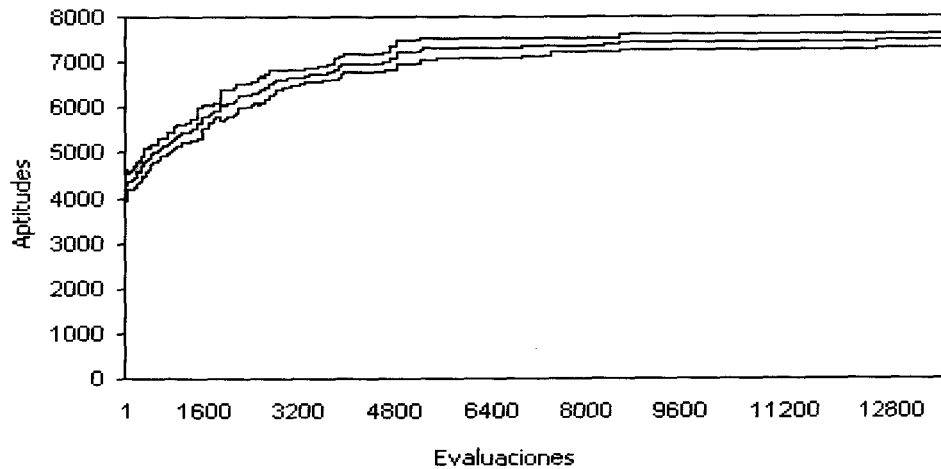


Figura 5.10: Curva de aptitud usando la función objetivo de adyacencia.

Los parámetros utilizados son:

- Tamaño de la población = 200
- Número de generaciones = 70
- Operador de cruce = *ERX*
- Probabilidad de cruce = 0.98
- Probabilidad de mutación = 0.02
- Tamaño del torneo = 4

Durante los experimentos realizados con la función de adyacencia, los mejores resultados obtenidos, aquellos donde era máximo el volumen empacado y de una forma ordenada, se obtuvieron haciendo  $\beta = 30$  y  $\gamma = 15$ . La evaluación final más baja obtenida fue 7,143 y la mejor evaluación fue 7,665.

Al igual que con la función de volumen, para esta función se calcularon los puntajes de volumen, adyacencia y espacios vacíos, para determinar que tan buenas son las soluciones. Estos puntajes calculados que nos indican la calidad de las soluciones se presentan en la Tabla 5.5.

Como lo muestran las estadísticas en la Tabla 5.5, las funciones de castigo y recompensa hacen un buen trabajo, aumentando el número de adyacencias entre objetos del mismo tipo. Las soluciones proporcionadas por esta función objetivo son soluciones

Función objetivo de adyacencia				
Operador de cruce		ERX		
Probabilidad de cruce		0.96		
Probabilidad de mutación		0.02		
Puntajes		Unidades cúbicas empacadas	Puntos de adyacencia	Espacios vacíos
Semillas	0.02	560	35	160
	0.16	560	36	122
	0.23	560	39	157
	0.39	560	36	150
	0.45	524	37	193
	0.52	550	38	169
	0.66	539	40	177
	0.74	548	38	180
	0.88	547	35	170
	0.93	560	35	148
Promedios		550.8	36.9	162.6

Tabla 5.5: Puntajes de empaqueo obtenidos con la función objetivo de adyacencia.

donde los objetos son empacados de una manera mejor ordenada que en las soluciones proporcionadas por la función objetivo de volumen. Sin embargo, el promedio de las unidades empacadas disminuye un poco. Esto puede considerarse como una desventaja de esta función, aún así, en general, sus soluciones siguen siendo mejores.

En la Figura 5.11, se muestra una de las mejores soluciones proporcionadas por el AGE al usar la función objetivo de adyacencia. Los objetos del mismo tipo aparecen del mismo color.

### 5.2.3 Función Objetivo de Espacios Vacíos

La función objetivo de adyacencia pudo resolver el problema de empaquetar objetos del mismo tipo en lugares cercanos unos de otros. Sin embargo, hay otro problema existente en las soluciones hasta ahora obtenidas. Este problema es que los objetos son empacados formando columnas en lugar de ser acomodados de una forma más homogénea que garantice estabilidad de la mercancía al momento de ser transportada. Este detalle fue corregido implementando una nueva función objetivo que se encarga de minimizar el número de unidades cúbicas existentes entre objeto y objeto una vez que estos han sido empacados. Esta nueva función es la función objetivo de espacios vacíos y se define como sigue:

$$f(\Delta_k, s_k, r, p, q, e) = \sum_{k=1}^r \Delta_k s_k \omega + \beta p - \gamma q - \epsilon e \text{ donde,}$$

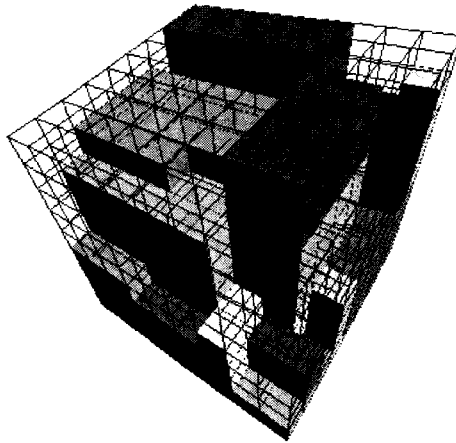


Figura 5.11: Una solución buena utilizando la función objetivo de adyacencia.

$$\Delta_k = l_k h_k w_k$$

- $\Delta_k$  : Volumen del objeto tipo  $k$
- $l_k$  : Largo del objeto tipo  $k$
- $h_k$  : Alto del objeto tipo  $k$
- $w_k$  : Ancho del objeto tipo  $k$
- $s_k$  : Número de objetos empacados del tipo  $k$
- $p$  : Número de puntos de adyacencia entre objetos del mismo tipo
- $q$  : Número de puntos de adyacencia entre objetos de tipos diferentes
- $e$  : Número de espacios vacíos entre objetos
- $\omega$  : Factor de peso
- $\beta$  : Constante de premio
- $\gamma$  : Constante de castigo
- $\epsilon$  : Constante de espacios vacíos
- $r$  : Número de tipos de objetos

La nueva función objetivo es básicamente la función de adyacencia más una función de castigo, la cual se encarga de reducir el número de unidades cúbicas que permanecen sin ocupar entre los objetos empacados. Esta función de castigo reduce el valor de aptitud de un individuo,  $\epsilon$  unidades por cada unidad cúbica desperdiciada entre dos objetos. El número total de unidades cúbicas vacías se representa por la letra  $e$ .

El AGE cuenta con una rutina especial que se encarga de contar las unidades cúbicas vacías localizadas únicamente entre objetos, sin tomar en cuenta aquellas unidades vacías que quedan sobre los objetos una vez que todos ellos han sido empacados. El valor de  $\epsilon$  es un número real el cual es ajustado experimentalmente de

acuerdo a las dimensiones del problema y del comportamiento que presente el AGE. Por lo general, los problemas que involucren números grandes de unidades cúbicas de mercancía requieren que  $\epsilon$  tome valores mayores.

En la Figura 5.12 muestra el comportamiento de la curva de aptitud del AGE y su desviación estándar, cuando el algoritmo fue corrido 10 veces, usando diferentes semillas de números aleatorios y con la función objetivo de espacios vacíos.

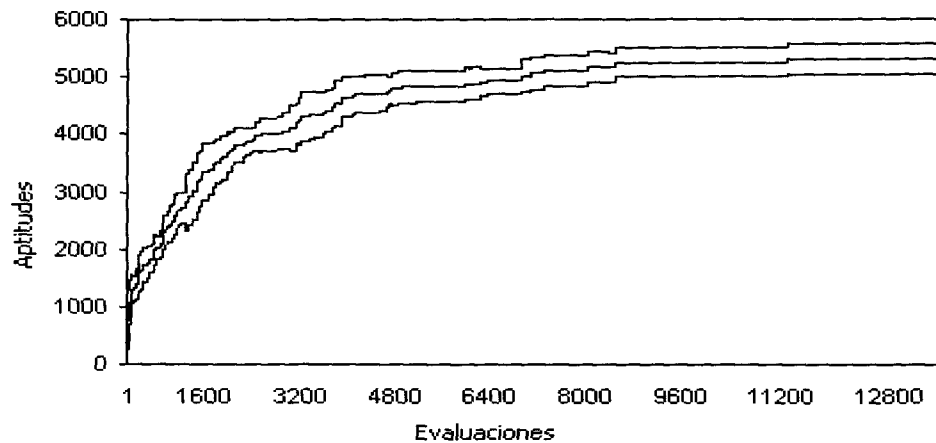


Figura 5.12: Curva de aptitud usando la función objetivo de espacios vacíos.

Los parámetros utilizados son:

- Tamaño de la población = 200
- Número de generaciones = 70
- Operador de cruce = *ERX*
- Probabilidad de cruce = 0.98
- Probabilidad de mutación = 0.02
- Tamaño del torneo = 4

Durante los experimentos realizados utilizando esta nueva función objetivo, los mejores resultados se obtuvieron haciendo  $\beta = 30$ ,  $\gamma = 15$  y  $\epsilon = 5$ . La evaluación final más baja obtenida fue 4,773 y la mejor fue 5,631.

Función objetivo de espacios vacíos				
Operador de cruce	ERX			
Probabilidad de cruce	0.98			
Probabilidad de mutación	0.02			
Puntajes	Unidades cúbicas empacadas	Puntos de adyacencia	Espacios vacíos	
Semillas	0.02	560	36	124
	0.16	560	40	120
	0.23	560	40	148
	0.39	560	31	159
	0.45	560	36	154
	0.52	536	36	169
	0.68	547	29	163
	0.74	560	37	149
	0.88	560	31	152
	0.93	550	30	125
Promedios	555.3	34.6	146.3	

Tabla 5.6: Puntajes de empaqueo obtenidos utilizando la función objetivo de espacios vacíos.

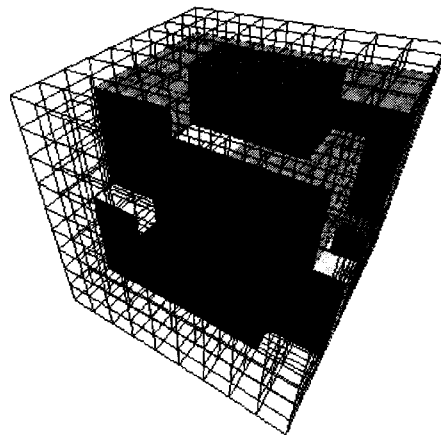


Figura 5.13: Una solución buena utilizando la función objetivo de espacios vacíos.

Las estadísticas en la Tabla 5.6 muestran que el número de espacios vacíos en las soluciones obtenidas usando la función de espacios vacíos, es menor que los obtenidos usando las primeras dos funciones objetivo. También, el número de unidades cúbicas empacadas aumenta con respecto a las soluciones de la función de adyacencia. Sin embargo, el puntaje con respecto a los puntos de adyacencia se ve perjudicado y disminuye un poco.

En la Figura 5.13 se muestra una de las mejores soluciones obtenidas con el AGE usando la función objetivo de espacios vacíos. Los objetos del mismo tipo aparecen del mismo color.

### 5.2.4 Comparación de las Funciones Objetivo

En la Tabla 5.7 se muestran los promedios de los puntajes obtenidos por el AGE con cada una de las funciones objetivo implementadas y al aplicarse para resolver el mismo problema de empaçado.

Puntajes	Funciones objetivo		
	Volumen	Adyacencias	Espacios vacíos
Unidades cúbicas empacadas	556	550.8	555.3
Puntos adyacentes	31	36.9	34.6
Espacios vacíos	154.2	162.6	143.3

Tabla 5.7: Puntajes obtenidos con diferentes funciones objetivo.

Estos resultados pueden interpretarse como se muestra en la Tabla 5.8. Es claro que la mejor función objetivo es la última implementada, la función objetivo de espacios vacíos, la cual no es tan buena como la función de adyacencia en lo relativo a empaçar objetos del mismo tipo en lugares cercanos, pero al sumar todos los factores de calificación resulta ser la mejor función objetivo.

Por lo tanto, la función objetivo de espacios vacíos es la mejor función objetivo para correr el AGE considerando los puntajes que obtuvo al calcular el volumen, los puntos de adyacencia y los espacios vacíos generados en sus soluciones.

Características	Funciones objetivo		
	Volumen	Adyacencia	Espacios vacíos
Para maximizar el volumen empacado	bueno	malo	bueno
Para empacar objetos ordenadamente	malo	bueno	bueno
Para empacar objetos de forma compacta	bueno	malo	bueno

Tabla 5.8: Características de las diferentes funciones objetivo.

### 5.3 Resolviendo Diversos Problemas de Empacado

En esta sección, se muestran los resultados obtenidos realizando experimentos con diversos problemas. Estos problemas se realizaron en una PC con procesador Celeron a 360 Mhz y 32 MB de memoria RAM, usando el siguiente conjunto de parámetros:

- Tamaño de la población = 200
- Número de iteraciones = 70
- Función objetivo = de espacios vacíos
- Operador de cruce = *ERX*
- Probabilidad de cruce = 0.98
- Probabilidad de mutación = 0.02

Los tipos de objeto *C*, *G* e *I* que fueron usados para experimentar en esta sección, se ilustran en la Figura 5.14.

Mientras más complicado es el problema de empacado, el AGE necesita más tiempo para resolverlo. En esta parte de la investigación se estudió el incremento en el tiempo que el algoritmo necesita para proporcionar una solución al complicar el problema de las siguientes dos formas:

1. Incrementando el número de objetos a empacar
2. Incrementando el tamaño de los objetos a empacar



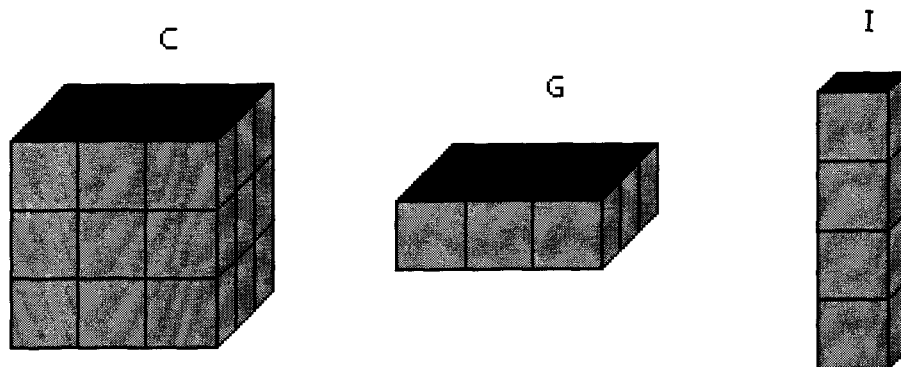


Figura 5.14: Tipos de objetos utilizados en experimentos con el tiempo requerido por el AGE para proporcionar una solución.

### 5.3.1 Incrementando el Número de Objetos

En esta parte de la investigación se estudió el tiempo que requiere el AGE para resolver problemas que se complican cada vez más al aumentar el número de objetos a empacar. El primer experimento se realizó con 3 objetos de cada tipo (*C*, *G* e *I*). Luego, cada vez que el algoritmo se corría, se duplicaba el número de objetos, de tal forma que el último experimento realizado involucró el empacado de 192 objetos de cada tipo. Al mismo tiempo que se aumentaba el número de objetos, el contenedor fue agrandado 2 unidades en largo, alto y ancho. Por lo tanto, el problema se complicó de trabajar con 9 objetos en el primer experimento hasta 288 objetos en el último. El tiempo necesitado para proporcionar la solución varió de 4.2 segundos en el primer experimento, hasta 6.25 minutos en el último.

Dimensiones del contenedor			Número de objetos			Total de objetos	Unidades cúbicas por empacar	Unidades cúbicas empacadas	Puntos de adyacencia	Espacios vacíos	Tiempo (minutos)
Ancho	Largo	Alto	Tipo C	Tipo G	Tipo I						
6	6	4	3	3	3	9	117	117	6 of 6	21	00:04.2
8	8	6	6	6	6	18	234	234	14 of 15	52	00:08.7
10	10	8	12	12	12	36	468	468	33 of 33	181	00:20.3
12	12	10	24	24	24	72	936	936	63 of 69	178	01:02.9
14	14	12	48	48	48	144	1872	1842	134 of 141	383	02:08.6
16	16	14	96	96	96	288	3744	3168	283 of 285	191	06:16.8

Tabla 5.9: Puntajes de empacado e incrementos en el tiempo de ejecución para problemas que involucran diferentes números de objetos.

Las estadísticas de los puntos de adyacencia resultaron ser bastante buenas, considerando que únicamente en los dos últimos experimentos quedaron unos cuantos objetos sin empacar, como se muestra en la Tabla 5.9. Es claro que el AGE requiere

Contenedor			Tipo C			Tipo G			Tipo I			Objetos totales	Unidades cúbica por empacar	Unidades cúbicas empacadas	Puntos adyacentes	Espacios vacíos	Tiempo (minutos)
A	L	H	A	L	H	A	L	H	A	L	H						
6	6	4	3	3	3	3	3	1	1	1	3	9	117	117	6 of 6	21	00:04.2
12	12	8	6	6	6	6	6	2	2	2	6	9	936	924	6 of 6	204	00:16.9
24	24	16	12	12	12	12	12	4	4	4	12	9	7488	7392	5 of 6	1716	02:58.7

A = Ancho, L = Largo, H = Alto

Tabla 5.10: Puntajes de empaqueo e incrementos en el tiempo de ejecución para problemas que involucran diferentes tamaños de objetos.

de más tiempo para resolver problemas que involucran un mayor número de objetos. La razón de esto es simple, por ejemplo, cuando el problema de empaqueo involucra 2 objetos, la longitud del cromosoma es de 2, en cambio cuando involucra 288 objetos, el cromosoma crece a una longitud de 288. Por lo tanto, el algoritmo necesita más tiempo para aplicar los operadores de selección, cruce y mutación cuando los cromosomas son más largos.

### 5.3.2 Incrementando el Tamaño de los Objetos

El tiempo requerido por el AGE para resolver problemas fue estudiado de nuevo, ahora complicando más los problemas al incrementar el tamaño de los objetos. El primer experimento se realizó con 3 objetos de cada tipo (*C*, *G* e *I*) nuevamente. Después, cada vez que el algoritmo fue ejecutado las dimensiones de los objetos se duplicaron, de tal forma que el último experimento realizado involucró objetos de tamaño 4 veces más grandes que los objetos originales. Al mismo tiempo que los objetos fueron agrandados, el contenedor fue duplicado en su alto, largo y ancho. De esta forma se tiene que del primer experimento al último hubo un incremento en el total de unidades cúbicas empacadas de 117 a 7,488, con tiempos de ejecución que van de 4.2 segundos a 2.95 minutos.

Los resultados en la Tabla 5.10 indican una fuerte dependencia entre los tamaños de los objetos y el tiempo requerido para hallar una solución del problema, siendo este mayor conforme mayor es el tamaño de los objetos. Cada vez que un individuo es evaluado, el AGE utiliza un arreglo de caracteres cuya longitud es igual al número total de unidades cúbicas del contenedor. Cuando una unidad cúbica es ocupada dentro del contenedor, su elemento correspondiente en el arreglo toma el valor de 1. Por el contrario, si permanece desocupado, su posición en el arreglo permanece con valor 0. El arreglo debe ser llenado de unos y ceros cada vez que un individuo es evaluado. Entonces es obvio que este proceso le toma más tiempo al AGE cuando el problema

involucra un contenedor con mayor capacidad de unidades cúbicas.

Como conclusión de estas dos últimas secciones, se puede decir que el AGE requiere de más tiempo para proporcionar una solución cuando el número de objetos a empacar es mayor, o bien, cuando son de mayor tamaño y es necesario un contenedor más grande.

## 5.4 Problemas de Empacado con $n$ Contenedores

Al adaptar el AGE para resolver problemas que involucran más de un contenedor se hace necesario desarrollar heurísticas que guíen el proceso de empacar los  $m$  objetos dentro de  $n$  contenedores.

En esta sección se explican y analizan dos métodos diferentes para resolver problemas que involucran más de un contenedor, usando el siguiente problema definido con fines experimentales:

Se trata de un problema con 2 contenedores con las siguientes dimensiones: Contenedor 1

- ancho = 9
- largo = 9
- alto = 9

Contenedor 2

- ancho = 8
- largo = 9
- alto = 6

Son 67 objetos por empacar, que corresponden a 869 unidades cúbicas y están distribuidos como se indica en la Tabla 5.11.

Los tipos de objeto  $A$ ,  $C$  y  $D$  se ilustran en la Figura 5.15

Tipo	Ancho	Largo	Alto	Número
A	1	1	1	20
C	3	3	3	27
D	1	2	3	20

Tabla 5.11: Catálogo de objetos utilizados para experimentos con problemas de  $n$  contenedores.

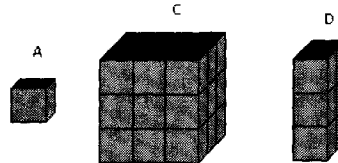


Figura 5.15: Los tipos de objetos  $A$ ,  $C$  y  $D$  fueron utilizados en experimentos con  $n$  contenedores.

### 5.4.1 Trabajando sin Heurística

Trabajando sin heurística es la forma más fácil de resolver el problema con más de un contenedor, y consiste en aplicar únicamente la función objetivo de espacios vacíos explicada en la sección anterior. La evaluación de los individuos considera el total de las unidades cúbicas empacadas, el total de los puntos de adyacencia entre objetos del mismo tipo y el número de unidades cúbicas vacías entre los objetos empacados en los  $n$  contenedores.

La curva de evaluaciones del AGE con los siguientes parámetros, se muestra en la Figura 5.16.

- Tamaño de la población = 200
- Número de iteraciones = 70
- Operador de cruce =  $ERX$
- Probabilidad de cruce = 0.98
- Probabilidad de mutación = 0.02

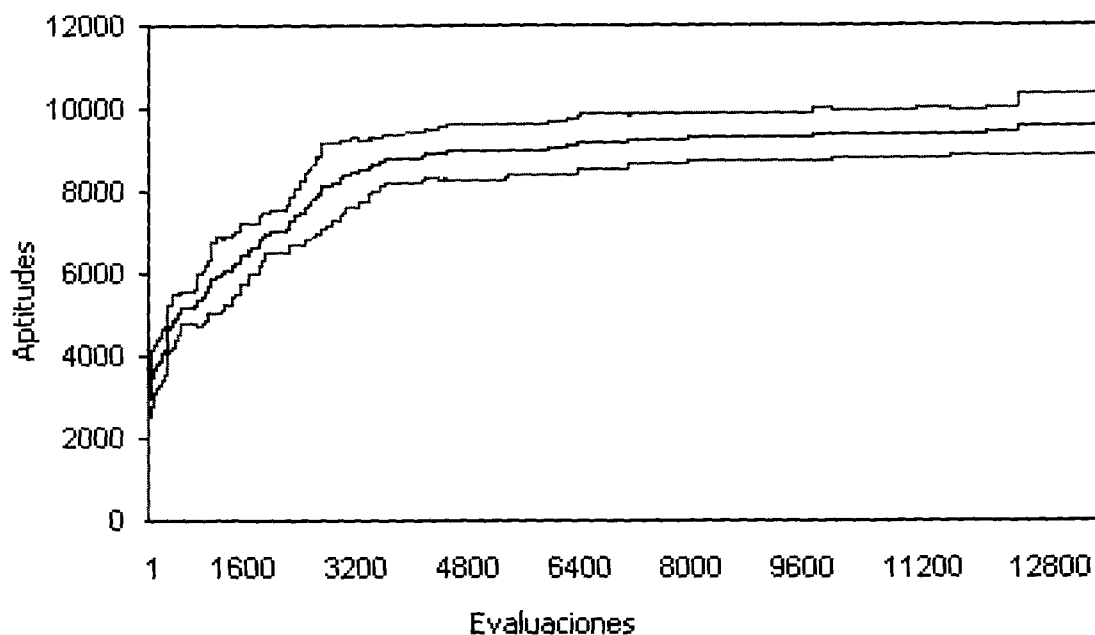


Figura 5.16: Curva de aptitud al resolver un problema con 2 contenedores y sin heurística.

Las soluciones generadas por el AGE son bastante buenas y se muestran en las Figuras 5.17 y 5.18.

Trabajar sin heurística alguna asegura que todos los objetos serán escogidos uniformemente al momento de asignarles uno de los  $n$  contenedores para ser empacados. Esta forma de empaqueo puede ser muy útil cuando no importa en que contenedor quede determinado tipo de objetos, sino que lo único importante es que todos los objetos sean empacados. Las soluciones en la Figuras 5.17 y 5.18 muestran como fueron empacados los objetos en ambos contenedores.

#### 5.4.2 Trabajando con la Heurística de Prioridad

La heurística de prioridad consiste en empaquear  $m$  objetos dentro de  $n$  contenedores siguiendo el orden indicado por la prioridad asignada a cada objeto. Esto significa que el AGE tratará de empaquear primero, en el primer contenedor, los objetos que tienen prioridad más alta y luego, en los siguientes contenedores, empaqueará los objetos que tienen prioridad menor, hasta empaquear todos los objetos. La prioridad es indicada por un valor que puede ser un número real o un número entero. Los valores de prioridad son asignados al momento de crear el catálogo de objetos.

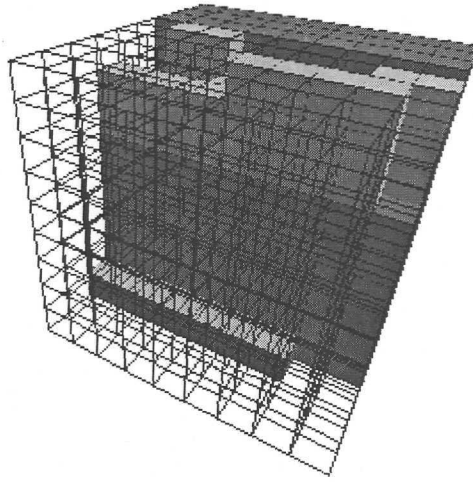


Figura 5.17: Una solución buena sin heurística: Contenedor 1.

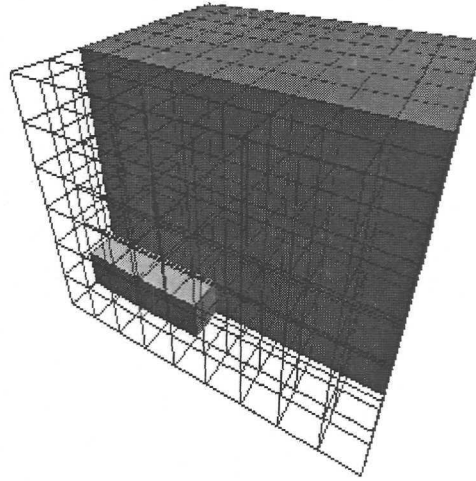


Figura 5.18: Una solución buena sin heurística: Contenedor 2.

La curva de evaluaciones del AGE con el siguiente conjunto de parámetros se muestra en la Figura 5.19.

- Tamaño de la población = 200
- Número de iteraciones = 70
- Operador de cruce = *ERX*
- Probabilidad de cruce = 0.98
- Probabilidad de mutación = 0.02

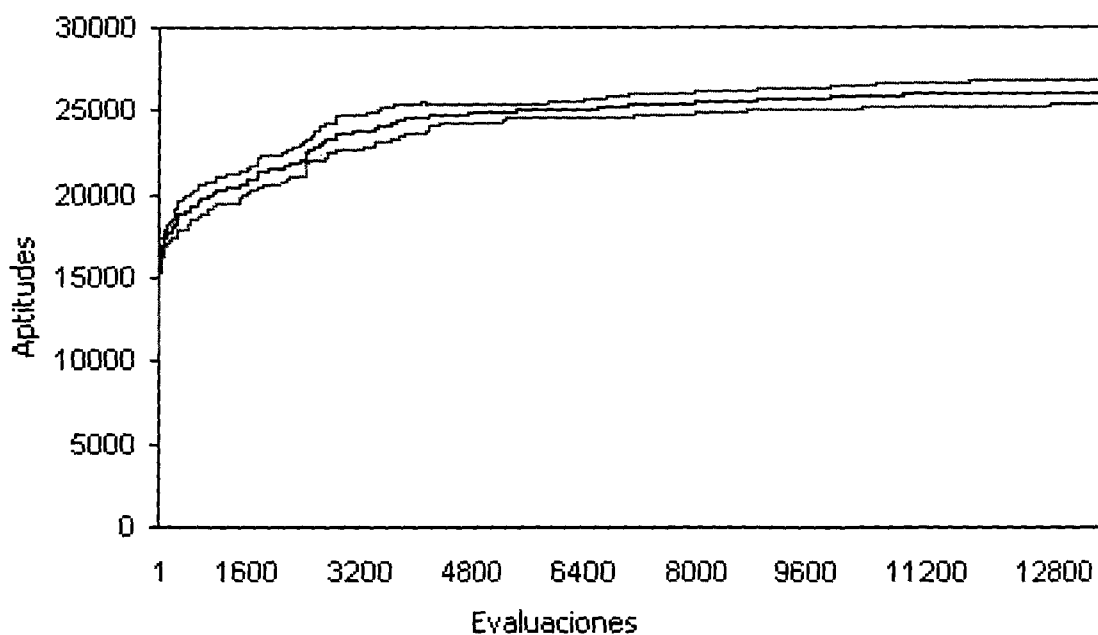


Figura 5.19: Curva de aptitud al resolver un problema con 2 contenedores usando la heurística de prioridad.

Al trabajar con la heurística de prioridad se asegura que los objetos con mayor peso sean empacados primero y luego los objetos de menor peso. De esta forma, esta heurística puede ser muy útil cuando entre la mercancía existan objetos que por alguna circunstancia tengan que ser empacados primero.

En las Figuras 5.20 y 5.21 se muestra como los objetos de color rojo fueron empacados en el primer contenedor. Estos objetos son del tipo  $C$  y son los objetos de mayor tamaño en el catálogo. Para este experimento se utilizó el volumen de los objetos como su valor de prioridad.

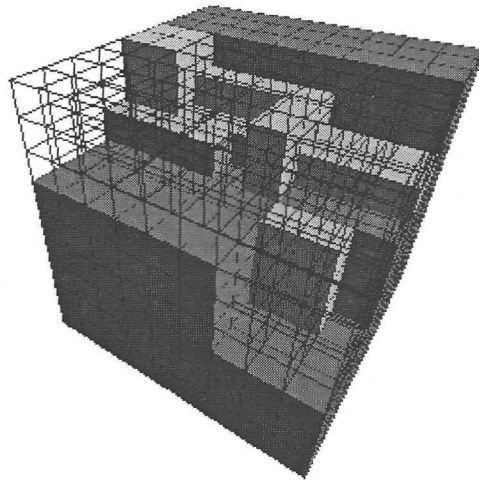


Figura 5.20: Una solución buena con la heurística de prioridad: Contenedor 1.

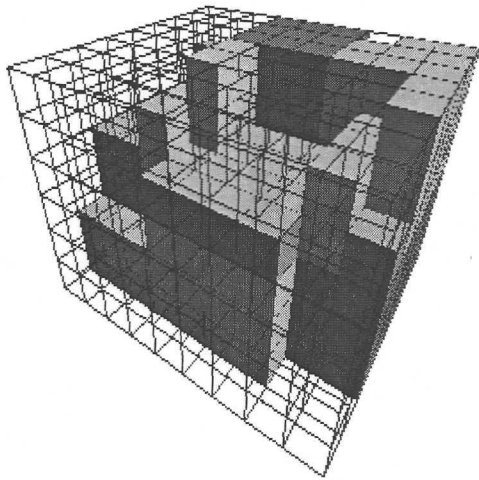


Figura 5.21: Una solución buena con la heurística de prioridad: Contenedor 2.

### 5.4.3 Puntaje de Empacado para Cargar 2 Contenedores

Las estadísticas en la Tabla 5.12 fueron generadas ejecutando el AGE 10 veces usando diferentes semillas de números aleatorios. Los puntajes de empaado indican que usando la función objetivo de espacios vacíos se puede lograr que todos los objetos sean empaados. Además, el promedio de los puntos adyacentes es muy cercano al valor máximo posible. El promedio de números de espacios vacíos es pequeño, indicando que los objetos fueron empaados de una forma compacta, con pocos espacios vacíos entre objetos.

En la Tabla 5.13 se muestran los puntajes de empaado obtenidos con el AGE al usar la heurística de prioridad con dos contenedores. Las estadísticas indican que



Semillas	Totales		
	Unidades cúbicas empacadas	Puntos adyacentes	Espacios vacíos
0.02	869	64 of 64	16
0.16	869	63 of 64	17
0.23	869	63 of 64	16
0.39	869	61 of 64	12
0.45	869	61 of 64	15
0.52	869	63 of 64	5
0.68	869	63 of 64	17
0.74	869	62 of 64	3
0.88	869	62 of 64	63
0.93	869	63 of 64	32
AVG	869	62.5	19.6

Tabla 5.12: Puntajes obtenidos por el AGE con un problema de 2 contenedores y sin heurística.

todos los objetos fueron empacados y que el número de puntos de adyacencia es alto. Sin embargo, a diferencia de los puntajes obtenido sin heurística, el número de espacios vacíos es elevado, debido a que los objetos grandes fueron empacados en el primer contenedor y el resto de los objetos en el segundo contenedor. Cuando es necesario empacar objetos de diferente forma es muy difícil hacerlo sin dejar espacios vacíos entre ellos, que es el caso del segundo contenedor.

Semillas	Totales		
	Unidades cúbicas empacadas	Puntos adyacentes	Espacios vacíos
0.02	869	62 of 64	73
0.16	869	62 of 64	200
0.23	869	64 of 64	59
0.39	869	64 of 64	35
0.45	869	52 of 64	133
0.52	869	59 of 64	36
0.68	869	60 of 64	139
0.74	869	60 of 64	205
0.88	830	61 of 64	104
0.93	869	61 of 64	51
AVG	865.1	60.5	103.5

Tabla 5.13: Puntajes obtenidos por el AGE con un problema de 2 contenedores y con la heurística de prioridad.

## 5.5 Resumen

En la primera parte experimental se obtuvo que el operador de cruce *ERX* proporciona mucho mejores soluciones que el operador *PMX*. También se obtuvo que el mejor conjunto de parámetros para correr el AGE es el siguiente:

- Tamaño de la población = 200

- Número de generaciones = 70
- Operador de cruce = *ERX*
- Probabilidad de cruce = 0.98
- Probabilidad de mutación = 0.02
- Tamaño del torneo = 4

Después de definir el mejor operador de cruce y el mejor conjunto de parámetros, tres funciones objetivo fueron implementadas y probadas, las cuales fueron llamadas “función objetivo de volumen”, “función objetivo de adyacencia” y “función objetivo de espacios vacíos”. De estas tres funciones, la mejor resultó ser la función objetivo de espacios vacíos, la cual considera el número total de unidades cúbicas empacadas, los puntos de adyacencia entre objetos del mismo tipo y además reduce el número de espacios vacíos entre los objetos, logrando soluciones donde toda la mercancía es empacada, de manera ordenada y compacta.

Posteriormente, al estudiar el incremento en los tiempos que el AGE necesita para proporcionar una solución conforme los problemas se vuelven más complicados, se demostró que el aumentar el número de objetos o bien, las dimensiones de los objetos, también se incrementa considerablemente el tiempo de ejecución del AGE.

Finalmente, se propusieron y analizaron dos formas diferentes de resolver el problema de empacado cuando este involucra más de un contenedor.

## Capítulo 6

# Resolviendo Problemas Reales de Empacado

En la industria ha existido siempre la necesidad de transportar bienes de un lugar a otro, y para este propósito las compañías usan normalmente contenedores en camiones, trenes, barcos o aviones. La meta principal es empaacar el mayor número posible de artículos para reducir costos, y por lo tanto, incrementar las ganancias. Es así que el interés se enfoca en la optimización de lo que se llama carga de contenedores.

En el capítulo anterior se describió el diseño y la implementación del AGE. También se mostraron y comentaron los resultados obtenidos al experimentar con problemas diversos, escogidos cuidadosamente para probar y mejorar el desempeño del algoritmo. Sin embargo, nunca se sabría si el AGE realiza un buen trabajo o no, si nunca se utiliza para resolver problemas reales de empaacado. Es por esta razón que se tomaron dos problemas de empaacado de una compañía real de transportación y se resolvieron usando el AGE. Por lo tanto, en este capítulo se encontrarán algunos detalles relacionados a la resolución de esos problemas, sus descripciones, y finalmente, los resultados experimentales obtenidos en esta parte de la investigación.

## 6.1 Experimentando con Problemas Reales

Como anteriormente se dijo, se resolvieron dos problemas de carga de contenedores de una compañía real de transportación usando el AGE. Esta compañía transporta diferentes productos de una ciudad a otra, usando contenedores de diferentes dimensiones, dependiendo de las ordenes de sus clientes. Algunas veces, varias ordenes diferentes deben ser transportadas usando el mismo contenedor. En este caso, los objetos que pertenecen al mismo tipo deben ser empaacados juntos, para permitir posteriormente un proceso de empaacado eficiente. Por lo tanto, los objetos de diferentes ordenes deben ser empaacados en lugares separados. Para esta etapa, los experimentos fueron realizados en una PC con un procesador Pentium 4 a 1.2 GHz y con 256 MB de memoria.

El primer problema de empaacado consiste en una orden de 280 artículos que deben

ser empacados en el mismo contenedor, y el segundo problema consiste en empacar los artículos de dos ordenes de diferentes clientes en un contenedor. Una orden tiene 581 artículos y la otra tiene 234. Los productos transportados en ambos problemas son cajas con forma de paralelepípedos que contienen radiadores de automóviles. Esas cajas están clasificadas de acuerdo a sus dimensiones en 34 tipos diferentes. La clasificación de las cajas se muestra en la Figura 6.1. Ambos problemas en este capítulo fueron resueltos bajo las mismas condiciones utilizadas en los experimentos anteriores, usando los mismos operadores de cruce y mutación, y tomando en cuenta lo siguiente:

Las restricciones básicas consideradas son:

- Dos o más artículos no pueden ocupar el mismo lugar al mismo tiempo.
- Un artículo no puede quedar suspendido en el aire, sino que debe ser colocado sobre una superficie plana que puede ser el fondo del contenedor o la parte superior de otro artículo.
- Los artículos del mismo tipo deben ser empacados en lugares cercanos unos a otros.
- Los artículos y contenedores tienen forma de paralelepípedos.

Y las suposiciones básicas son:

- Los artículos no tienen peso.
- Los contenedores son similares a cajas grandes.
- Los artículos son colocados en el interior del contenedor desde arriba.

En la Figura 6.2 se muestran las posiciones posibles en las cuales puede ser colocado un artículo dentro de los contenedores.

Para resolver problemas reales de manera más eficiente en términos de tiempo y calidad, se realizó el siguiente preprocesamiento:

1. Escalamiento: El AGE depende fuertemente del número total de unidades cúbicas que se necesitan empacar, para proporcionar una solución, como se mostró en la Tabla 5.9 (Capítulo 5), cuando se experimentó con diversos problemas incrementando el tamaño de los objetos. Por esta razón, se decidió reducir la complejidad del problema, escalando las dimensiones tanto del contenedor como de los objetos a empacar, reduciéndolos 10 veces. El proceso de escalamiento no afecta la parte esencial de los problemas, simplemente le simplifica el proceso de solución al AGE.

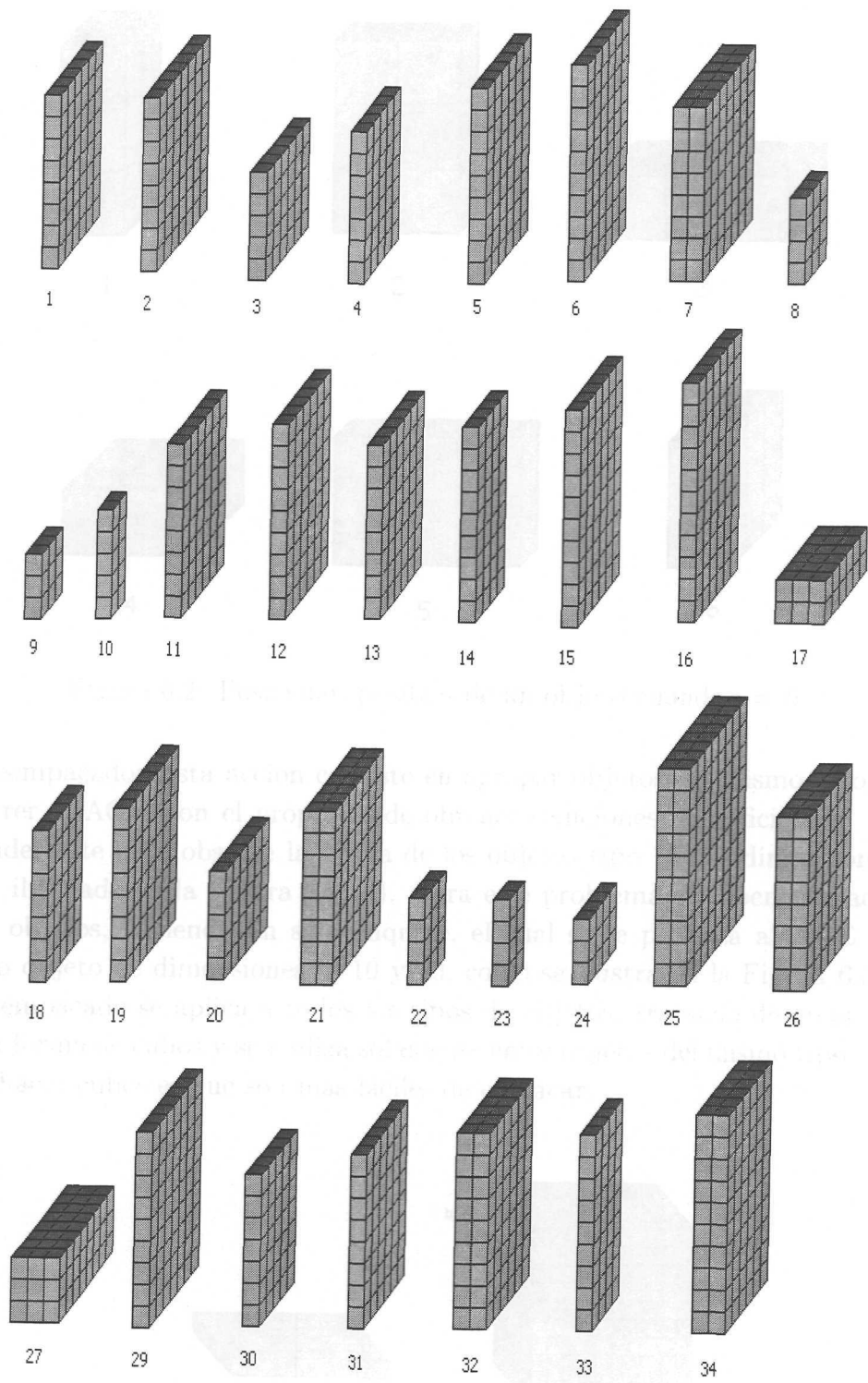


Figura 6.1: Clasificación de las cajas dependiendo de sus dimensiones.

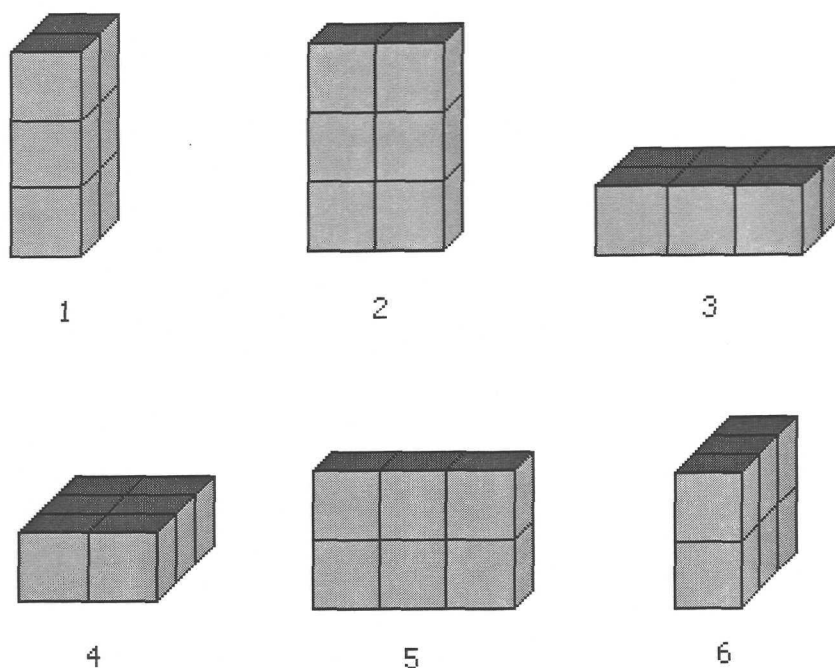


Figura 6.2: Posiciones posibles de un objeto cuando  $p = 6$ .

2. Preempacado: Esta acción consiste en agrupar objetos del mismo tipo antes de correr el AGE, con el propósito de obtener soluciones más eficientes. Para entender este paso observe la forma de los objetos tipo 15 con dimensiones 6, 1 y 10, ilustrado en la Figura 6.3 (a). Para este problema se deben empaclar 10 de los objetos, haciendo un sólo paquete, el cual se le presenta al AGE como un solo objeto de dimensiones 6, 10 y 10, como se ilustra en la Figura 6.3 (b). El preempacado se aplica a todos los tipos de objetos, tratando de crear paquetes con forma de cubos y se realiza solamente entre objetos del mismo tipo. La razón de hacer cubos es que son más fáciles de empaclar.

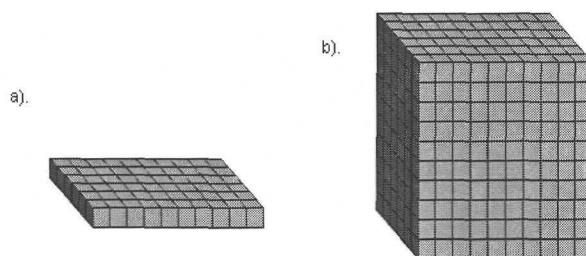


Figura 6.3: a) Forma del objeto tipo 15. b) Paquete de objetos tipo 15.

El escalamiento y el preempacado se aplicó a ambos problemas. Los puntajes

utilizados para analizar la calidad de las soluciones proporcionadas por el AGE son los mismos utilizados en el capítulo anterior: volumen, puntos adyacentes y espacios vacíos.

## 6.2 Primer Problema Real de Empacado

El primer problema como dijimos anteriormente, consiste en empaquetar 280 objetos pertenecientes a 14 diferentes tipos, de cantidades y dimensiones como se muestra en la Tabla 6.1. La mercancía equivale a  $13,049 \text{ dm}^3$  (decímetros cúbicos). Las formas de los objetos pueden verse en la Tabla 6.1. El contenedor en el cual deben ser empaquetados los objetos tiene las siguientes dimensiones:

- ancho =  $2.1 \text{ m}$
- largo =  $4.5 \text{ m}$
- alto =  $2 \text{ m}$

Tipo	Largo (cm)	Ancho (cm)	Alto (cm)	Cantidad
32	57.5	16	85	19
29	71	14	85	15
15	64	13	101	10
31	56	14	79	79
30	53.5	14	68.5	30
8	24.6	13.6	38.7	16
33	38.5	14	87	2
5	57.5	13	85	5
34	64	16	101	7
16	59	13	113	1
2	71	10	79	33
1	56	10	79	6
4	53.5	10	68.5	20
3	56	10	44.5	37

Tabla 6.1: Dimensiones de los objetos utilizados para resolver el primer problema real de empaquetado.

Para resolver el problema de empaquetado, el AGE se corrió 5 veces con diferentes semillas generadoras de números aleatorios y utilizando la función objetivo de espacios vacíos, explicada anteriormente. El conjunto de parámetros utilizado es el siguiente:

- Tamaño de la población = 200

- Número de generaciones = 100
- Operador de cruce = *ERX*
- Probabilidad de cruce = 0.98
- Probabilidad de mutación = 0.2
- Tamaño del torneo = 4

En la Figura 6.4 se muestra la curva de aptitud generada por el AGE junto con su desviación estándar. La curva de aptitud es el promedio de las 5 corridas del algoritmo. En la Figura 6.5 se muestra una de las mejores soluciones obtenidas con el AGE, en donde logra empaclar  $13,049 \text{ dm}^3$ , que corresponde al total de los objetos.

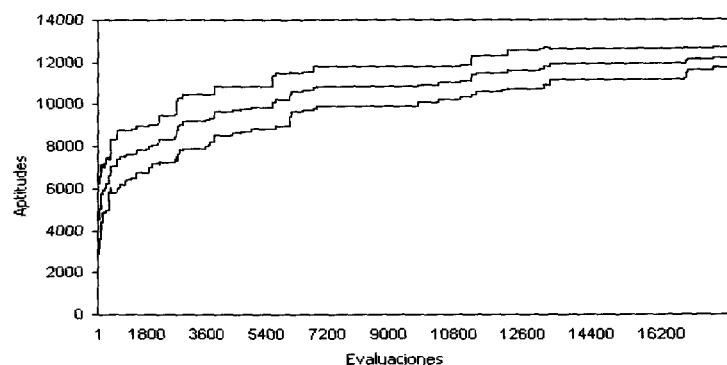


Figura 6.4: Comportamiento de la curva de aptitud del AGE al resolver el primer problema real de empaclado.

Las estadísticas en la Tabla 6.2 indican los puntajes obtenidos por el AGE cuando se corrió para resolver el primer problema.

Semillas	Unidades cúbicas por empaclar	Unidades cúbicas empacladas	Puntos de adyacencia	Espacios vacíos	Máxima evaluación
0.15	13049	12953	18 of 27	5443	33380
0.37	13049	13049	12 of 27	5771	33304
0.41	13049	12665	12 of 27	5695	32228
0.78	13049	13049	8 of 27	5319	33732
0.99	13049	13049	21 of 27	5787	33342
AVG	13049	12953	14.2	5603	33197.2

Tabla 6.2: Puntajes de empaclado del primer problema.



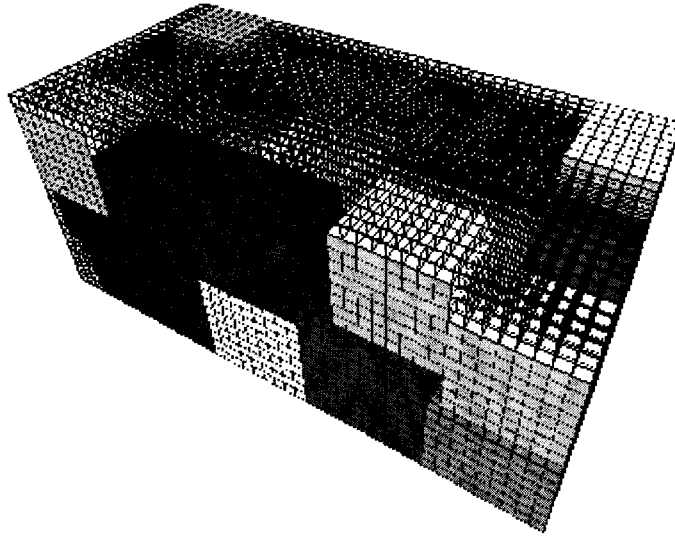


Figura 6.5: Solución del primer problema real.

Semillas	Unidades cúbicas por empacar	Unidades cúbicas empacadas	Puntos de adyacencia	Espacios vacíos	Máxima evaluación
0.15	13049	13049	18 of 27	5204	33910
0.37	13049	13049	10 of 27	5082	33984
0.41	13049	13049	11 of 27	5460	33612
0.78	13049	12883	18 of 27	4810	33806
0.99	13049	13049	15 of 27	5230	33866
AVG	13049	13015.8	14.4	5157.2	33835.6

Tabla 6.3: Puntajes de empacado del primer problema con 2 secuencias de carga.

Los puntajes de empacado que aparecen en la Tabla 6.2 son los mejores resultados obtenidos para el primer problema, cuando se le permitió al AGE colocar los objetos en cualquier lugar del contenedor, incluso hasta suspendidos en el aire. Sin embargo, dado que en la realidad, los objetos no pueden permanecer suspendidos en el aire, se pensó en una mejor solución en la cual sea obligatoria la restricción de que los objetos siempre tengan que ser colocados sobre una superficie. Para garantizar esta mejor solución, el empacado de los objetos se organizó en dos secuencias, en donde cada secuencia consistió en empacar la mitad de los objetos en una mitad del contenedor, como si se estuvieran resolviendo dos problemas por separado.

Los nuevos resultados obtenidos son mucho mejores como puede verse en la Tabla 6.3. El AGE no tuvo ningún problema en colocar cada uno de los objetos dentro del contenedor. Analizando los datos de la tabla se puede ver que durante las cinco corridas del algoritmo, el promedio de los puntos adyacentes obtenidos es 14.4 de 27 puntos posibles, lo cual significa que el 53.33% de los objetos del mismo tipo fueron empacados

juntos. Este es un buen porcentaje de puntos de adyacencia considerando el número tan grande de objetos empacados. El número de espacios vacíos entre objetos nos indica el número de unidades cúbicas vacías que quedaron entre los objetos empacados, el cual para este problema, resultó ser de 5,157.2. En este caso, dado que toda la mercancía fue empacada, el número de espacios vacíos pasa a segundo plano, aunque siempre es recomendable lograr el menor número de ellos en una solución.

Comparando los números de la Tabla 6.2 y la Tabla 6.3, se puede ver que el número de espacios vacíos de la primera solución a la segunda, se redujo en 445.8 unidades. Además, el número de unidades cúbicas empacadas se incrementó en 62.8 unidades y finalmente, el número de puntos de adyacencia entre objetos del mismo tipo, resultó ser un poco mayor. Estos números nos sugieren que el método de cargar el contenedor mediante una secuencia compuesta de dos pasos de carga, proporciona mejores soluciones y por lo tanto hace que el proceso de carga sea más eficiente.

### 6.3 Segundo Problema Real de Empacado

El segundo problema consiste en empacar dos ordenes de productos pertenecientes a diferentes clientes dentro de un mismo contenedor con las siguientes dimensiones:

- largo = 10.4  $m$
- ancho = 2.36  $m$
- alto = 2.4  $m$

Una de las ordenes tiene 581 objetos y la otra 234, pertenecientes a 15 tipos diferentes y equivale a 33,615  $dm^3$  (decímetros cúbicos). Los radiadores están colocados en cajas de distintas dimensiones, dependiendo de las especificaciones de cada radiador. Todas las cajas como se dijo anteriormente, tienen forma de paralelepípedos.

Dado que se requiere empacar las mercancía de ambos clientes de forma separada, se ha dividido el problema en dos subproblemas.

El primer subproblema consiste en empacar 581 cajas en un espacio con las siguientes dimensiones:

- largo = 7.6  $m$
- ancho = 2.36  $m$
- alto = 2.4  $m$

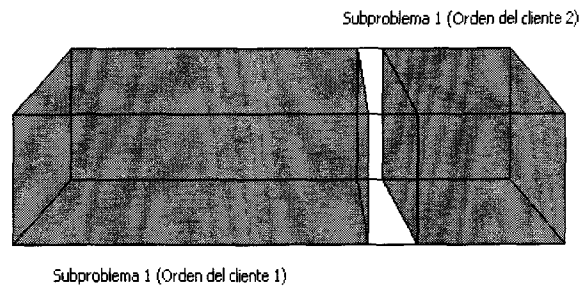


Figura 6.6: División del contenedor para empacar las diferentes ordenes de mercancía.

En la Tabla 6.4 se muestran los tipos, dimensiones y cantidades de objetos a empacar en el primer subproblema. Las formas de estos objetos pueden verse en la Figura 6.1.

Tipo	Largo (cm)	Ancho (cm)	Alto (cm)	Cantidad
29	71	14	85	50
15	64	13	101	11
5	57.5	13	85	7
8	24.6	13.6	38.7	61
34	64	16	101	40
32	57.5	16	85	85
31	56	14	79	259
30	53.5	14	68.5	61
33	38.5	14	87	6
25	80	20	105	1

Tabla 6.4: Dimensiones de los objetos del primer subproblema.

El segundo subproblema consiste en empacar 234 cajas que equivalen a  $9,679 \text{ dm}^3$  dentro de un espacio con las siguientes dimensiones:

- largo =  $2.6 \text{ m}$
- ancho =  $2.36 \text{ m}$
- alto =  $2.4 \text{ m}$

En la Tabla 6.5 se muestran los tipos, dimensiones y cantidades de los objetos a empacar en el segundo subproblema. Las formas de estos objetos pueden verse en la Figura 6.1.

Para cada subproblema se corrió el algoritmo 5 veces con diferentes semillas generadoras de números aleatorios, usando el mismo conjunto de parámetros para el AGE

Tipo	Largo (cm)	Ancho (cm)	Alto (cm)	Cantidad
5	57.5	13	85	33
1	56	10	79	41
2	71	10	79	48
10	21	8	49	12
4	53.5	10	68.5	67
3	56	10	44.5	32
8	24.6	13.6	38.7	1

Tabla 6.5: Dimensiones de los objetos del segundo subproblema.

que se usaron en el primer problema. Los resultados obtenidos en ambos subproblemas se muestran en la Tabla 6.6 y la Tabla 6.7 respectivamente.

Semillas	Unidades cúbicas por empacar	Unidades cúbicas empacadas	Puntos de adyacencia	Espacios vacíos	Máxima evaluación
0.15	33615	29927	62 of 83	13769	76087
0.37	33615	30471	71 of 83	13285	78257
0.41	33615	30603	66 of 83	13097	78811
0.78	33615	30287	60 of 83	13402	77522
0.99	33615	30844	56 of 83	12828	79743
AVG	33615	30426.4	63	13276.2	78084

Tabla 6.6: Puntajes del primer subproblema.

Semillas	Unidades cúbicas por empacar	Unidades cúbicas empacadas	Puntos de adyacencia	Espacios vacíos	Máxima evaluación
0.15	9679	9679	16 of 27	3399	25623
0.37	9679	9679	20 of 27	3805	25241
0.41	9679	9679	23 of 27	3750	25314
0.78	9679	9679	22 of 27	3519	25539
0.99	9679	9679	23 of 27	3173	25891
AVG	9679	9679	20.8	3529.2	25521.6

Tabla 6.7: Puntajes del segundo subproblema.

Las estadísticas en la Tabla 6.8 y en la Tabla 6.9, indican los puntajes obtenidos por el AGE cuando se corrió 5 veces para resolver el primero y segundo subproblemas respectivamente. Esta vez, cada uno de los subproblemas se organizó en secuencias de carga, tal como se hizo con el primer problema real. El primer subproblema se dividió en 4 secuencias, mientras que el segundo subproblema fue dividido en 2 secuencias de carga. Como se explicó anteriormente, la organización por secuencias de carga permite obtener las soluciones más eficientes, con los mejores puntajes de empacado. Como se

muestra en las tablas, solamente en una corrida del algoritmo quedaron sin empacar unas cuantas unidades cúbicas de mercancía, que aunque bien, no es lo deseable, es un resultado bastante bueno considerando la cantidad de mercancía que se está empacando. De igual forma, se puede ver que los porcentajes de puntos de adyacencia de objetos del mismo tipo obtenidos son igual de buenos, siendo para el primer subproblema un 94.22 % y para el segundo subproblema 51.85%.

Semillas	Unidades cúbicas por empacar	Unidades cúbicas empacadas	Puntos de adyacencia	Espacios vacíos	Máxima evaluación
0.15	33615	33615	76 of 83	6794	94222
0.37	33615	33615	76 of 83	5294	95722
0.41	33615	33507	78 of 83	6768	93936
0.78	33615	33615	79 of 83	7597	93510
0.99	33615	33615	82 of 83	6028	95024
AVG	33615	33593.4	78.2	6496.2	94482.8

Tabla 6.8: Puntajes del primer subproblema con 4 secuencias de carga.

Semillas	Unidades cúbicas por empacar	Unidades cúbicas empacadas	Puntos de adyacencia	Espacios vacíos	Máxima evaluación
0.15	9679	9679	16 of 27	3392	25633
0.37	9679	9679	15 of 27	2958	26061
0.41	9679	9679	12 of 27	3681	25320
0.78	9679	9295	10 of 27	2995	24842
0.99	9679	9679	17 of 27	4088	24943
AVG	9679	9602.2	14	3422.8	25359.8

Tabla 6.9: Puntajes del segundo subproblema con 2 secuencias de carga.

Haciendo una comparación entre los datos de la Tabla 6.6 y Tabla 6.8, se puede ver que el promedio de los espacios vacíos entre objetos, se redujo en 6,780 unidades cúbicas, mientras que el promedio de unidades cúbicas empacadas se incrementó en 3,167 unidades cúbicas. Además, el puntaje de puntos de adyacencia entre objetos del mismo tipo aumentó en 15.2 puntos. Haciendo la comparación ahora entre la Tabla 6.7 y la Tabla 6.9 se puede ver que el promedio de los espacios vacíos se redujo en 106.4 unidades cúbicas. Sin embargo el número de unidades cúbicas se redujo 76.8 unidades cúbicas al igual que el promedio de puntos de adyacencia entre objetos del mismo tipo. Cabe recordar que estas soluciones obtenidas organizando el empacado en secuencias de carga, aseguran que todos los objetos serán colocados sobre superficies planas y que por lo tanto son soluciones totalmente factibles y reales, algo que no se puede decir de las primeras soluciones dadas, en las cuales no se utilizan las secuencias de carga.

Durante los experimentos realizados para resolver los dos problemas reales, los mejores resultados, donde la mercancía fue empacada en su totalidad, de manera ordenada y con un mínimo de unidades cúbicas desperdiciadas, se obtuvieron asignando los siguientes valores a las variables de la función objetivo de espacios vacíos:

- $\beta = 3$
- $\gamma = 3$
- $\epsilon = 3$

## 6.4 Resumen

En este capítulo se utilizó el AGE para resolver dos problemas reales de empacado, de una compañía que transporta mercancía de una ciudad a otra.

Durante la experimentación con estos casos reales, se hizo evidente que cuando el número de artículos a empacar es muy grande, se requieren de mejores recursos computacionales para poder ejecutar el AGE. Sobretudo se hizo necesario aumentar la velocidad de procesamiento, debido al incremento significativo en el número de operaciones que el algoritmo debía realizar. Es por esto que para resolver los problemas reales de empacado el AGE requirió de más tiempo.

Además, fue necesario preprocesar los datos antes de correr el algoritmo, haciendo escalamiento y preempacado de los objetos. El escalamiento consiste en redimensionar los objetos involucrados en el empacado, mientras que el preempacado consiste en agrupar objetos del mismo tipo. Estas dos simples operaciones simplifican el trabajo del AGE sin influir en la esencia de los problemas.

A pesar de las complicaciones, el AGE fue capaz de resolver ambos problemas de empacado de una manera eficiente, proporcionando buenas soluciones. Esta forma automatizada de hacer optimización de carga de contenedores asegura que a cada artículo le sea asignada una posición única en el contenedor, lo cual es verdaderamente importante para garantizar que toda la mercancía sea empacada. Este tipo de garantía no puede tenerse cuando se realiza carga de contenedores de manera empírica, como aún sigue realizándose en muchas compañías transportistas.

## Capítulo 7

### Conclusiones y Trabajo Futuro

El Algoritmo Genético de Empacado (AGE), es el resultado de una tesis orientada al diseño y desarrollo de una aplicación para realizar optimización del empaqueo de  $m$  objetos dentro de  $n$  contenedores, usando Algoritmos Genéticos que son una técnica de la Inteligencia Artificial.

En la vida real un problema típico de empaqueo puede involucrar la optimización de varias variables, lo que lo convierte en un problema muy difícil de resolver. Con el propósito de simplificar el caso real y abstraer de él un modelo que pueda servir con fines de investigación, se definió una instancia del problema, usando solamente objetos con forma de paralelepípedos, tanto para los contenedores como para los objetos que se requieren empaocar.

Durante el proceso de diseño del AGE, unas de las principales tareas realizadas, fue el análisis del problema de empaqueo y su reducción a un modelo simple. El modelo resultante influiría de manera importante en el diseño de la estructura del cromosoma. La decisión final fue la de utilizar cromosomas de números enteros, permitiendo trabajar los cromosomas como permutaciones que representarían cada posible solución como el orden de los objetos a empaocar.

Después de implementar el Algoritmo Genético se hicieron pruebas con los operadores de cruce *ERX* y *PMX*, de los cuales el *ERX* resultó ser el mejor para este problema de empaqueo. También, de toda la serie de experimentos, se llegó al mejor conjunto de parámetros de entrada del AG, el cual es el siguiente:

- Tamaño de la población = 200
- Número de generaciones = 70
- Operador de cruce = *ERX*
- Probabilidad de cruce = 0.98

- Probabilidad de mutación = 0.02
- Tamaño del torneo = 4

La mejor función objetivo resultó ser la de espacios vacíos, la cual toma en cuenta el número total de unidades cúbicas empacadas, los puntos de adyacencia entre objetos del mismo tipo y los espacios vacíos entre los objetos empacados. Esta función objetivo se define como sigue:

$$f(\Delta_k, s_k, r, p, q, e) = \sum_{k=1}^r \Delta_k s_k \omega + \beta p - \gamma q - \epsilon e \text{ donde,}$$

$$\Delta_k = l_k h_k w_k$$

- $\Delta_k$  : Volumen del objeto tipo  $k$
- $l_k$  : Largo del objeto tipo  $k$
- $h_k$  : Alto del objeto tipo  $k$
- $w_k$  : Ancho del objeto tipo  $k$
- $s_k$  : Número de objetos empacados tipo  $k$
- $p$  : Número de puntos de adyacencia entre objetos del mismo tipo
- $q$  : Número de puntos de adyacencia entre objetos de diferente tipo
- $e$  : Número de unidades cúbicas vacías entre objetos
- $\omega$  : Factor de peso
- $\beta$  : Constante de premio
- $\gamma$  : Constante de castigo
- $\epsilon$  : Constante de espacios vacíos
- $r$  : Número de tipos de objetos

La función objetivo de espacios vacíos ofrece mejores soluciones que las funciones de volumen y de adyacencia. Por otra parte, la selección de una heurística para problemas de empacado con más de un contenedor, depende fuertemente de los requerimientos establecidos. En especial, la heurística de prioridad propuesta en este documento, es una buena opción dado que nos permite indicarle al algoritmo cuales tipos de objetos nos interesa que sean empacados primeramente.

Durante la experimentación con los dos casos reales, se hizo evidente que cuando el número de artículos a empacar es muy grande, se requieren de mejores recursos computacionales para poder ejecutar el AGE. Sobretudo se hizo necesario aumentar la



velocidad de procesamiento, debido al incremento significativo en el número de operaciones que el algoritmo debía realizar. Es por esto que para resolver los problemas reales de empaçado el AGE requirió de más tiempo.

Además, fue necesario preprocesar los datos antes de correr el algoritmo, haciendo escalamiento y preempacado de los objetos. El escalamiento consiste en redimensionar los objetos involucrados en el empaçado, mientras que el preempacado consiste en agrupar objetos del mismo tipo. Estas dos simples operaciones simplifican el trabajo del AGE sin influir en la esencia de los problemas.

A pesar de las complicaciones, el AGE fue capaz de resolver ambos problemas de empaçado de una manera eficiente, proporcionando buenas soluciones. Esta forma automatizada de hacer optimización de carga de contenedores asegura que a cada artículo le sea asignada una posición única en el contenedor, lo cual es verdaderamente importante para garantizar que toda la mercancía sea empaçada. Este tipo de garantía no puede tenerse cuando se realiza carga de contenedores de manera empírica, como aún sigue realizándose en muchas compañías transportistas.

Dado que este algoritmo únicamente considera objetos en forma de paralelepípedos, en el futuro sería interesante:

- Diseñar otra estructura de cromosoma que pueda trabajar con objetos de diferentes formas.
- Incrementar el número de restricciones a considerar en el problema. Restricciones como el peso de los objetos, el orden en el cual deben ser descargados, formas diferentes de cargar los contenedores, etc.
- Mejorar el tiempo de procesamiento cambiando la forma en que se evalúa cada individuo.
- Encontrar mejores funciones objetivo que puedan considerar más restricciones, así como heurísticas para resolver problemas que involucren más de un contenedor.
- Evitar la utilización del preempacado y del escalamiento, permitiendo al algoritmo resolver el problema utilizando menos especificaciones.

## Bibliografía

- [1] J. Bean. A multiple-choice genetic algorithm for a nonlinear cutting stock problem. *Department of industrial and operations engineering, (University of Michigan):An Arbor, MI48109-2117, 2000.*
- [2] M. Carey and D. Johnson. *Computers and Intractability. A guide to the theory of NP-Completeness.* Freeman, New Jersey, 1979.
- [3] M. Gen and R. Cheng. *Genetic Algorithms and Engineering Optimization.* Wiley Series in Engineering Design and Automation, 1998.
- [4] J. George and D. Robinson. A heuristic for packing boxes into a container. *Computers and Operations Research, (Vol. 7):147-156, 1980.*
- [5] D. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, MA, 1989.
- [6] B. Golden. Approches to the cutting stock problem. *AIIE Transactions, 1976.*
- [7] C. Goulimis. Optimal solutions for the cutting stock problem. *Elsevier Science Publishers, 1988.*
- [8] D. Harald. A typology of cutting and packing problems. *European Journal of Operational Research, (Febrero 1989):145-159, 1990.*
- [9] R. Haupt and S. Haupt. *Practical Genetic Algorithms.* Wiley Interscience, NY, 1998.
- [10] C. Kao and F. Lin. A stochastic approach for the one-dimensional bin-packing problems. *IEEE International Conference on Systems, Man and Cybernetics, (Sept. 1993):1545 -1551, 1992.*
- [11] C. Pimpawat and N. Chaiyaratana. Using a co-operative co-evolutionary genetic algorithm to solve a three-dimensional container loading problem. *Proceedings of the 2001 Congress on Evolutionary Computation, pages 1197-1204, 2001.*

- [12] D. Pool. *Computational Intelligence, a logical approach*. Oxford University Press, NY, 2000.
- [13] M. A. Potter and K. A. De Jong. A cooperative coevolutionary approach to function optimization. *Proceedings of the Third International Conference on Parallel Problem Solving from Nature*, pages 249–257, 1994.
- [14] O. Takayuki. Approaches to 3d free-form cutting and packing problems and their applications: a survey.
- [15] Y. Tam, Y. Wu, W. Huang, and Wong. C. An effective quasi-human based heuristic for solving rectangle packing problem. *The 1998 IEEE Asia-Pacific Conference on Circuits and Systems*, pages 137–140, 1998.
- [16] T. Terno, G. Scheithauer, and U. Sommerweib. An efficient approach for the multi-pallet loading problem. *Institute for Numerical Mathematics, Technical University Dresden*, pages 1–17, 1997.
- [17] M. Valenzuela. *Apuntes de clase no publicados*. Monterrey, N.L., México, 2000.
- [18] D. Whitley, T. Starkweather, and D. Fuquay. Scheduling problems and traveling salesmen: The genetic edge recombination operator. *International Conference of Genetic algorithm, 1989*, pages 133–140, 1989.

Centro de Información-Biblioteca



30002006141964