

SEP

DGEST

INSTITUTO TECNOLÓGICO DE TIJUANA

**DIVISIÓN DE ESTUDIOS DE POSGRADO E
INVESTIGACIÓN**



**CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN
DE ALGORITMOS HEURÍSTICOS APLICADOS AL
PROBLEMA DE EMPACADO DE OBJETOS EN
CONTENEDORES**

TRABAJO DE TESIS

Presentado por
MARCELA QUIROZ CASTELLANOS

Para Obtener el Grado de
DOCTOR EN CIENCIAS EN COMPUTACIÓN

Director de Tesis
DRA. LAURA CRUZ REYES

Co-Director de Tesis
DR. JOSÉ TORRES JIMÉNEZ

Tijuana, BC, Julio, 2014

SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
Instituto Tecnológico de Tijuana

"2014, Año de Octavio Paz"

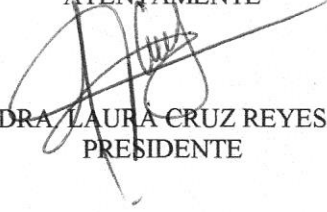
Asunto: Se autoriza impresión
de Trabajo de Tesis

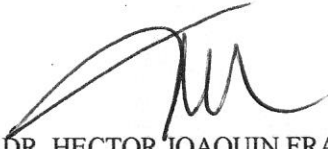
Tijuana, B.C., 10 de Junio del 2014

C. Lic. Doroteo Luna Castañeda
Jefe del Depto. de Servicios Escolares
Presente.

En lo referente al trabajo de tesis escrito, con título "**Caracterización del Proceso de Optimización de Algoritmos Heurísticos Aplicados del Problema de Empacado de Objetos en Contenedores**", presentado por la **C. Marcela Quiroz Castellanos**, alumna del Doctorado en Ciencias en Computación con número de control D02070989, informamos a usted que después de una minuciosa revisión, de acuerdo con lo establecido en el reglamento vigente para este caso, nuestro dictamen es: Se aprueba en todas sus partes, en virtud de reunir los requisitos de un trabajo de grado de Doctorado y a la vez se autoriza a la interesada para que proceda de inmediato a la impresión del mismo.

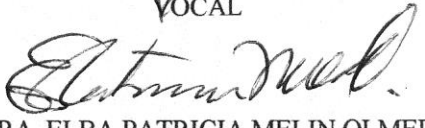
ATENTAMENTE


DRA. LAURA CRUZ REYES
PRESIDENTE


DR. HECTOR JOAQUIN FRAIRE HUACUJA
SECRETARIO


DR. JUAN JAVIER GONZALEZ BARBOSA
VOCAL


DR. OSCAR CASTILLO LOPEZ
VOCAL


DRA. ELBA PATRICIA MELIN OLMEDA
VOCAL

c.c.p. Oficina de Titulación
c.c.p. División de Estudios de Posgrado e Investigación
c.c.p. Expediente
c.c.p. Interesado

EPMO/*inf

Calz. Tecnológico S/N Esq. Calle Cuatémoc y Av. Castillo de Chapultepec,
Fracc. Tomás Aquino, C.P. 22414, Tijuana, Baja California. Tels.
(664)6078405, Conmutador (664)6078400 Ext. 101, e-mail:
direccion@tijuana.mx,
www.tectijuana.edu.mx



DECLARACIÓN DE ORIGINALIDAD

Yo, Marcela Quiroz Castellanos, en mi calidad de autor manifiesto que este documento de tesis es producto de mi trabajo original y que se realizó sin violar o usurpar derechos de terceros, tales como derechos de publicación, derechos de autor, patentes y similares. Por lo tanto, la obra es de mi exclusiva autoría y soy titular de los derechos que surgen de la misma.

Asimismo, declaro que en las citas textuales que he incluido y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y las publicaciones de referencia.

En caso de presentarse cualquier reclamación o acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión, asumiré toda la responsabilidad y relevo de ésta a mi director y codirectores de tesis, así como al Instituto Tecnológico de Tijuana y sus autoridades.

Tijuana, BC, Julio, 2014

A handwritten signature in black ink, consisting of a large, stylized 'M' and 'Q' followed by 'C' and 'Castellanos' in a cursive script.

M.C. Marcela Quiroz Castellanos

RESUMEN

A pesar de los esfuerzos de la comunidad científica en el desarrollo algoritmos de alto desempeño, a la fecha, no existe un algoritmo heurístico que sea capaz de encontrar la mejor solución para todas las posibles instancias de un problema NP-duro. Trabajos recientes en el área de algoritmia experimental han destacado la necesidad de explicar el desempeño de los algoritmos heurísticos. Sin embargo, para explicar el comportamiento de un algoritmo heurístico es necesario identificar y estudiar los factores que lo afectan. El objetivo principal de esta investigación es promover la caracterización y el análisis de los problemas NP-duros y las heurísticas que los solucionan, para generar conocimiento, comprender la dificultad de las instancias y diseñar mejores algoritmos de solución.

En este trabajo se formula un ***enfoque experimental*** para un *estudio integral del proceso de optimización*, con la finalidad de identificar relaciones inherentes entre los factores que afectan al desempeño algorítmico. El enfoque propuesto combina métodos de análisis exploratorio de datos y de análisis causal en cuatro etapas. En la etapa de *caracterización* se identifican factores que influyen en el desempeño y se cuantifican mediante índices. En la etapa de *refinación de características* se descartan índices incorrectos e irrelevantes y se generan nuevos índices. En la etapa denominada *estudio de relaciones* se realiza un análisis de las características del proceso de optimización con el propósito de obtener relaciones de desempeño que, posteriormente, se convierten en *explicaciones del comportamiento algorítmico*.

Para evaluar la contribución del enfoque experimental aquí propuesto, se analizó el proceso de optimización del problema de empaqueo de objetos en contenedores (BPP). El análisis permitió identificar propiedades que definen la estructura de una instancia de BPP y explicar el desempeño de los algoritmos. El conocimiento obtenido del análisis fue utilizado para generar nuevos conjuntos de instancias difíciles y para diseñar nuevas estrategias inteligentes que fueron integradas en un algoritmo denominado GGA-CGT. Resultados experimentales revelan que GGA-CGT supera la efectividad de los mejores algoritmos del estado del arte; en particular, obtiene la solución óptima de 11 instancias que los mejores algoritmos no pueden encontrar. La generalidad del enfoque de análisis experimental posibilita analizar otros problemas de optimización NP-duros. Así mismo, GGA-CGT podría ser utilizado para solucionar otros problemas de agrupación.

ABSTRACT

Throughout the search for the best possible solutions for NP-hard problems, a wide variety of heuristic algorithms have been proposed, however, despite the efforts of the scientific community to develop new strategies, there is no efficient algorithm capable of finding the best solution for all possible situations. Recent works in experimental analysis of heuristics have identified the need to explain the observed performance. To understand the behavior of a heuristic algorithm is necessary to characterize and study the factors that affect it. The main objective of this research is to promote the characterization and analysis of the NP-hard problems and the heuristics used to solve them, for generating knowledge, understanding the degree of difficulty of the instances and designing better algorithms.

In this work we formulate an ***experimental approach*** for a comprehensive study of the optimization process, in order to identify inherent relationships among the factors that affect the algorithmic performance. The proposed approach combines methods of exploratory data analysis and causal inference in three stages. In the *characterization* phase factors that influence performance are identified and quantified by indexes. In the *characteristics refining* phase, incorrect and redundant indexes are discarded. In the *study of relations* phase, an analysis of the characteristics of the optimization process is made in order to obtain performance relations that eventually will become algorithm behavior explanations.

To evaluate the contribution of the proposed approach we performed a study of the optimization process for the Bin Packing Problem (BPP). The analysis allowed us to identify the characteristics that define the structure of a BPP instance and to understand the performance of the heuristics. The knowledge gained from the analysis was used to generate new sets of difficult instances and to design new high-impact strategies that were incorporated into a high-performance algorithm referred to as GGA-CGT. The experimental results show that GGA-CGT outperforms the effectiveness of the best algorithms on the state-of-the-art; in particular, GGA-CGT obtains the optimal solution of 11 instances that the best algorithms cannot find. The generality of the approach of experimental analysis would allow to analyze other NP-hard problems. Furthermore, GGA-CGT could be used to solve other grouping problems.

APORTACIONES DEL TRABAJO DE TESIS

1. Artículo “Análisis de Relaciones Causales entre los Parámetros de un Algoritmo Genético y su Desempeño con Problemas de Empacado”. **Presentado en:** 17th International Congress on Computer Science Research CIICC'11.
2. Artículo "Improving the Performance of Heuristic Algorithms Based on Causal Inference". **Presentado en:** 10th Mexican International Conference on Artificial Intelligence, MICA-2011 y **publicado en** *Advances in Artificial Intelligence*, Springer Berlin Heidelberg, 2011, p. 137-148.
3. Artículo “Enhancing Accuracy of Hybrid Packing Systems through General-Purpose Characterization”. **Publicado en:** *Hybrid Artificial Intelligent Systems*, Springer Berlin Heidelberg, 2011, p. 26-33.
4. Artículo “Heurísticas de Agrupación Híbridas Eficientes para el Problema de Empacado de Objetos en Contenedores”. **Publicado en** *Computación y Sistemas*, 2012, vol. 16, no 3, p. 349-360.
5. Capítulo de libro "Algorithm Selection: from Meta-Learning to Hyper-heuristics". **Publicado en:** *Intelligent Systems*, 2012, p. 77. ISBN 979-953-307-593-7.
6. Capítulo de libro "Heuristic Algorithms: An Application to the Truck Loading Problem". **Publicado en:** *Logistics Management and Optimization through Hybrid Artificial Intelligence Systems*, 2012, p. 238-267. ISBN13: 9781466602977.
7. Artículo "Improving the Performance of Heuristic Algorithms Based on Exploratory Data Analysis". **Publicado en:** *Recent Advances on Hybrid Intelligent Systems*. Springer Berlin Heidelberg, 2013, p. 361-375.
8. Artículo "A visualization tool for heuristic algorithms analysis". **Publicado en:** *7th International Conference on Knowledge Management in Organizations: Service and Cloud Computing*, Springer Berlin Heidelberg, 2013, vol. 172, p. 515-524.
9. Artículo "Characterization of the Optimization Process". **Publicado en:** *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*. Springer International Publishing, 2014, p. 493-507.
10. Artículo "A Grouping Genetic Algorithm with Controlled Gene Transmission for the Bin Packing Problem". **Publicado en:** *Computers & Operations Research*, Elsevier, 2015, vol. 55, p. 52-64.

CONTENIDO

DECLARACIÓN DE ORIGINALIDAD	<i>i</i>
RESUMEN	<i>ii</i>
ABSTRACT	<i>iii</i>
APORTACIONES DEL TRABAJO DE TESIS	<i>iv</i>
CAPÍTULO 1: INTRODUCCIÓN.....	1
1.1 ANTECEDENTES	3
1.2 DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN	5
1.3 HIPÓTESIS	6
1.4 JUSTIFICACIÓN	7
1.4.1 ESTABLECIMIENTO DE TEORÍAS.....	7
1.4.2 APLICACIÓN DEL CONOCIMIENTO OBTENIDO.....	9
1.5 OBJETIVOS	11
1.6 ALCANCES Y LIMITACIONES	12
1.7 ORGANIZACIÓN DEL DOCUMENTO	12
CAPÍTULO 2: MARCO TEÓRICO.....	14
2.1 PROBLEMAS DE OPTIMIZACIÓN Y ALGORITMOS HEURÍSTICOS	15
2.2 ANÁLISIS EXPERIMENTAL DEL DESEMPEÑO DE ALGORITMOS	16
2.3 PROCESO DE OPTIMIZACIÓN Y SU CARACTERIZACIÓN.....	18
2.4 EL PROBLEMA DE EMPACADO DE OBJETOS EN CONTENEDORES.....	21
CAPÍTULO 3: ESTADO DEL ARTE	24
3.1 CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN	25
3.1.1 ANÁLISIS DE DATOS EXPLORATORIO.....	25
3.1.2 ANÁLISIS DE LA ESTRUCTURA DEL ESPACIO DE BÚSQUEDA	26
3.1.3 ESTUDIO DE FENÓMENOS DE TRANSICIÓN DE FASE	27
3.1.4 ANÁLISIS DE LA ESTRUCTURA DE LAS SOLUCIONES ÓPTIMAS	28
3.1.5 ANÁLISIS DEL PROCESO DE OPTIMIZACIÓN	29
3.2 EL PROCESO DE OPTIMIZACIÓN DE BPP	30
3.2.1 ALGORITMOS HEURÍSTICOS PARA SOLUCIÓN DE BPP	30
3.2.2 INSTANCIAS DE PRUEBA	34
3.2.3 RESULTADOS EXPERIMENTALES DE ALGORITMOS	37
3.2.4 INDICES DE CARACTERIZACIÓN DE BPP	39
CAPÍTULO 4 CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN: UN NUEVO ENFOQUE DE ANÁLISIS EXPERIMENTAL.....	42

4.1	ENFOQUE EXPERIMENTAL PARA CARACTERIZAR Y EXPLICAR EL PROCESO DE OPTIMIZACIÓN ALGORÍTMICO	43
4.1.1	ETAPA 1: CARACTERIZACIÓN	44
4.1.2	ETAPA 2: REFINACIÓN DE CARACTERÍSTICAS	45
4.1.3	ETAPA 3: ESTUDIO DE RELACIONES	46
4.2	ANÁLISIS DEL PROCESO DE OPTIMIZACIÓN DE BPP	47
4.2.1	ETAPA 1. CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN DE BPP	47
	Caracterización de BPP	47
	Caracterización del Comportamiento Algorítmico.....	51
	Caracterización del Desempeño Final	57
4.2.2	ETAPA 2. REFINACIÓN DE CARACTERÍSTICAS.....	60
	Análisis de Correlación	62
	Análisis de Aportación y Consistencia.....	63
	Validación del Conjunto Final de Índices	68
4.2.3	ETAPA 3. ESTUDIO DE RELACIONES	72
4.3	EXPLICACIONES DEL COMPORTAMIENTO ALGORÍTMICO.....	82
4.3.1	GENERACIÓN DE INDIVIDUOS.....	82
4.3.2	SUSTITUCIÓN Y CONSERVACIÓN DE INDIVIDUOS	84
4.3.3	COMPARACIÓN DE HGGA-BP ORIGINAL Y REDISEÑADO	86
4.4	CONCLUSIONES DEL ESTUDIO	87
CAPÍTULO 5: DIFICULTAD EN INSTANCIAS DE BPP: NUEVOS ÍNDICES Y CASOS DE PRUEBA.....		88
5.1	INSTANCIAS ESTÁNDAR DIFÍCILES	88
5.2	NUEVO ÍNDICE DE CARACTERIZACIÓN DE BPP: <i>Gap</i>	91
5.3	NUEVAS INSTANCIAS DIFÍCILES TIPO HARD28	93
5.4	NUEVO ÍNDICE DE CARACTERIZACIÓN DE BPP: <i>Pct_ñ</i>	95
5.5	NUEVAS INSTANCIAS DIFÍCILES <i>u-w1-wn</i>	97
5.6	NUEVAS INSTANCIAS DIFÍCILES TIPO UNIFORM	99
5.7	CONCLUSIONES DEL ESTUDIO	100
CAPÍTULO 6: UN ALGORITMO GENÉTICO PARA BPP CON TRANSMISIÓN DE GENES CONTROLADA.....		102
6.1	COMPONENTES DE GGA-CGT	104
6.1.1	CODIFICACIÓN GENÉTICA	104
6.1.2	FUNCIÓN DE APTITUD	105
6.1.3	POBLACIÓN INICIAL.....	106
6.1.4	OPERADOR DE CRUZAMIENTO POR AGRUPACIÓN	107
6.1.5	OPERADOR DE MUTACIÓN POR AGRUPACIÓN	111
6.1.6	HEURÍSTICAS DE REACOMODO PARA OPERADORES DE AGRUPACIÓN	113
6.2	TÉCNICA DE REPRODUCCIÓN.....	116
6.2.1	ESQUEMA DE SELECCIÓN.....	117
6.2.2	ESTRATEGIA DE REEMPLAZO.....	119
6.2.3	COMPARACIÓN DE REPRODUCCIÓN CONTROLADA CON OTRAS TÉCNICAS	120

6.3	EXPERIMENTOS COMPUTACIONALES.....	122
6.3.1	COMPARACIÓN DE GGA-CGT CON UN GENÉTICO DE AGRUPACIÓN.....	124
6.3.2	COMPARACIÓN DE GGA-CGT CON EL ESTADO DEL ARTE	126
6.3.3	ANÁLISIS DEL DESEMPEÑO DE GGA-CGT	127
CAPÍTULO 7: CONCLUSIONES Y TRABAJOS FUTUROS		133
7.1	CONCLUSIONES.....	133
7.2	TRABAJO FUTURO	136
REFERENCIAS BIBLIOGRÁFICAS		138

Capítulo 1

INTRODUCCIÓN

En cualquier entidad, la eficiencia y el alto desempeño son la clave para alcanzar el éxito. Sin embargo, un alto porcentaje de las tareas vitales para el funcionamiento de una organización involucran problemas de optimización complejos, difíciles de resolver. Para favorecer la obtención de mejores beneficios, estos problemas deben ser resueltos mediante el uso de herramientas computacionales capaces de adaptarse a diferentes escenarios y obtener buenas soluciones sin consumir altos volúmenes de recursos.

Dada la importancia de los problemas de optimización se han realizado diversos estudios, con el fin de proporcionar soluciones que permitan la minimización de costos. Sin embargo, el proceso de solución requerido presenta una alta complejidad, pues en la mayoría de los casos, los problemas del mundo real pertenecen a una clase especial de problemas denominados NP-duros [Garey79], lo cual implica que no se conocen algoritmos eficientes para resolverlos de manera exacta en el peor caso.

Para la solución de problemas de optimización NP-duros, se ha propuesto gran variedad de algoritmos heurísticos, los cuales han mostrado un desempeño satisfactorio. Sin embargo, a pesar de los esfuerzos de la comunidad científica en el desarrollo de nuevas

estrategias, a la fecha no existe un algoritmo que sea la mejor opción para todas las posibles situaciones [Wolpert97]. Para este tipo de problemas, el diseño de algoritmos adecuados a condiciones específicas suele ser la mejor opción. Uno de los retos principales del análisis de comportamiento de los algoritmos heurísticos consiste en identificar cuáles estrategias hacen que un algoritmo muestre un mejor desempeño y bajo qué condiciones lo obtienen.

Gran parte de los recientes progresos en el desarrollo de algoritmos ha sido facilitado por una mejor comprensión de las propiedades de las instancias de los problemas de optimización y del desempeño de los algoritmos que las solucionan, ejemplo de ello son las numerosas investigaciones sobre la complejidad del problema de satisfacción de restricciones (SAT, por su sigla en inglés) y el desempeño de sus algoritmos de solución [Smyth04]. Sin embargo, el porcentaje de trabajos de investigación llevados a cabo para comprender cómo y porqué los algoritmos siguen cierto comportamiento, es mucho menor que el de trabajos realizados para crear dichos algoritmos [Chiarandini07].

La mayoría de los estudios de desempeño algorítmico se han centrado exclusivamente en identificar conjuntos de instancias de diferente grado de dificultad, y en la reducción del tiempo necesario para resolver esos casos. Si bien, este es un objetivo importante, la mayoría de los enfoques han sido bastante particulares, mostrando un mayor énfasis en ajustar las configuraciones de los algoritmos que en comprender los motivos subyacentes de su desempeño.

El principal problema abordado en este trabajo es cómo ampliar los conocimientos actuales sobre la relación entre las características de los problemas de optimización, el espacio de búsqueda que los define, el comportamiento de los algoritmos que los solucionan, y el desempeño final alcanzado por dichos algoritmos, en particular para algoritmos del estado del arte que resuelven el problema de empaqueo de objetos en contenedores. El objetivo final de esta investigación es utilizar estos conocimientos para mejorar los métodos de caracterización existentes y contribuir al desarrollo de técnicas

que permitan realizar un análisis sólido del desempeño de algoritmos heurísticos y proporcionen fundamentos que faciliten el establecimiento de teorías y el diseño de mejores algoritmos e instancias de prueba.

1.1 ANTECEDENTES

Este trabajo se plantea en el contexto de un proyecto de investigación sobre algoritmia experimental llevado a cabo en la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Cd. Madero. A la fecha, en el marco de dicho proyecto, se han desarrollado dos tesis de doctorado y siete tesis de maestría; dichas tesis han abordado aspectos relacionados con el análisis experimental de algoritmos heurísticos, con el fin de explicar la conducta algorítmica observada. En particular se tiene interés en estudiar la familia de problemas de asignación. Los problemas abordados han sido: Bin Packing Problem, Knapsack Problem y Distributed Fragmentation, Assignment and Reallocation; también conocidos por respectiva sigla en inglés como BPP, KP y DFAR.

El primer trabajo [Cruz04] consistió en caracterizar un conjunto de algoritmos con la finalidad de crear modelos de predicción, de tal manera que para una nueva instancia de BPP se pudiera seleccionar el algoritmo que mejor la resolviera. Los resultados obtenidos fueron satisfactorios; sin embargo, estos no permitían explicar por qué un algoritmo era mejor que otros para resolver una instancia dada.

La metodología para la selección de algoritmos propuesta por Cruz [Cruz04] se extendió en cinco tesis de maestría. Santiago aplica la metodología de selección al problema DFAR [Santiago04]. Landero explora varias técnicas de clasificación y agrupación para formar grupos de dominación; cada grupo de instancias es asociado con el algoritmo que mejor las resuelve [Landero04]. García incorpora retroalimentación a la arquitectura de selección y caracteriza el comportamiento algorítmico utilizando índices de superficie de aptitud tomados de la literatura [García04]. Posteriormente, Álvarez incorpora nuevos índices de caracterización de BPP derivados de la estadística descriptiva e introduce la

multietiqueta para tratar grupos dominantes traslapados [Álvarez06]. Luego, Ferat desarrolla un sistema de clasificación basado en redes neuronales para la selección dinámica y autoadaptiva de algoritmos heurísticos [Ferat07].

Más adelante, Pérez presenta una metodología para el análisis experimental de algoritmos heurísticos donde incorpora algunas técnicas de análisis multivariadas y modelado causal [Pérez07]. Paralelamente, Landero propone una metodología semejante que incorpora un conjunto mayor de índices de caracterización y muestra la aplicabilidad del modelado causal en la selección de algoritmos [Landero08]. Los resultados obtenidos en ambos trabajos fueron satisfactorios, sin embargo, las explicaciones de desempeño fueron un tanto limitadas, pues no se incluyen todos los tipos de índices del problema y del algoritmo, así mismo los algoritmos tratados en ambos trabajos fueron versiones básicas de estrategias metaheurísticas y ninguna de las metodologías fue probada para explicar algoritmos de alto desempeño.

Recientemente, García muestra la aplicabilidad de la caracterización del comportamiento algorítmico al desarrollar dos índices de la intensidad del uso de heurísticas dentro de un algoritmo hiper-heurístico, el estudio experimental de estos índices permitió identificar la necesidad de incluir un mayor uso de técnicas de exploración para mejorar el desempeño algorítmico [García10].

El análisis de los trabajos de investigación descritos en los párrafos anteriores, reveló que hasta ahora, no se han conjuntado la caracterización del problema, el proceso de solución y el rendimiento de algoritmos heurísticos de alto desempeño. Así mismo, se observó que la calidad del conocimiento obtenido depende principalmente del poder de caracterización y de la validez de los índices de caracterización propuestos. En el trabajo de investigación aquí propuesto, se combinan diferentes técnicas de caracterización en un enfoque de análisis experimental para obtener un conocimiento detallado sobre el comportamiento de algoritmos heurísticos de alto desempeño y la dificultad de las instancias que solucionan. Para probar el enfoque de análisis, se aborda el problema de

empacado de objetos en contenedores, BPP (por su sigla en inglés). Este problema de optimización tiene un amplio número de aplicaciones industriales y logísticas, además de que frecuentemente se presenta como un sub-problema en un extenso número de problemas prácticos. BPP es de especial interés para los investigadores del ITCM ya que su formulación forma parte de varios sistemas logísticos en desarrollo, relacionados con la distribución de productos embotellados y el embarque de contenedores en terminales portuarias [Cruz07]. Debido a lo anterior, BPP es objeto de estudio del análisis aquí propuesto.

1.2 DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN

El problema de explicar formalmente porqué un algoritmo sigue cierto comportamiento al resolver un conjunto instancias de un problema se puede describir de la siguiente manera.

Dados:

1. un algoritmo heurístico A para resolver un problema P ,
2. un conjunto de instancias $I = \{i_1, i_2, \dots, i_n\}$ del problema P .

Se busca:

1. un conjunto de indicadores $F = \{f_1, f_2, \dots, f_m\}$ que caractericen factores del problema y del algoritmo que definan el proceso de optimización algorítmico,
2. un conjunto de relaciones $R = \{r_1, r_2, \dots, r_p\}$, para $1 \leq p \leq m$, tal que cada r_i establece una relación causal entre los elementos de G_i , donde $G_i \subseteq F$ que incluye factores que impactan el desempeño algorítmico,
3. aplicar el conocimiento adquirido a través de relaciones causales R , al desarrollo de nuevas instancias del problema y al diseño de mejores estrategias de solución.

En otras palabras, se busca un conjunto de factores críticos que caractericen el proceso de optimización algorítmico y permitan explicar porqué el algoritmo objeto de estudio sigue cierto comportamiento al resolver un problema de optimización y aplicar el conocimiento adquirido en: 1) la creación de nuevas instancias representativas del problema, 2) el diseño de un algoritmo de alto desempeño.

1.3 HIPÓTESIS

Las hipótesis de investigación, en el marco del proceso de solución al problema planteado en este proyecto de investigación, se describen a continuación:

H1: Es posible identificar un conjunto de factores críticos que definan la estructura de las instancias de BPP, la estructura y comportamiento de un algoritmo de solución, así como el desempeño final alcanzado por este algoritmo mediante caracterización basada en índices.

H2: Es posible encontrar un conjunto de relaciones entre los factores que impactan el proceso de optimización, y explicar porqué un algoritmo heurístico de alto desempeño muestra cierto comportamiento al resolver un conjunto instancias de BPP, mediante la aplicación de estadística multivariada, análisis causal y análisis exploratorio de datos.

H3: Es posible generar instancias representativas de la estructura de BPP, mediante la caracterización del problema y las propiedades que definen una instancia difícil para los algoritmos de solución.

H4: Es posible diseñar un nuevo algoritmo de solución de BPP que muestre un alto nivel de competencia con respecto a los mejores procedimientos del estado del arte, usando los conocimientos obtenidos a través de la caracterización del proceso de optimización de algoritmos de solución en la formulación de nuevas estrategias que permitan alcanzar un alto desempeño.

1.4 JUSTIFICACIÓN

Los primeros retos en el campo de las ciencias de la computación estuvieron estrechamente relacionados con la ingeniería, dando lugar al desarrollo de algoritmos y sistemas de alta complejidad. Sin embargo, diseñar algoritmos y construir herramientas no significa que realmente se tenga un conocimiento y comprensión de éstos. Estudios especiales y mucho trabajo son necesarios para caracterizar correctamente los algoritmos desde una perspectiva científica. En esta sección se muestra la importancia del trabajo de investigación aquí propuesto, el cual contribuirá a promover la aplicación de técnicas empíricas de análisis dentro de las ciencias de la computación, facilitando la obtención de teorías que fundamenten los resultados y comportamientos observados en algoritmos heurísticos que solucionan problemas de optimización. Finalmente, se presentan algunas aplicaciones prácticas del conocimiento que puede obtenerse producto del análisis del proceso de optimización algorítmico.

1.4.1 ESTABLECIMIENTO DE TEORÍAS

Las ciencias de la computación representan una disciplina joven y en constante evolución, que ha sido concebida de diferentes formas por investigadores de diversas áreas, generando informes y discusiones con percepciones variadas sobre si se trata verdaderamente de una ciencia [Denning89, Simon96, Denning05, Feitelson06].

Para que esta disciplina sea considerada una ciencia debería integrar tres perspectivas igualmente ascendentes: ciencia, matemáticas e ingeniería [Denning05, Feitelson06]. Sin embargo, las investigaciones en las ciencias de la computación normalmente son parte de las últimas dos perspectivas [Feitelson06, Snodgrass10]. Las matemáticas son aplicadas mediante formalismos, teorías y algoritmos. La ingeniería se presenta a través del esfuerzo constante por construir mejores algoritmos y herramientas, más rápidos, más sencillos y más económicos. Pero las ciencias de la computación han mostrado muy pocos ejemplos de la primera perspectiva, aquella que consiste en observar, describir y entender fenómenos, por lo que parecen no haber alcanzado aún la estructura y el grado

de entendimiento al que las ciencias aspiran y por los cuales son reconocidas [Feitelson06, Snodgrass10].

Las ciencias de la computación han puesto al descubierto profundas verdades matemáticas y han construido magníficos sistemas que han dado paso a la “era de las tecnologías de la información”. Sin embargo, aún existe muy poco conocimiento sobre el comportamiento de estos sistemas de alta complejidad, la forma en la que se relacionan, porqué interactúan de cierta forma en diferentes entornos, cuáles son las limitaciones inherentes que los restringen o de qué manera pueden ser utilizados para construir mejores sistemas [Snodgrass10].

Dado que la ciencia está relacionada con el descubrimiento de las leyes que gobiernan el universo y consiste en desarrollar teorías predictivas generales que describan y expliquen fenómenos observados, así como en evaluar dichas teorías [Chalmers99], la perspectiva científica puede ayudar a comprender las herramientas computacionales y la computación misma a través del descubrimiento de principios generales que revelen nuevos enfoques y permitan identificar limitaciones en algoritmos y herramientas y predecir conductas futuras.

Un ejemplo de un modelo computacional que fue producto de una generalización empírica es la *teoría de la localidad* [Denning05]. Este modelo surgió de un estudio de los costos de la gestión de transferencias de páginas entre la memoria principal y una unidad de disco, en el área de sistemas operativos. En este estudio, Denning observó que los datos relevantes para el contexto de un programa en ejecución tienden a agruparse en el espacio y tiempo en fragmentos locales, y que esto es un efecto de la forma en que los seres humanos escriben los programas, más que de las restricciones subyacentes del sistema de cómputo. Cuando los algoritmos de gestión de memoria respetan este modelo general el rendimiento mejora dramáticamente. La teoría resultante, que los sistemas de procesamiento de información construidos por humanos exhibirán localidad, es

inherentemente predictiva y ha sido probada y generalizada para aplicarse en otros sistemas, por ejemplo: memoria virtual, navegadores web, entre otros [Denning05].

Para lograr establecer la teoría de la localidad, Denning primero reconoció un tipo de comportamiento que ocurría en una clase de sistemas y luego buscó factores responsables de la conducta observada. Para encontrar dichos factores, realizó experimentos, identificó y ajustó factores candidatos, recolectó datos y los analizó cuidadosamente y fue actualizando sus hipótesis. Él revisó su modelo basado en análisis experimentales mientras lo expandía a una clase más amplia de sistemas que exhiben localidad. De este modo, Denning fue capaz de lograr una mayor claridad sobre este fenómeno y sobre cómo y por qué un sistema de cómputo lo presenta, para finalmente crear una teoría.

El ejemplo presentado en los párrafos anteriores muestra la importancia de utilizar una metodología empírica para entender el comportamiento de sistemas y algoritmos. Debido a lo anterior, diversos autores han destacado la necesidad de incluir verdaderas “técnicas de ciencia” en las ciencias de la computación para consolidar las teorías, algoritmos y sistemas obtenidos [Feitelson06, Morrison10, Snodgrass10]. La aplicación de métodos empíricos de análisis ayudará a explicar el desempeño de los algoritmos, descubrir sus limitaciones, y predecir su comportamiento futuro. Se espera que la investigación a desarrollar en el trabajo aquí propuesto permita contribuir en este proceso de consolidar las ciencias de la computación como una verdadera ciencia.

1.4.2 APLICACIÓN DEL CONOCIMIENTO OBTENIDO

La caracterización del proceso de optimización algorítmico además de ayudar a obtener explicaciones de desempeño, permite adquirir un conocimiento a detalle de cada una de las fases de este proceso. El conocimiento de la estructura del problema y del algoritmo de solución, los valores que los definen y las relaciones entre estos valores, posibilita, a

su vez, comprender la conducta del algoritmo y el porqué de los resultados obtenidos. Este conocimiento puede ser aplicado en áreas como:

- *Clasificación de instancias.* En el trabajo de Cruz [Cruz04] se propone el siguiente método de agrupación de instancias: Dados un conjunto de casos de un problema de optimización combinatoria y un conjunto de dos o más algoritmos de solución heurística, se forman grupos de casos por afinidad de desempeño algorítmico. Es así que las instancias se pueden clasificar por su grado de dificultad, distinguiendo entre instancias fáciles y difíciles para el conjunto total de algoritmos, y aquellas que tienen que ver con el algoritmo que las soluciona.
- *Explicación del desempeño de algoritmos.* El identificar los factores que influyen en el desempeño de un algoritmo y establecer las relaciones que existen entre ellos posibilita la explicación del comportamiento de éste al resolver un problema real. Lo anterior permite comprender el funcionamiento del algoritmo ante diferentes tipos de problemas y el porqué de su buen o mal desempeño [Pérez07, Landero08].
- *Pronósticos de desempeño.* Dado que no se conoce un algoritmo heurístico que sea la mejor opción para todas las posibles situaciones, es necesario elegir el algoritmo más adecuado para un problema específico. Con el conocimiento de las relaciones de desempeño, podría ser posible, al llegar un nuevo caso de un problema, pronosticar el comportamiento de un conjunto de algoritmos y elegir el mejor. Esto posibilitaría un ahorro de tiempo en la solución de problemas, obteniendo buenos resultados [Cruz04, Álvarez06].
- *Rediseño de algoritmos.* El entendimiento del comportamiento de un algoritmo, así como de su diseño lógico puede sugerir cambios, tanto en la estructura del algoritmo, como en los parámetros que lo controlan, para mejorar su desempeño ante distintos tipos de problemas [Quiroz09].

- *Desarrollo de nuevos procedimientos de solución.* Los resultados mostrados por los algoritmos podrían ser mejorados al identificar las estrategias que hacen a un algoritmo superior a otro ante ciertas condiciones e integrar dichas estrategias de manera complementaria, por ejemplo mediante hibridación.

1.5 OBJETIVOS

A continuación se presentan el objetivo general y los objetivos específicos planteados para esta investigación.

Objetivo General

Estudiar el desempeño de algoritmos del estado del arte que dan solución al problema de empaçado de objetos en contenedores (BPP) mediante la caracterización del proceso de optimización algorítmico.

Objetivos Específicos

Para lograr alcanzar el objetivo principal planteado en el párrafo anterior se han definido los siguientes objetivos específicos:

1. Estudiar diferentes técnicas que permitan la caracterización de la estructura del problema, el comportamiento de los algoritmos, la estructura de las soluciones óptimas y el desempeño final.
2. Caracterizar el problema de empaçado de objetos en contenedores, BPP, e identificar las propiedades que definen una instancia difícil de resolver.
3. Caracterizar el comportamiento de algoritmos de solución de BPP e identificar estrategias y condiciones que permiten alcanzar un alto desempeño.
4. Obtener un modelo de relaciones entre las características del problema y del comportamiento algorítmico que impactan en el desempeño.

5. Diseñar un nuevo conjunto de instancias de prueba aleatorias y representativas de la estructura caracterizada de BPP.
6. Incorporar conocimiento del dominio del problema en un algoritmo de solución de BPP para alcanzar un alto nivel de competencia con respecto a las mejores estrategias del estado del arte.

1.6 ALCANCES Y LIMITACIONES

1. La caracterización del proceso de optimización se aplicará sólo al problema de empaqueo de objetos de contenedores de una dimensión 1D-BPP y algoritmos de solución del estado del arte relevantes para este problema.
2. Dada la naturaleza de BPP, para algunas de las instancias de prueba que se diseñen podría no ser posible encontrar la solución óptima, mediante algoritmos exactos, por el excesivo tiempo de cómputo requerido. Para estas instancias, el análisis comparativo se basará en un límite inferior del valor óptimo.
3. Los factores identificados pudieran ser un subconjunto de los factores reales que afectan el desempeño. La literatura destaca que incluso los análisis más detallados y profundos no garantizan la identificación total de todos los posibles factores [Carnegie09].

1.7 ORGANIZACIÓN DEL DOCUMENTO

El documento está organizado de la siguiente manera: en el Capítulo 2 se presenta una breve revisión de los fundamentos teóricos necesarios para el análisis experimental de algoritmos metaheurísticos. Se introducen los principales fundamentos del análisis del comportamiento de algoritmos y de la caracterización del proceso de optimización y se describe el problema de empaqueo de objetos en contenedores y las heurísticas aplicadas para resolver este problema.

En el Capítulo 3 se presenta una reseña de trabajos relacionados con la caracterización del proceso de optimización, así como de los enfoques propuestos para dicho estudio. Así mismo, se incluye un análisis comparativo de trabajos que estudian el desempeño de algoritmos heurísticos en problemas NP-duros. Finalmente se describen los algoritmos sobresalientes del estado del arte para la solución de BPP, las características con las que fueron generadas las instancias de prueba estándar y los índices de caracterización que han sido propuestos para estudiar la dificultad de las instancias de este problema.

En el Capítulo 4 se detalla el enfoque de análisis experimental propuesto y su aplicación en el estudio del proceso de optimización del algoritmo genético híbrido HGGA-BP. Así mismo, se presenta el rediseño del algoritmo HGGA-BP obteniendo como producto final una nueva versión del algoritmo que muestra una mejora significativa en el desempeño. El Capítulo 5 presenta tres nuevos índices de caracterización de BPP que complementan los presentados en el Capítulo 4 para describir la dificultad de las instancias; así mismo se proponen tres nuevos conjuntos de casos de prueba representativos de la estructura caracterizada de BPP. En el Capítulo 6 se presenta un nuevo algoritmo genético para BPP, denominado GGA-CGT, que incorpora el conocimiento obtenido del análisis del proceso de optimización de BPP para obtener resultados de calidad, superando a los mejores algoritmos del estado del arte.

En el Capítulo 8 se presentan las conclusiones de la investigación realizada, así como sugerencias para trabajos futuros.

Capítulo 2

MARCO TEÓRICO

En este capítulo se presenta una revisión de los conceptos y enfoques relacionados con problemas de optimización combinatoria y su solución con algoritmos heurísticos. Así mismo, se describen las principales técnicas empleadas para la caracterización y análisis del proceso de optimización algorítmico.

La mayoría de los trabajos relacionados con el análisis de desempeño de algoritmos heurísticos, están enfocados en análisis comparativos de resultados experimentales. No obstante, trabajos recientes en el área han señalado la necesidad de explicar el porqué de los resultados observados [Borenstein05, Chiarandini07, Snodgrass10]. Para lograr comprender y explicar el comportamiento de un algoritmo es necesario identificar y estudiar los factores que lo afectan.

El objetivo principal de la caracterización de algoritmos es entender de qué manera el desempeño de un algoritmo es afectado por una serie de factores que lo causan. El conocimiento adquirido puede conducir a mejores predicciones del desempeño ante nuevas situaciones y a la definición de algoritmos mejorados.

2.1 PROBLEMAS DE OPTIMIZACIÓN Y ALGORITMOS HEURÍSTICOS

Los problemas *NP-duros* son de gran interés en las ciencias computacionales. Una de las características de los problemas de este tipo es que los algoritmos exactos empleados para resolverlos requieren una cantidad exponencial de tiempo en el peor de los casos. En otras palabras estos problemas son muy difíciles de resolver [Garey79]. Al resolver problemas NP-duros es necesario conformarse con soluciones “buenas”, que en algunos casos pueden obtener resultados óptimos. En estas condiciones se hace uso de algoritmos que dan soluciones aproximadas en un tiempo razonable, pero no garantizan obtener la mejor solución.

Los *algoritmos heurísticos* utilizan estrategias llamadas heurísticas, para obtener soluciones de calidad (no necesariamente óptimas) a problemas complejos de manera eficiente. Las *heurísticas* son procedimientos basados en el sentido común, que ofrecen una buena solución a problemas (particulares) difíciles, de un modo fácil y rápido [Díaz96]. Se dice que los algoritmos heurísticos son *deterministas* cuando sólo pueden encontrar una única solución para cada instancia de un problema (sin importar cuántas veces se ejecute el algoritmo, la solución encontrada será siempre la misma) a diferencia que los algoritmos *no-deterministas* o *aleatorios*.

Un *algoritmo metaheurístico* es un algoritmo heurístico no determinista de propósito general, que combina diferentes heurísticas definidas de acuerdo al problema. Estos algoritmos parten de soluciones iniciales y, mediante alteraciones inteligentes, exploran soluciones vecinas de su entorno en búsqueda de mejores soluciones [Michalewicz04, Dréo06, Gonzalez07]. Existen procedimientos que combinan estrategias heurísticas de diferente naturaleza, dichas técnicas son conocidas como *algoritmos híbridos*.

Dadas sus características, este tipo de algoritmos ha sido muy estudiado por la comunidad científica en su aplicación a problemas complejos del mundo real. Sin embargo, hasta la fecha, no se ha encontrado un algoritmo que supere a otros algoritmos

en todas las circunstancias [Wolpert97]. Los trabajos relacionados con la temática han tratado de analizar el comportamiento de los algoritmos heurísticos con la finalidad de identificar algoritmos de alto desempeño.

2.2 ANÁLISIS EXPERIMENTAL DEL DESEMPEÑO DE ALGORITMOS

El desempeño de un algoritmo basado en heurísticas es determinado por la eficiencia y la efectividad mostradas por éste al resolver diferentes instancias de un problema. La *efectividad* de un algoritmo se refiere a la calidad de la solución encontrada o a su confiabilidad en la tarea de encontrar soluciones adecuadas y es altamente dependiente de la estructura del problema. La *eficiencia* por otra parte, caracteriza el comportamiento del algoritmo en tiempo de ejecución (por ejemplo el tiempo computacional y los requerimientos de memoria) y está muy relacionada con el conocimiento que se tenga del dominio y la complejidad del problema [Pérez07].

Los criterios para medir el desempeño de algoritmos heurísticos dependen de los métodos elegidos para su caracterización, que pueden ser teóricos o experimentales. En los primeros, para cada algoritmo, se determina matemáticamente la cantidad de recursos necesarios como función del tamaño del caso considerado mejor, peor o promedio. Los segundos se basan en la experimentación para realizar la caracterización y, a diferencia de los métodos teóricos, permiten describir el comportamiento de casos específicos [McGeoch07].

El estudio teórico de la efectividad en los algoritmos heurísticos es muy poco común, pues la aleatoriedad en dichos algoritmos y la complejidad de los problemas que solucionan dificultan un correcto análisis matemático. Asimismo, la aplicabilidad de los resultados teóricos es muy limitada, pues éstos son obtenidos partiendo de condiciones idealizadas que no ocurren en situaciones prácticas [Moret02, Hoos07, McGeoch07]. Dado que, hasta la fecha, no se han encontrado algoritmos exactos que resuelvan en

tiempo polinomial problemas NP-duros, parece ser que la única opción factible es analizar la efectividad de los algoritmos heurísticos de manera experimental.

A pesar de la importancia del análisis de desempeño experimental, éste no ha sido suficientemente explotado en el estudio de algoritmos heurísticos. Una de la razones que dificultan este estudio es que, generalmente, los algoritmos son considerados como cajas negras cuyo funcionamiento interno es desconocido [Watson03, Chiarandini07, Müller-Hannemann10].

Barr [Barr95] y McGeoch [McGeoch02] intentan romper con el paradigma de las cajas negras y sugieren la existencia de tres categorías principales de *factores que afectan el desempeño algorítmico*: problema, algoritmo y ambiente.

- a. *Factores del problema*. Son los factores que definen la instancia del problema a resolver: dimensión, distribución de los parámetros, estructura del espacio de soluciones, entre otros.
- b. *Factores del algoritmo*. Incluyen las estrategias heurísticas utilizadas (procesos de construcción de solución inicial y parámetros de búsqueda asociados), códigos de computadoras empleados, configuración de control interno del algoritmo y comportamiento en la ejecución, entre otros.
- c. *Factores del ambiente*. Se refieren al ambiente físico en el que serán ejecutados los algoritmos como el software (sistema operativo y compilador) y el hardware (velocidad del procesador y memoria). Así como también aquellos relacionados con el programador (por ejemplo, pericia, habilidad de afinación y lógica).

En análisis específicos algunos de estos factores pueden ser controlados para facilitar el estudio. Por ejemplo, el factor del ambiente podría excluirse realizando el experimento en las mismas condiciones ambientales.

2.3 PROCESO DE OPTIMIZACIÓN Y SU CARACTERIZACIÓN

La caracterización del problema y del proceso de solución es una parte esencial en el análisis de desempeño de algoritmos, y permite identificar los factores que influyen en el comportamiento algorítmico. Un análisis de desempeño de calidad requiere la definición de indicadores adecuados que cuantifiquen el impacto de los factores críticos para el desempeño. Para lograr comprender el desempeño de un algoritmo sobre un problema, es necesario realizar un estudio integral de todo el proceso de solución [Smith-Miles09]. El proceso de optimización puede entenderse como la acción de resolver un problema de optimización (entrada) mediante un algoritmo (proceso), obteniendo una solución final (salida). Este proceso es ilustrado en la Figura 1.1.

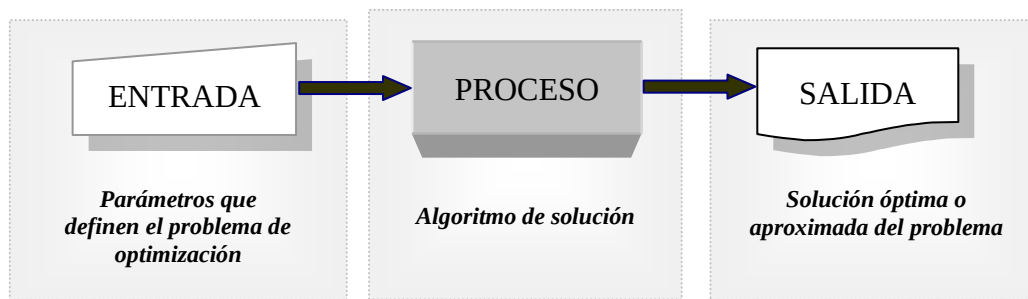


Figura 1.1 Proceso de optimización de un problema.

La *entrada* consiste en una instancia o caso particular del problema de optimización a resolver, compuesta por un conjunto de parámetros específicos que lo definen. El *proceso* incluye el conjunto de estrategias utilizadas para dar solución al problema, es decir, el algoritmo utilizado. La *salida* proporciona la solución óptima (exacta) o aproximada del problema.

Una vez identificados los factores que describen cada etapa del proceso de optimización algorítmico, éstos deben ser analizados para identificar relaciones inherentes entre ellos que permitan explicar el desempeño del algoritmo objeto de estudio. Diferentes trabajos proponen estrategias que pueden ser tomadas en cuenta para facilitar el análisis de estos factores [Barr95, Cohen95, McGeoch02, Lemeire07, Berberich10]. En general, las

estrategias para análisis de datos propuestas en esos trabajos y en trabajos de otras disciplinas [Judd09, Ott10] pueden ser agrupadas en dos categorías: análisis de datos exploratorio y análisis de datos confirmatorio.

- a. *Análisis de datos exploratorio* (Exploratory Data Analysis, EDA). El objetivo de este tipo de análisis es obtener conocimiento a partir del conjunto de datos y de su estructura subyacente. Integra un conjunto de métodos estadísticos, gráficos y de análisis multivariado que ayudan a obtener un modelo que describe las relaciones existentes entre los factores bajo estudio [Tukey77, Hartwig79, Liu96, Johnson00, Gazis10].
- b. *Análisis de datos confirmatorio* (Statistical Hypothesis Testing). Comienza con el establecimiento modelos supuestos acerca de relaciones entre variables del conjunto de datos. Posteriormente, los modelos propuestos son complementados y validados mediante técnicas estadísticas de estimación de parámetros y pruebas de hipótesis [Hartwig79, Cohen95, Liu96, Ott10].

La identificación de técnicas de análisis de datos apropiadas para los tipos de problemas y oportunidades que se presentan al analizar el desempeño algorítmico es un área de investigación aún en desarrollo. La selección de las herramientas a utilizar depende de las características de los datos objeto de estudio (problema, algoritmo y solución). Por ejemplo, los métodos exploratorios gráficos de análisis de datos son los más apropiados para analizar tendencias en grandes conjuntos de datos multivariados. Así mismo, los métodos de análisis de datos confirmatorio pudieran algunas veces ser problemáticos, dependiendo de cuanto conocimiento se tiene del modelo; por consiguiente, los métodos de modelado de datos, con bases exploratorias, que no comienzan asumiendo relaciones funcionales entre los factores estudiados, son los más apropiados cuando el objetivo es descubrir esas relaciones o evaluar varios modelos [McGeoch02, Berberich10].

Para el análisis de desempeño de algoritmos, la literatura propone una gran cantidad de métodos de análisis confirmatorio, sin embargo, profundiza muy poco en las técnicas exploratorias [McGeoch92, Hooker94, Cohen95, Hooker95, Barr95, Brglez07]. Lo anterior puede justificarse por diversas razones. Mucha de la creatividad en la investigación es incluida en la fase exploratoria y la fase confirmatoria es utilizada sólo para validar conocimiento obtenido en la fase exploratoria. Dado que la fase exploratoria envuelve creatividad, la cual es subjetiva y difícil de entender o enseñar, la mayoría de los trabajos concentran su atención en técnicas confirmatorias.

Otra razón para enfocarse en los métodos confirmatorios es que la fase exploratoria de investigación no es comúnmente reportada. Generalmente, los artículos científicos se concentran en la confirmación experimental de resultados, presentando resúmenes exactos y concisos, sin dar detalles de la forma en la que dichos resultados fueron obtenidos. La tendencia general es presentar los resultados adecuándolos de tal forma, que los atajos tomados para facilitar ciertos análisis, así como las técnicas probadas y resultados erróneos que ocurrieron en el proceso, sean removidos [Gent99].

Cada uno de los enfoques mencionados ha contribuido en el proceso de caracterización de problemas y algoritmos de solución. Sin embargo, la complejidad de los problemas y sus estrategias de solución han mostrado que un solo enfoque no es suficiente para entender el proceso de optimización algorítmico y que es necesario conjuntar y profundizar en estas técnicas. Un análisis de desempeño completo debería estar formado por ambos tipos de estudio, exploratorio y confirmatorio. Donde las técnicas exploratorias permitan visualizar, resumir y construir modelos de desempeño algorítmico y las técnicas confirmatorias evalúen los modelos propuestos mediante pruebas de hipótesis estadísticas, mostrando si dicho modelo necesita ser modificado o refinado [Cohen95, Liu96, Feitelson06].

En este trabajo de investigación se espera contribuir al análisis metodológico del desempeño de algoritmos al estudiar e integrar herramientas de análisis exploratorias y

confirmatorias con el fin de obtener mejores explicaciones del comportamiento algorítmico. El enfoque de caracterización propuesto en este trabajo (ver Capítulo 4) consta de tres etapas principales, las cuales permiten explicar el proceso de optimización: la primera consiste en la identificación de los factores que impactan el proceso de optimización, en la segunda, se estudia la aportación de cada uno de los indicadores propuestos y en la tercera se buscan relaciones entre las características.

2.4 EL PROBLEMA DE EMPACADO DE OBJETOS EN CONTENEDORES

El problema de empaado de objetos en contenedores en una dimensión, (*one-dimensional Bin Packing Problem*), de aquí en adelante llamado BPP, consiste en almacenar objetos de diferentes tamaños, o pesos, en el menor número de contenedores de tamaño fijo. Éste es un problema clásico de optimización combinatoria NP-duro, considerado intratable pues demanda una gran cantidad de recursos para su solución [Garey79, Basse98, Álvarez06].

Dado un conjunto $N = \{1, \dots, n\}$ de objetos a distribuir en contenedores del mismo tamaño (capacidad), sea

c = capacidad de cada contenedor

w_i = peso del objeto i , tal que $0 < w_i \leq c$ para $1 \leq i \leq n$

BPP consiste en asignar cada objeto a un contenedor de tal forma que la suma de los pesos de los objetos en cada contenedor no exceda c y el número de contenedores m utilizado sea mínimo [Martello90b]. Es decir, se busca encontrar el menor número de subconjuntos B_j , para $1 \leq j \leq m$, de una partición del conjunto N

$$\bigcup_{j=1}^m B_j = N$$

tal que:

$$\sum_{i \in B_j} w_i \leq c \quad 1 \leq j \leq m, \quad i \in N = \{1, \dots, n\}$$

La complejidad de BPP hace prácticamente imposible el uso de algoritmos exactos para su solución. Para encontrar una solución óptima se deben considerar todas las posibles particiones del conjunto de n objetos en n o menos subconjuntos. Inconvenientemente la complejidad de BPP crece de manera dramática con el tamaño del problema, dado que el número de posibles particiones es mayor que $(n/2)^{n/2}$ [Basse98].

Los algoritmos generalmente utilizados en la solución de BPP son procedimientos metaheurísticos que incluyen estrategias inteligentes. Estas son heurísticas sencillas, utilizadas comúnmente para generar soluciones candidatas, que se diferencian por la manera en que los objetos son tratados antes de ser acomodados y por la forma en que se elige el contenedor que almacenará cada objeto. Las *principales estrategias para BPP* son descritas a continuación:

- *Primer Ajuste* (First Fit, FF) [Johnson74]: Cada objeto considerado es colocado en el primer contenedor que tenga suficiente capacidad disponible. Si ningún contenedor parcialmente lleno puede almacenarlo, el objeto es colocado en un contenedor nuevo (vacío). Una variación a este método se establece cuando los objetos son tomados según el orden decreciente de sus pesos (los objetos son ordenados de mayor a menor peso antes de ser acomodados), dicha variante es conocida como *Primer Ajuste Decreciente* (First Fit Decreasing, FFD).
- *Mejor Ajuste* (Best Fit, BF) [Johnson74]: Cada objeto es acomodado en el contenedor más lleno que lo pueda almacenar, agregando contenedores cuando sea necesario. De igual manera que con FF, existe una variación, llamada *Mejor Ajuste Decreciente* (Best Fit Decreasing, BFD), que considera los objetos en orden decreciente de sus pesos.
- *Peor Ajuste* (Worst Fit, WF) [Johnson74]: Contrario a la estrategia anterior (Best Fit, BF), cada objeto en consideración es almacenado en el contenedor menos lleno con capacidad residual suficiente para contenerlo. La variante que toma los

objetos según el orden decreciente de sus pesos es denominada *Peor Ajuste Decreciente* (Worst Fit Decreasing, WFD).

- *Best 3-Fit Decreasing* (B3FD) [Alvim04]: Busca llenar cada contenedor con tres objetos. Inicialmente se abre un número límite de contenedores. Si existe un contenedor vacío, se selecciona y se coloca el objeto actual, de otro modo, se intenta llenar cada contenedor con objetos que no han sido seleccionados y por pares suman la capacidad residual del contenedor. Para el resto de los objetos, el elemento actual es insertado en el contenedor más lleno en el que ajuste (como en BF). Si no existe contenedor con capacidad suficiente un nuevo contenedor es agregado a la solución.

Las heurísticas descritas en los párrafos anteriores, en general, muestran resultados aceptables. Sin embargo, existen ciertos tipos de instancias de BPP, para las que se ha demostrado que las soluciones obtenidas por estos algoritmos están muy alejadas de las soluciones óptimas [Martello90b, Schwerin97, Wäscher96, Schoenfeld02]. Debido a lo anterior ha sido necesario diseñar procedimientos más elaborados, que integran técnicas variadas, en la búsqueda de buenas soluciones. Sin embargo a la fecha no existe un algoritmo que sea superior a todos los demás ante los diferentes tipos de instancias, por lo que es importante identificar las características que definen el comportamiento de los algoritmos para lograr comprender y mejorar el desempeño de los mismos.

El enfoque de caracterización propuesto en este trabajo (ver Capítulo 4 y Capítulo 5) será aplicado para la creación de nuevas instancias representativas de BPP y el diseño de un algoritmo metaheurístico para BPP de alto desempeño.

Capítulo 3

ESTADO DEL ARTE

Los avances de la última década han mostrado un aumento dramático en el desarrollo de procedimientos para la solución de problemas de optimización en aplicaciones prácticas, destacándose el uso de algoritmos heurísticos, cuya generalidad y alto desempeño los han colocado como el estado del arte en la solución de problemas NP-duros.

En este capítulo se ofrece una síntesis de los principales trabajos relacionados con la caracterización y el estudio del desempeño de algoritmos y se lleva a cabo un análisis comparativo de trabajos que intentan obtener explicaciones de desempeño en algoritmos heurísticos. Posteriormente, se presenta una breve revisión de trabajos relacionados con la caracterización del proceso de optimización de BPP. Se explora el desempeño de los algoritmos heurísticos para BPP que han mostrado resultados sobresalientes, así como las instancias de prueba y los índices de caracterización propuestos.

3.1 CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN

Las implementaciones reales de algoritmos heurísticos generalmente explotan una gran cantidad de ideas y “trucos” relacionados con las características de los problemas que resuelven. Puede argumentarse que un procedimiento algorítmico exitoso es aquél que permite hibridaciones y adaptaciones fácilmente aplicables [Birattari01]. Sin embargo, en la mayoría de los casos, existe poco conocimiento a detalle del porqué las heurísticas muestran cierto desempeño y se carece de bases teóricas sobre cómo caracterizar y relacionar los factores del problema y del algoritmo que conducen a un desempeño exitoso [Watson03, Borenstein05, Chiarandini07, Müller-Hannemann10].

En general, los trabajos relacionados con la caracterización y análisis del proceso de optimización algorítmico, pueden ser agrupados en cuatro enfoques: 1) análisis de datos exploratorio; 2) análisis de la estructura del espacio de búsqueda del problema; 3) estudio de parámetros que ocasionan transiciones de fases en la dificultad del problema; y 4) análisis de la estructura de las soluciones óptimas generadas para las instancias.

3.1.1 ANÁLISIS DE DATOS EXPLORATORIO

El análisis de datos exploratorio utiliza métodos estadísticos y análisis multivariado para obtener conocimiento a partir de un conjunto de índices de caracterización. Existe una amplia cantidad de trabajos que tratan de relacionar *características de algunas clases de instancias* de problemas de optimización específicos con el *desempeño de varios algoritmos* [Leyton-Brown02, Nudelman04, Smyth04, Gagliolo07, Madani09, Messelis09, Smith-Miles09]. Estos trabajos usan indirectamente la definición de clases de instancias para predecir o explicar de manera intuitiva e informal que tipo de algoritmo muestra un mejor desempeño ante ciertos tipos de casos de prueba del problema objeto de estudio, auxiliándose, en algunos casos, de análisis tabulares y gráficos, y en otros, utilizando técnicas de aprendizaje automático (ver Tabla 3.1, Grupo 1, en Sección 3.1.5).

Otros trabajos, en cambio, intentan encontrar una relación entre diversos *parámetros y configuraciones que definen un algoritmo* y el *desempeño final* alcanzado al aplicarse a una clase de instancias de un problema de optimización [Sugihara97, McKay01, Burke02, Thierens02, Hutter06, Halim07, Birattari09]. En algunas de estas investigaciones se aplican procesos de afinación de parámetros basados en diseños experimentales, mientras que en otras se utilizan estrategias de análisis gráficos y de aprendizaje automático para intentar encontrar configuraciones y parámetros que permitan obtener un mejor desempeño (ver Tabla 3.1, Grupo 2).

El éxito del análisis de datos exploratorio depende directamente del alcance y validez del conjunto de índices utilizados para caracterizar el proceso de optimización algorítmico. Las índices de caracterización objeto de análisis deben ser cuidadosamente elegidos, de tal manera que sean capaces de caracterizar correctamente el proceso de optimización y explicar el desempeño algorítmico.

3.1.2 ANÁLISIS DE LA ESTRUCTURA DEL ESPACIO DE BÚSQUEDA

La *superficie de aptitudes* es considerada como la visualización del espacio de búsqueda, definido a través de las aptitudes de los individuos, imaginándolo como una superficie formada por picos, valles, montañas y planicies [Wright32]. El análisis de la estructura del espacio de búsqueda del problema (landscape) ha permitido estudiar algunos aspectos de las características del espacio de soluciones transitado por los algoritmos heurísticos. Este tipo de análisis ayuda a aumentar el conocimiento que se tiene sobre el problema y algoritmo estudiados, al sugerir explicaciones e ideas que proporcionen descripciones intuitivas de los aspectos importantes del proceso de búsqueda [Jones95].

Algunos autores han intentado caracterizar la *estructura del espacio de búsqueda* trazado por un algoritmo al solucionar una clase de instancias del problema, y asociar las propiedades de éste con el *desempeño final obtenido* [Mitchell92, Jones95, Corne03, Mitchell03, Merz04, Halim07, Ochoa09] (ver Tabla 3.1, Grupo 3). Otros investigadores,

han explorado la estructura del espacio de soluciones de problemas de optimización, recorriendo la superficie de aptitudes del problema de manera aleatoria, intentando encontrar medidas más generales de dificultad [Smyth04, Czogalla09] (ver Tabla 3.1, Grupo 4).

Sin embargo, el análisis de las medidas propuestas ha demostrado que, actualmente, no existen indicadores que sean capaces de capturar, de manera precisa, la dificultad de la búsqueda. Hasta ahora, no ha sido posible relacionar, de manera exitosa, las propiedades de la estructura del espacio de búsqueda, con el comportamiento algorítmico y el desempeño final [Merz00a, Merz00b, Corne03, Merz04, Borenstein05].

3.1.3 ESTUDIO DE FENÓMENOS DE TRANSICIÓN DE FASE

Una *transición de fase* puede ser definida como un cambio abrupto en el comportamiento cualitativo de un sistema (problema), por ejemplo, la transformación del estado del agua de líquido a gaseosos o el proceso mediante el que un sistema que pasa de un estado de desorden a un estado de orden [Borgs01, Leyton-Brown02]. En el caso de los problemas de optimización NP-duros, una transición de fase puede ser definida como un cambio abrupto en la complejidad del problema que ocurre cuando un parámetro apropiadamente definido es variado. Una de las razones del interés en tales transiciones de fases es que los problemas más difíciles tienden a ocurrir en la región de transición.

Un gran número de resultados experimentales demuestran que para algunos tipos de problemas existe una propiedad de transición de fase alrededor de la cuál ocurren los problemas más difíciles [Cheeseman91, Martin01, Achlioptas05, Mertens06, Rangel09, Zeng09]. La identificación de estas propiedades ha permitido caracterizar la estructura y complejidad de dichos problemas de optimización. El conocimiento obtenido en el estudio de estos fenómenos de *transición de fase* ha sido aplicado para identificar instancias de prueba con alto *grado de dificultad* y para sugerir nuevos algoritmos

diseñados especialmente para las instancias difíciles que se presentan en la zona de transición [Cook91, Béjar01, Caramanis03, Piñol08, Rangel09, Zeng09].

Sin embargo, existen un gran número de problemas de optimización, para los cuales no ha sido posible identificar la existencia del fenómeno transición de fases. Por otro lado, no parece haberse realizado un estudio sobre la relación entre los parámetros que definen la transición de fases en la complejidad del problema y el comportamiento de los algoritmos de solución (ver Tabla 3.1, Grupo 5, en Sección 3.1.5).

3.1.4 ANÁLISIS DE LA ESTRUCTURA DE LAS SOLUCIONES ÓPTIMAS

El estudio de la *estructura de las soluciones óptimas* de problemas NP-duros ha permitido identificar relaciones entre las características de los valores que definen las soluciones y la dificultad de las instancias. En el proceso de solución de problemas de optimización es posible identificar instancias que incluyen conjuntos de variables que tienen valores fijos en todas las soluciones óptimas, este tipo de conjunto de variables es denominado “columna vertebral” (backbone).

Algunos trabajos se han concentrado en el estudio de la *estructura de las soluciones óptimas* de los problemas NP-duros, intentando identificar propiedades que puedan ser asociadas con la *dificultad de las instancias*. Para algunos problemas de optimización ha sido posible identificar conjuntos de variables que presentan valores fijos en todas las soluciones óptimas, y en varios casos se ha mostrado que la magnitud de estos conjuntos está relacionada con el grado de dificultad de las instancias [Monasson99, Slaney01, Zeng09].

Sin embargo, a la fecha, son pocos los estudios que han cuantificado directamente la relación entre el tamaño de dichos conjuntos y la dificultad de las instancias [Singer00, Slaney01, Watson03, Zeng09] (ver Tabla 3.1, Grupo 6, en Sección 3.1.5). Así mismo, son limitados los trabajos que han utilizado directamente esta propiedad para mejorar el

desempeño de algoritmos de solución [Schneider03, Zhang04, Zhang05, Zeng09] (ver Tabla 3.1, Grupo 7, en Sección 3.1.5).

3.1.5 ANÁLISIS DEL PROCESO DE OPTIMIZACIÓN

Cada uno de los enfoques anteriores, ha contribuido a la caracterización del proceso de optimización. Sin embargo, la complejidad de los problemas NP-duros y los algoritmos heurísticos ha mostrado que un único enfoque de caracterización no es suficiente para comprender el desempeño de los algoritmos y la dificultad de las instancias, destacando la necesidad de un análisis más completo. La Tabla 3.1 resume las características de los estudios presentados en los principales trabajos sobre análisis de desempeño algorítmico, donde se destacan las etapas del proceso de optimización algorítmica que han sido analizadas por los diferentes autores. Los trabajos han sido integrados en siete grupos, tomando en cuenta el tipo de investigación realizado.

Tabla 3.1 Trabajos relacionados con el análisis de desempeño de algoritmos.

Trabajos	Problema		Algoritmo			Desempeño
	Instancias	Soluciones óptimas	Parámetros	Trayectoria de búsqueda	Estructura	
Grupo 1	✓					✓
Grupo 2			✓			✓
Grupo 3				✓		✓
Grupo 4	✓			✓		✓
Grupo 5	✓*					✓
Grupo 6		✓				✓
Grupo 7		✓		✓		✓
Este trabajo	✓	✓	✓	✓	✓	✓
Grupo 1	[Leyton-Brown02, Nudelman04, Smyth04, Gagliolo07, Madani09, Messelis09, Smith-Miles09]					
Grupo 2	[Sugihara97, McKay01, Burke02, Thierens02, Hutter06, Halim07, Birattari09]					
Grupo 3	[Mitchell92, Jones95, Corne03, Mitchell03, Merz04, Halim07, Ochoa09]					
Grupo 4	[Smyth04, Czogalla09]					
Grupo 5	[Cheeseman91, Martin01, Achlioptas05, Mertens06, Rangel09] *Identificación de transición de fase					
Grupo 6	[Singer00, Slaney01, Watson03]					
Grupo 7	[Schneider03, Zhang04, Zhang05]					

El análisis de los trabajos del estado del arte enfocados a la caracterización de problemas de optimización y algoritmos heurísticos reveló que, hasta ahora, no ha sido posible establecer una conexión entre las características que han sido definidas para caracterizar

las tres etapas del proceso de optimización (entrada - proceso - salida). La mayor parte de las investigaciones se han concentrado en demostrar que las heurísticas propuestas son efectivas o superiores a otras, para solucionar algún problema de optimización, pero no se ha demostrado el porqué de su buen o mal desempeño. Diversos autores han coincidido en afirmar que actualmente aún existe una fuerte falta de conocimiento sobre la forma exacta en que el desempeño de diferentes heurísticas se relaciona con las características de las instancias de un problema, destacando la necesidad de una guía para la descripción y análisis de desempeño de implementaciones reales de estas estrategias [Birattari01, Smyth04, Hoos04, Stützle04, Borenstein05, Chevalier06, Chiarandini07, Smith-Miles09].

Así mismo, gran número de los problemas NP-duros y heurísticas de solución no han sido estudiados aún y se tiene poco conocimiento sobre la dificultad de las instancias de estos problemas y el comportamiento de los algoritmos [Merz04]. Los últimos trabajos realizados en este campo han señalado la necesidad de conjuntar todas las técnicas de caracterización propuestas para obtener mejores explicaciones del desempeño de los algoritmos heurísticos y la dificultad de las instancias que solucionan [Smith-Miles09]. Un estudio de este tipo es importante, ya que puede proporcionar bases sólidas para el análisis y diseño de algoritmos.

3.2 EL PROCESO DE OPTIMIZACIÓN DE BPP

En esta sección se presentan trabajos relevantes relacionados con la caracterización del proceso de optimización de BPP. En la búsqueda de mejores soluciones para este problema, se han diseñado algoritmos elaborados que integran técnicas heurísticas. Así mismo, se han propuesto bancos de instancias de prueba e índices de caracterización, para evaluar el desempeño de los algoritmos y estudiar la dificultad de las instancias.

3.2.1 ALGORITMOS HEURÍSTICOS PARA SOLUCIÓN DE BPP

A lo largo de los últimos veinte años, se han propuesto procedimientos variados para la solución de BPP, los resultados más destacados han sido obtenidos mediante el uso de algoritmos heurísticos. Martello y Toth [Martello90b] proponen un procedimiento de branch-and-bound (MTP). Falkenauer [Falkenauer96] propone un algoritmo genético híbrido de agrupación (HGGA) que utiliza un esquema de representación por grupos y un método de optimización local inspirado en el criterio de dominación propuesto por Martello y Toth [Martello90a].

Scholl et al. [Scholl97] desarrollan un método híbrido (BISON) que combina búsqueda tabú con un procedimiento de branch-and-bound basado en varios límites y un nuevo esquema de ramificación. Coffman et al. [Coffman97] desarrollan un estudio exhaustivo de los principales métodos heurísticos deterministas para BPP, incluyendo análisis del peor caso de los algoritmos primer ajuste decreciente y mejor ajuste decreciente. Schwerin y Wäscher [Schwerin99] proponen un nuevo límite inferior para BPP basado en el problema de corte y lo integran al procedimiento MTP, obteniendo resultados de alta calidad con su nuevo procedimiento (MTPCS).

Fleszar y Hindi [Fleszar02] desarrollan un algoritmo (Perturbation-MBS') que incorpora una versión modificada de la heurística MBS de Gupta y Ho [Gupta99] y una búsqueda de vecindad variable, así como límites inferiores, obteniendo buenos resultados. Levine y Ducatelle [Levine04] desarrollan un algoritmo híbrido (HACO-BP) que implementa la metaheurística optimización basada en colonia de hormigas (ACO) incluyendo una estrategia de búsqueda local, apoyada en el criterio de dominación de Martello y Toth [Martello90a]. Bhatia y Basu [Bhatia04] presentan un algoritmo genético de agrupación multi-cromosómico (MGGA), así como una nueva heurística de empaçado por reemplazo (better-fit).

Alvim et al. [Alvim04] desarrollan una heurística de mejora híbrida (HI_BP), en la que retoman la estrategia dual utilizada por Scholl et al. [Scholl97], así como las técnicas de reducción de Martello y Toth [Martello90a], obteniendo los mejores resultados

reportados hasta ese momento en el estado del arte. Singh y Gupta [Singh07] proponen un enfoque evolutivo híbrido (C_BP), que integra un algoritmo genético de agrupación (H-SGGA) y una versión mejorada de la heurística Perturbation-MBS' de Fleszar y Hindi [Fleszar02], tomando en cuenta el pesos promedio de los objetos para seleccionar una u otra heurística. Aunque el desempeño obtenido es comparable con la estrategia HI_BP de Alvim et al. [Alvim04], el procedimiento C_BP presenta un comportamiento menos robusto en términos del número de veces que logra encontrar las soluciones.

Recientemente, Stawowy [Stawowy08] propone el uso de una estrategia evolutiva (ES), un algoritmo no especializado, que incluye mutaciones inteligentes y una técnica de reducción del tamaño del problema, donde las soluciones son manipuladas mediante un esquema de representación de permutaciones de objetos con separadores de grupos, los resultados obtenidos por el metaheurístico no hibridizado son comparables con los de otros algoritmos mucho más complicados.

Rohlfshagen y Bullinaria [Rohlfshagen10] desarrollan un algoritmo genético inspirado en el proceso de formación de proteínas (ESGA) que obtiene resultados prometedores al compararse con otras estrategias del estado del arte. Loh et al. [Loh08] desarrollan un procedimiento (WA) que hace uso del concepto de recocido de pesos, el algoritmo propuesto es sencillo y fácil de implementar y obtiene un desempeño de alta calidad, superando las soluciones obtenidas por el mejor algoritmo del estado del arte HI_BP [Alvim04].

Gómez-Meneses y Randall [Gómez-Meneses09] presentan un enfoque evolutivo híbrido de optimización extrema (HEO) que incorpora una búsqueda local para mejorar el tamaño de los contenedores. Lewis [Lewis09] propone un algoritmo de escalado de colina (HC) que utiliza un esquema sencillo de mejora basado en el criterio de dominación y obtiene buenas soluciones, superando para algunos grupos de instancias el desempeño de algoritmos complejos, sin rebasar a los mejores del estado del arte.

El último trabajo en esta área fue presentado por Fleszar y Charalambous [Fleszar11], los autores proponen una modificación al algoritmo Perturbation-MBS' de Fleszar y Hindi [Fleszar02] que utiliza un método para controlar el peso promedio de los objetos que son empacados en cada contenedor (Perturbation-SAWMBS). La nueva heurística supera el desempeño reportado para los mejores algoritmos del estado del arte HI_BP, C_BP y WA [Alvim04, Singh07, Loh08], así mismo los autores publican nuevos resultados para la heurística WA, que contradicen el desempeño presentado inicialmente por sus autores.

La Tabla 3.2 muestra las características y técnicas utilizadas por los trabajos más destacados en el estado del arte de BPP. Las estrategias mayormente explotadas, y que han permitido obtener los mejores resultados, consisten principalmente en:

1. El uso de heurísticas simples de BPP para generar soluciones iniciales de calidad.
2. La aplicación de técnicas de reducción para disminuir el tamaño del problema.
3. La inclusión de técnicas de búsqueda local basadas en el criterio de dominación.
4. El uso de límites inferiores del tamaño de la solución como criterio de paro.

Tabla 3.2 Técnicas destacadas en los mejores algoritmos del estado de arte de BPP.

Trabajo	Estrategia	Heurísticas para BPP	Técnicas de reducción	Criterio de dominación	Límites	Corto tiempo de ejecución
[Falkenauer96]	Algoritmo Genético	✓		✓		
[Levine04]	Colonia de Hormigas	✓		✓		
[Alvim04]	Heurística Híbrida	✓	✓		✓	✓
[Singh07]	Algoritmo Genético	✓			✓	✓
[Stawowy08]	Algoritmo Genético	✓	✓			
[Loh08]	Búsqueda Local	✓				✓
[Fleszar11]	Búsqueda Local	✓	✓			✓
Este trabajo	Algoritmo Genético	✓	✓		✓	✓

3.2.2 INSTANCIAS DE PRUEBA

El desempeño de los algoritmos mencionados en el apartado anterior ha sido evaluado con diferentes clases de instancias de referencia consideradas retadoras. Los casos de prueba, elegidos por los diferentes investigadores para comparar las cualidades de sus estrategias con las de otros algoritmos del estado del arte, conforman un conjunto de 1615 casos estándar reconocidos por la comunidad científica. Dichas instancias de prueba pueden ser encontradas en sitios de Internet reconocidos [Beasley90, Klein09, ESICUP, CaPaD]. El valor de la solución óptima para cada una de estas instancias es conocido. Las instancias consideradas se han dividido en cuatro grupos, tomando en cuenta su origen y los sitios de los que fueron obtenidas. El primer grupo consiste en dos conjuntos de instancias de prueba propuestas por Falkenauer [Falkenauer96].

1. **Uniform** [Beasley90]: Conjunto de 80 instancias identificadas con la letra **u** debido a que su principal característica es que los pesos de los objetos están *uniformemente* distribuidos entre 20 y 100. La capacidad del contenedor c es de 150 y existen cuatro clases de casos cada uno con $n = 120, 250, 500$ y 1000 objetos. Cada clase posee 20 instancias identificadas respectivamente por u_{120} , u_{250} , u_{500} y u_{1000} .
2. **Triplets** [Beasley90]: Conjunto de 80 instancias difíciles, identificadas con la letra **t**. Su nombre se debe a que las instancias fueron construidas con una solución óptima conocida de $n/3$ contenedores, de tal forma que cada contenedor de la solución óptima debe almacenar exactamente *tres objetos* que lo llenan completamente. El tamaño de las instancias es de $n = 60, 120, 249$ y 501 , definiéndose así cuatro clases, el tamaño del contenedor c es de 100, mientras que los pesos están distribuidos entre 25 y 50. Cada clase posee 20 instancias identificadas respectivamente por t_{60} , t_{120} , t_{249} y t_{501} .

El segundo grupo está formado por tres conjuntos de casos de prueba introducidos por Scholl et al. [Scholl97]. En cada conjunto, las diferentes clases de problemas fueron

creados variándose el número de objetos n , la capacidad del contenedor c y los posibles pesos de los objetos.

1. **Data Set 1** [Klein09]: Construidas de forma similar que algunas instancias propuestas por Martello y Toth [Martello90a] que resultaron difíciles. Son un conjunto de 720 instancias denotadas con **n - c - w** por los datos que maneja: $n = 50, 100, 200$ y 500 , capacidad del contenedor $c = 100, 120, 150$ y pesos w generados *uniformemente* en intervalos de $[1,100]$, $[20,100]$ y $[30,100]$. La combinación de los diferentes parámetros resulta en 36 clases, cada clase contiene 20 instancias.
2. **Data Set 2** [Klein09]: Conjunto formado por 480 instancias con pesos generados con una distribución *uniforme*, denotadas por **n - w - b - r** . Cada sigla representa la configuración de un parámetro de entrada: número de objetos n con valores de $50, 100, 200$ y 500 , la capacidad de los contenedores c es 1000 . Con el objetivo de generar instancias cuyo número medio de objetos por caja variara entre tres y nueve, se consideraron otros dos parámetros: $\bar{w}$ que representa el peso medio deseado $\bar{w} = c/3, c/5, c/7, c/9$, y una desviación $\delta = 20\%, 50\%$ y 90% que determina la variación máxima de un peso dado w en relación a $\bar{w}$. Por ejemplo cuando $\bar{w} = C/5$ y $\delta = 50\%$ los pesos de los objetos fueron generados de manera aleatoria con una distribución uniforme en el intervalo discreto $[100, 300]$. Combinando las características anteriores se cuenta con 48 clases, cada una con 10 instancias.
3. **Data Set 3** [Klein09]: Conjunto formado por una única clase con diez instancias consideradas difíciles, mejor conocidas como ***hard***. Cada instancia posee $n = 200$ objetos, capacidad de contenedor $c = 100000$ y pesos distribuidos *uniformemente* entre 20000 y 35000 . Los pesos de los objetos se encuentran ampliamente dispersos y el número de elementos por contenedor cae entre tres y cinco.

El tercer grupo incluye tres clases de instancias sugeridas por Schwerin, Wäscher y Gau. Las primeras dos clases was 1 y was 2 se han definido como *ffd-duras* (difíciles de resolver por la heurística FFD) [Schwerin97, Schwerin99]. La tercera clase también ha sido considerada difícil [Wäscher96].

1. **Was 1** [ESICUP]: Formado por 100 instancias de capacidad $c = 1000$. Cada instancia posee $n = 100$ objetos. Los pesos de los objetos varían entre 150 y 200.
2. **Was 2** [ESICUP]: 100 instancias con una capacidad $c = 1000$, cada una con 120 objetos con pesos entre 150 y 200.
3. **Gau 1** [ESICUP]: Conjunto de 17 instancias de características variadas. La capacidad de los contenedores es de 10000, el número de objetos n varía de 57 a 239 mientras que los pesos se encuentran distribuidos entre 2 y 7332.

El grupo cuatro contiene un conjunto de 28 instancias difíciles que también han sido consideradas en problemas de corte de una dimensión (one-dimensional Cutting Stock Problem). Los resultados óptimos son conocidos para todas las instancias [Belov04].

1. **Hard28** [CaPaD]: Son las 28 instancias más difíciles propuestas por Schoenfield [Schoenfield02], dichas instancias no pudieron ser resueltas por algoritmos de corte ni por métodos de reducción. La capacidad del contenedor c es 1000 y los pesos de los objetos están uniformemente distribuidos entre 1 y 800. El número de objetos n varía entre 160 y 200.

Los últimos dos conjuntos de instancias, Gau 1 y Hard28, han mostrado poseer casos de prueba con un alto grado de dificultad. Destacándose el conjunto Hard28, para el que existe un mayor número de instancias que los algoritmos no logran solucionar de manera óptima.

3.2.3 RESULTADOS EXPERIMENTALES DE ALGORITMOS

Las tablas mostradas a continuación, detallan el desempeño alcanzado por los mejores algoritmos para BPP. Los resultados de HI_BP fueron obtenidos al ejecutar el código en C proporcionados por la Dra. Adriana Alvim. Los resultados de los procedimientos WA y Perturbation-SAWMBS fueron obtenidos de la publicación de Fleszar y Charalambous [Fleszar11], donde se destaca que los resultados anunciados por Loh et al. [Loh08] para la heurística WA son incorrectos. Para verificar los resultados de la heurística WA se realizó la codificación de dicho algoritmo, obteniendo resultados similares a los reportados por Fleszar y Charalambous, por lo que se coincide con dichos autores en que los resultados publicados por Loh et al. son incorrectos. Para los otros algoritmos, los resultados fueron obtenidos de sus respectivas publicaciones. La calidad de las soluciones es comparada, tomando en cuenta el número de instancias para las que los algoritmos alcanzan la solución óptima, así como el tiempo promedio de ejecución medido en segundos. En todos los casos, cuando los algoritmos no logran encontrar la solución óptima, las soluciones encontradas poseen un contenedor extra, respecto a la solución óptima.

La Tabla 3.3 muestra los resultados obtenidos por las estrategias heurísticas del estado del arte que han alcanzado un desempeño sobresaliente: HI_BP [Alvim04], C_BP [Singh07], WA [Loh08] y Perturbation-SAWMBS [Fleszar11], siendo el procedimiento Perturbation-SAWMBS el que ha mostrado el mejor desempeño. La Tabla 3.4 presenta los resultados obtenidos por los mejores algoritmos metaheurísticos reportados en la literatura: HGGA [Falkenauer96], HACO-BP [Levine04], H-SGGA [Singh07], donde destaca el desempeño del procedimiento HGGA. En la Tabla 3.3 se han resaltado los resultados obtenidos para el conjunto de instancias Hard28, que contiene las instancias más difíciles, para los algoritmos conocidos; identificar las características que definen este conjunto de instancias podría ayudar a comprender su dificultad y proponer mejores estrategias para solucionarlas.

Tabla 3.3 Resultados obtenidos por algoritmos heurísticos aplicados a BPP (tiempo en seg.).

Instancias	No. Instancias	HI_BP		C_BP		WA		Pert.-SAWMBS	
		Óptimos	Tiempo	Óptimos	Tiempo	Óptimos	Tiempo	Óptimos	Tiempo
U	80	80	0.03	79	0.18	71	0.28	79	0.00
T	80	80	0.98	80	0.01	0	0.15	80	0.00
Data Set 1	720	720	0.19	719	0.31	703	0.25	720	0.01
Data Set 2	480	480	0.01	480	0.04	468	0.04	480	0.00
Data Set 3	10	10	4.60	10	0.92	9	0.08	10	0.16
Was 1	100	100	0.02	100	0.00	100	0.00	100	0.00
Was 2	100	100	0.02	100	0.00	100	0.02	100	0.01
Gau 1	17	12	0.60	15	0.43	13	0.02	16	0.04
Hard28	28	5	0.48	--	--	5	0.59	5	0.24
Total	1615	1587	0.77	1583	0.24	1469	0.16	1590	0.05
		P4 1.7 GHz		P4 2.4 GHz		Core2 2.33 GHz		Core2 2.33 GHz	

Tabla 3.4 Resultados obtenidos por algoritmos metaheurísticos aplicados a BPP (tiempo en seg.).

Instancias	No. Instancias	HGGA		HACO-BP		H-SGGA	
		Óptimos	Tiempo	Óptimos	Tiempo	Óptimos	Tiempo
U	80	76	2448	78	62.50	78	0.45
T	80	78	629.25	--	--	9	3.24
Data Set 1	720	--	--	--	--	717	0.34
Data Set 2	480	--	--	--	--	479	0.05
Data Set 3	10	--	--	--	--	10	0.42
Total	1370	154	1538.6	78	62.50	1293	0.9
HGGA	R4000 Silicon Graphics workstation						
HACO-BP	Sun Microsystems Blade 100 machines, 502 MHz UltraSPARC-IIe processors						
H-SGGA	2.4 GHz Pentium IV						

La revisión de los resultados obtenidos por los mejores algoritmos de solución de BPP reveló que aún existen instancias de la literatura que presentan un alto grado de dificultad y para las que las estrategias incluidas en los procedimientos no parecen conducir a mejores soluciones. Al realizar el análisis de la literatura, se observó que ninguna de las estrategias del estado del arte ha sido analizada para explicar el porqué de su buen o mal desempeño. Por otro lado, tampoco existe un análisis general sobre la estructura y complejidad del problema de empaqueo de objetos en contenedores. Es importante identificar cuáles son las características críticas de las instancias de BPP que pueden ser causa de su grado de dificultad. Así mismo, es necesario comprender el comportamiento de los algoritmos y evaluar las estrategias que les permiten alcanzar su

desempeño. Los primeros esfuerzos por contribuir a la caracterización al proceso de optimización de BPP son descritos en la siguiente sección.

3.2.4 INDICES DE CARACTERIZACIÓN DE BPP

Es sabido que, factores como el número de objetos, la tendencia central de sus pesos y su distribución, afectan el grado de dificultad de una instancia de BPP. De esta forma, diferentes autores han propuesto conjuntos de índices para la caracterización de BPP. Para modelar la estructura de una instancia de BPP, Cruz [Cruz04] formuló un conjunto de índices de dificultad, descritos en la Tabla 3.5. Por otro lado, en el trabajo de Álvarez [Álvarez06] se propone un conjunto de índices propósito general, basados en estadística descriptiva. La Tabla 3.6 muestra las funciones propuestas por el autor, los índices son agrupados en cuatro categorías.

Tabla 3.5 Indicadores propuestos en el trabajo de Cruz [Cruz04].

Ecuación	Descripción	Identificador de Ecuación
$p = \frac{n}{nmax}$	p es el índice del tamaño del caso, donde n = número de objetos, $nmax$ = tamaño máximo solucionado.	(3.1)
$t = \frac{\sum_{i=1}^n w_i / n}{c}$	t es el índice de capacidad ocupada por un objeto promedio, donde w_i = peso del objeto i , c = capacidad del contenedor.	(3.2)
$d = \sqrt{\frac{\sum_{i=1}^n \left[t - \left(\frac{w_i}{c} \right) \right]^2}{n}}$	El índice de dispersión d expresa el grado de la dispersión del cociente del peso de los objetos entre el tamaño del contenedor.	(3.3)
$f = \frac{\sum_{i=1}^n factor(c, w_i)}{n}$	El índice de factores f expresa la proporción de objetos cuyo peso w_i es factor de la capacidad del contenedor, donde $factor(c, w_i) = \begin{cases} 1 & \text{si } (c \bmod w_i) = 0 \\ 0 & \text{en caso contrario} \end{cases}$	(3.4)
$b = \begin{cases} 1 & \text{si } c \geq \sum_{i=1}^n w_i \\ \frac{c}{\sum_{i=1}^n w_i} & \text{en caso contrario} \end{cases}$	El uso de contenedor b expresa la proporción del peso total que se puede asignar en un contenedor de capacidad c .	(3.5)

Los índices propuestos por Cruz y Álvarez, fueron retomados en los trabajos de Pérez y Landero [Pérez07, Landero08]. Los autores proponen, medir aspectos en cada etapa del proceso de optimización para analizar el desempeño de estrategias heurísticas aplicadas en la solución de BPP. Cada autor desarrolla índices de desempeño para caracterizar el comportamiento de diferentes variantes del algoritmo de interés. La Figura 3.1 muestra los factores que se recomienda medir para cada fase.

Tabla 3.6 Indicadores propuestos en el trabajo de Álvarez [Álvarez06].

Medidas	Estadístico
Centralización: indican valores con respecto a los que se agrupan los pesos de los objetos.	Media aritmética
	Media geométrica
	Media armónica
	Mediana
	Moda
Dispersión: muestran el grado de concentración de los pesos con respecto a las medidas centralizadas.	Rango
	Desviación media
	Varianza
	Desviación estándar
	Cuasivarianza
	Cuasidesviación estándar
	Coeficiente de variación
Posición: dividen el conjunto de pesos en grupos del mismo tamaño.	Error estándar
	Cuartiles
	Deciles
Forma: buscan cuantificar algunos aspectos formales de la distribución de frecuencias de los pesos.	Percentiles
	Coeficiente de asimetría de Pearson
	Coeficiente de asimetría de Pearson mediana
	Coeficiente de asimetría de Pearson moda
	Coeficiente de asimetría de Bowley
	Curtosis

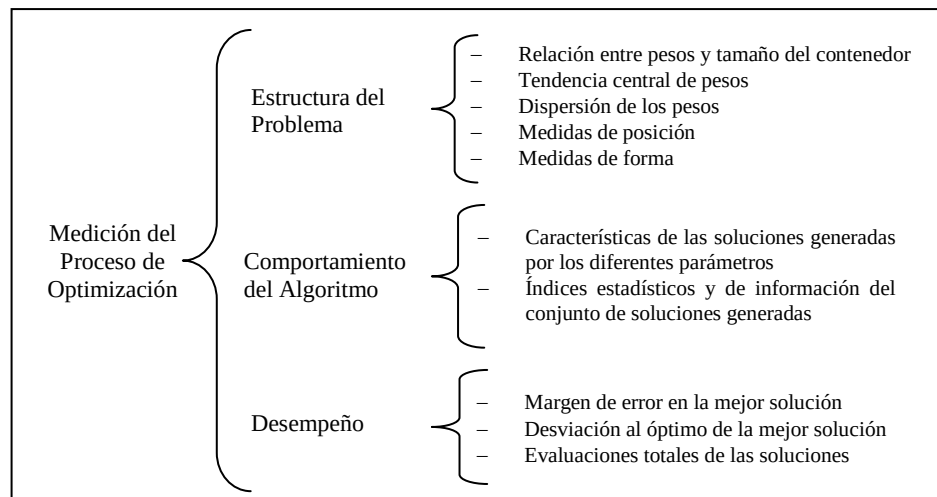


Figura 3.1 Caracterización propuesta por Pérez y Landero [Pérez07, Landero08].

Los trabajos descritos en los párrafos anteriores representan los primeros esfuerzos en la búsqueda de explicaciones del desempeño de algoritmos de solución de BPP. Sin embargo, los índices propuestos para la caracterización de las instancias de BPP no son suficientes para explicar la dificultad de algunas instancias, como las Hard28. Por otro lado, los algoritmos de BPP considerados en dichos trabajos son heurísticas elementales, cuyo desempeño y complejidad está muy lejos del observado por los procedimientos del estado del arte de BPP.

La caracterización del desempeño de los algoritmos heurísticos es un problema desafiante, y aún existe un largo camino por recorrer antes de lograr comprender en detalle el grado de dificultad de las diferentes instancias de los problemas de optimización y los algoritmos que los solucionan. Se espera que los conocimientos obtenidos, producto de este trabajo de investigación, permitan desarrollar una nueva guía de análisis de desempeño que integre técnicas de caracterización y análisis que permitan tener un conocimiento detallado sobre el proceso de optimización algorítmico.

Capítulo 4

CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN: UN NUEVO ENFOQUE DE ANÁLISIS EXPERIMENTAL

En el Capítulo 2 de esta tesis se definió el proceso de optimización algorítmico y se describieron las herramientas de análisis de datos, propuestas en la literatura, para el estudio de los factores que afectan dicho proceso. Así mismo, se hizo hincapié en la necesidad de utilizar métodos de análisis exploratorio para la obtención de modelos de explicación del proceso de optimización. Por otro lado, en el Capítulo 3 se revisaron los principales avances sobre la caracterización del proceso de optimización, destacando la necesidad de una guía que: facilite el análisis de desempeño de heurísticas aplicadas a problemas NP-duros y permita comprender la dificultad de las instancias y explicar el comportamiento de los algoritmos.

En este capítulo se presenta un enfoque experimental para la caracterización y análisis del proceso de optimización algorítmico y su aplicación en el estudio y rediseño de un algoritmo genético híbrido para BPP. El enfoque propuesto combina técnicas de análisis de datos con el objetivo de explicar el proceso de optimización.

4.1 ENFOQUE EXPERIMENTAL PARA CARACTERIZAR Y EXPLICAR EL PROCESO DE OPTIMIZACIÓN ALGORÍTMICO

En la Figura 4.1 se presenta el enfoque propuesto para la caracterización y análisis del comportamiento de un algoritmo heurístico que resuelven un problema de optimización.

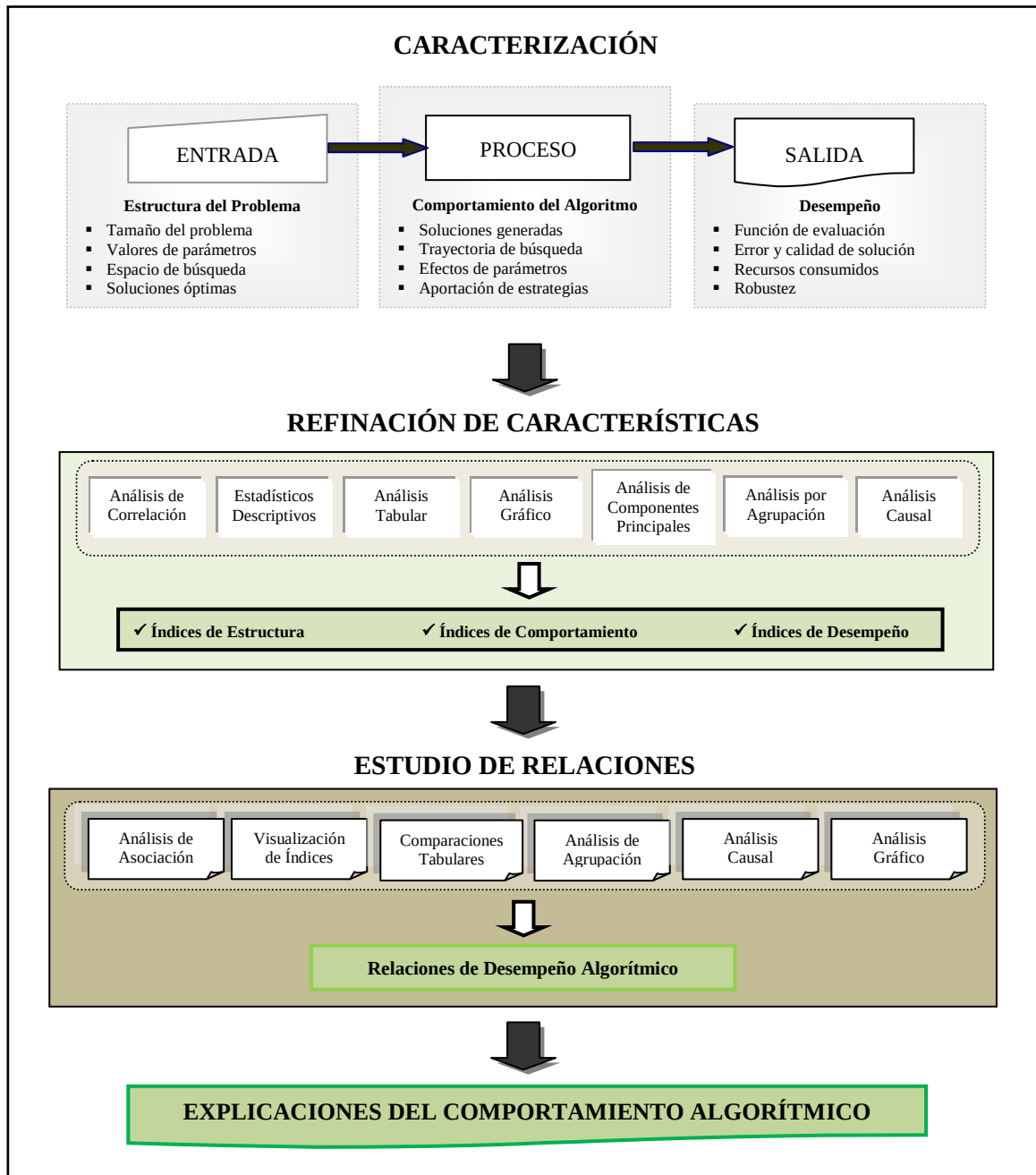


Figura 4.1 Enfoque experimental para estudiar el proceso de optimización algorítmico.

El **enfoque** para el *estudio integral del proceso de optimización*, mostrado en la Figura 4.1, combina métodos de análisis exploratorio de datos en tres etapas, con la finalidad de identificar relaciones entre los factores que afectan el proceso de optimización. Primero, en la etapa de *caracterización* se identifican factores que influyen en el desempeño y se cuantifican mediante índices. Después, en la etapa de *refinación de características* se descartan índices incorrectos, redundantes o irrelevantes y se generan nuevas variables. Finalmente, en la etapa *estudio de relaciones* se realiza un análisis de las características del proceso de optimización con el propósito de obtener relaciones de desempeño que se convierten en *explicaciones del comportamiento algorítmico*.

4.1.1 ETAPA 1: CARACTERIZACIÓN

El objetivo principal de la etapa de *caracterización* es identificar, en cada una de las fases del proceso de optimización, factores relevantes para describir la estructura de las instancias, el comportamiento del algoritmo, las soluciones generadas y el desempeño final. Después de identificar los elementos que intervienen en cada fase del proceso de optimización, es necesario analizar qué aspectos es posible medir en cada categoría. Es necesario estudiar a detalle el problema de optimización y el procedimiento algorítmico, con el objetivo de seleccionar índices de caracterización que proporcionen información útil, que permita describir el proceso de optimización y observar tendencias importantes. La Figura 4.1, en la etapa de caracterización, muestra algunos de los factores críticos que afectan el proceso de optimización [Barr95, McGeoch02, Pérez07].

En la *entrada*, los índices de caracterización elegidos deben proporcionar información acerca de las instancias del problema, su estructura y dificultad, entre los factores que es importante describir en cada instancia se encuentran: su tamaño, los parámetros que la definen, la estructura del espacio de búsqueda, la estructura de las soluciones óptimas. En el *proceso*, los índices elegidos deben caracterizar del comportamiento del algoritmo en ejecución; los algoritmos heurísticos, al ser multipropósito, tienen muchas variantes que se ajustan al problema que resuelven; por esta razón, la elección de los índices de

caracterización se encuentra ligada directamente a la estrategia utilizada. Algunas de las características que es importante medir son: efectos de los parámetros, trayectorias de búsqueda, calidad de las soluciones, desempeño y aportación de las estrategias incluidas. En la *salida*, se buscan indicadores importantes del desempeño del algoritmo, como son: la calidad de las soluciones finales, el tiempo de ejecución, el número de iteraciones y la robustez.

4.1.2 ETAPA 2: REFINACIÓN DE CARACTERÍSTICAS

El objetivo de la etapa de *refinación de características* es conservar el número mínimo de índices que permitan caracterizar tanto el problema como el desempeño del algoritmo sin sacrificar la calidad del conocimiento que pueda obtenerse al evaluar dichas medidas. De esta forma, debe realizarse una selección de índices eliminando aquellos que no sean relevantes, así como los redundantes, los incorrectos y los difícilmente calculables. En caso de ser necesario, nuevas variables son definidas mediante el uso de técnicas de análisis multivariado. Algunas herramientas de análisis de datos exploratorio permiten obtener índices de calidad para la caracterización del proceso de optimización, a través de técnicas como: identificación y tratamiento de datos no coherentes, resumen de datos, transformación de variables, estandarización de datos, análisis multivariado, entre otros.

El *análisis de correlación* permite identificar índices con un alto grado de asociación lineal entre ellos, que posiblemente miden un mismo factor; los índices redundantes pueden dificultar y causar errores en el estudio de relaciones, por lo que se recomienda seleccionar sólo los índices de caracterización representativos, eliminando aquellos no relevantes. Por otro lado, los *estadísticos descriptivos*, el *análisis tabular* y el *análisis gráfico* ayudan a observar tendencias importantes en los valores medidos por los índices de caracterización, para analizar la aportación y consistencia de cada índice; el objetivo es identificar valores inconsistentes o incorrectos, que se pueden deber a posibles fallas en el cálculo y selección de los índices de caracterización y sugieren la necesidad de una revisión de éstos. El *análisis de componentes principales* y el *análisis de agrupación* son

útiles para resumir los índices de caracterización y validar su poder de discriminación. Finalmente, el *análisis causal* ayuda a validar si el conjunto de índices seleccionados permiten describir factores que impactan el desempeño algorítmico.

4.1.3 ETAPA 3: ESTUDIO DE RELACIONES

En la tercera etapa, *estudio de relaciones*, se realiza un análisis exploratorio de las características del proceso de optimización con el propósito de obtener relaciones que expliquen el comportamiento de dicho proceso. El objetivo es identificar los factores que tienen mayor impacto en la dificultad de las instancias, el comportamiento del algoritmo y el desempeño final. Para cumplir este objetivo, se buscan asociaciones de causa-efecto entre los índices que caracterizan cada par de etapas del proceso de optimización, es decir: entrada→salida, entrada→proceso, proceso→salida.

En esta etapa, se utilizan *análisis de asociación*, *técnicas de visualización de datos* y *comparaciones tabulares* para descubrir relaciones de desempeño entre pares de índices y visualizar la intensidad, la forma y la dirección de estas relaciones. Los métodos de análisis multivariado, como el *análisis de agrupación* y el *análisis gráfico*, ayudan a identificar grupos de instancias del problema que tienen características comunes, para estudiar el comportamiento del algoritmo dentro de cada uno de los grupos. De manera complementaria, el uso de *herramientas de análisis causal* permite estudiar los índices de caracterización para determinar la estructura causal (modelo) que explique, en su conjunto, las relaciones de desempeño existentes en todo el proceso de optimización.

El conocimiento obtenido, producto del enfoque de caracterización propuesto, permite comprender el comportamiento del algoritmo heurístico objeto de estudio, explicando de qué manera su desempeño es afectado por el conjunto de factores que lo causan. Este conocimiento puede ser utilizado para clasificar y generar instancias con diferente grado de dificultad, así como para visualizar posibles mejoras en la estructura del algoritmo y mejorar su desempeño. En las siguientes secciones se muestra la aplicación del enfoque

de análisis propuesto en el estudio de desempeño de un algoritmo genético que soluciona el problema de empaqueo de objetos en contenedores (BPP).

4.2 ANÁLISIS DEL PROCESO DE OPTIMIZACIÓN DE BPP

En esta sección se presenta la aplicación del enfoque experimental propuesto para la caracterización del proceso de optimización de BPP, mediante el análisis de desempeño de un algoritmo genético del estado del arte denominado HGGA-BP [Nieto07]. Como aportaciones, producto del estudio realizado, se introduce un nuevo conjunto de índices para la caracterización de BPP y del comportamiento algorítmico; además, se realiza el rediseño de la estructura de algoritmo HGGA-BP, mejorando su desempeño.

4.2.1 ETAPA 1. CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN DE BPP

En esta etapa se lleva a cabo la caracterización del problema de investigación (BPP), del comportamiento del algoritmo de interés (HGGA-BP) y del desempeño final obtenido. Una vez identificados los elementos que influyen en el proceso de optimización se analiza qué aspectos es posible medir en cada categoría, y se definen los índices que caracterizan estos factores.

Caracterización de BPP

BPP consiste en almacenar un conjunto de n objetos de diferentes pesos $W = \{w_1, \dots, w_n\}$ en el menor número de contenedores de tamaño fijo sin violar la capacidad c de ningún contenedor. Caracterizar la estructura de una instancia de BPP es un punto clave para predecir el desempeño que tendrá un algoritmo heurístico al momento de solucionarla. Es conocido que factores como el número de objetos, la tendencia central de sus pesos y la forma en que se distribuyen, impactan en el grado de dificultad que una instancia pueda tener para un algoritmo de solución; sin embargo, el reto es la formulación de índices que sean capaces de caracterizar estos factores.

En la Sección 3.2 se introdujeron las instancias estándar de BPP propuestas para evaluar el desempeño de los algoritmos de solución; así mismo se presentó un conjunto de índices para la caracterización de BPP propuestos en la literatura [Cruz04, Álvarez06, Pérez07, Landero08]. Dichos índices son retomados en este trabajo. Para comenzar, se realizó el análisis del conjunto de índices para BPP de la literatura (ver Sección 3.2.4, Tabla 3.5 y Tabla 3.6), analizando si permitían discriminar entre las diferentes instancias de BPP (ver Sección 3.2.2). Para contribuir en este estudio y en la caracterización de BPP, cada instancia fue graficada mostrando la distribución de los pesos de los objetos en relación a la capacidad del contenedor.

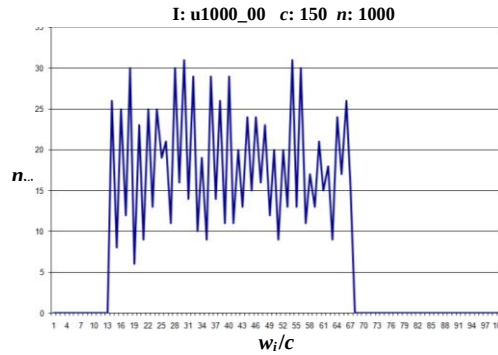


Figura 4.2 Gráfica de distribución de pesos para la instancia u1000_00.

La Figura 4.2 muestra la gráfica de distribución de una instancia de la clase u_1000 (Uniform). En la parte superior de la gráfica puede apreciarse el nombre de la instancia (I: u1000_00), la capacidad del contenedor (c : 150) y el número de objetos (n : 1000). El eje horizontal representa el peso de los objetos w_i como porcentaje de la capacidad del contenedor ($0 < w_i/c \leq 1$). El eje vertical cuantifica el número de objetos n_w en cada porcentaje de pesos w_i/c . Para la instancia de la gráfica, los pesos de los objetos están distribuidos de manera uniforme entre 20 y 100, es decir, entre el 13% y el 67% de la capacidad del contenedor. En la gráfica también puede observarse que el número de objetos con cada peso varía entre 6 y 31.

La Figura 4.3 muestra ejemplos de gráficas de distribución de las instancias estándar. En ella puede observarse la variedad de formas que toma la distribución de frecuencia de

los pesos de los objetos, así como los rangos de valores para los pesos, sugiriendo la importancia de una correcta caracterización de estos factores. El análisis de las gráficas de distribución permite identificar diferencias entre las distintas clases de instancias y ayuda a definir índices de caracterización.

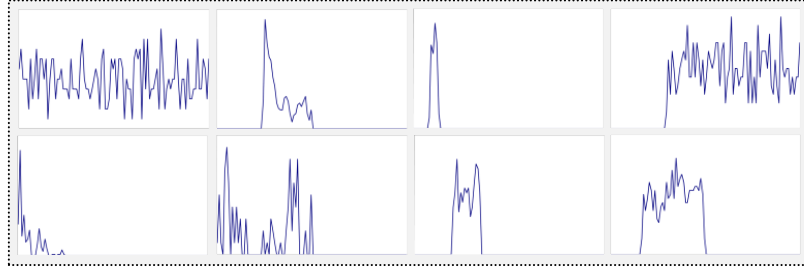


Figura 4.3 Ejemplos de gráficas de distribución de pesos para las instancias estándar de BPP.

El estudio de las gráficas de caracterización reveló que los índices propuestos por Cruz y Álvarez [Cruz04, Álvarez06] logran caracterizar la mayor parte de la estructura de una instancia de BPP. Sin embargo, hay ciertos factores que no fueron tomados en cuenta por los autores y que permiten puntualizar diferencias entre instancias. En este trabajo de investigación se proponen cinco nuevos índices de caracterización de BPP. Los nuevos índices permiten: **ubicar el rango de la distribución del conjunto de pesos**; **identificar tendencias importantes en los pesos** de los objetos, al tomar en cuenta la frecuencia de repetición de los pesos; y contribuir en la **caracterización de la forma de la distribución de los pesos**. Dichos índices son definidos a continuación.

Los índices *Menor* y *Mayor* miden, respectivamente, el peso del objeto más pequeño y del objeto más grande en relación al tamaño del contenedor y permiten ubicar el inicio y el final de la distribución de pesos. Los índices son definidos por medio de la Ecuación 4.1 y la Ecuación 4.2, donde W es el arreglo de pesos de objetos, c es la capacidad del contenedor, $\min(W)$ y $\max(W)$ son funciones que devuelven el peso menor y el peso mayor de W , respectivamente.

$$Menor = \frac{\min(W)}{c} \quad (4.1)$$

$$Mayor = \frac{\max(W)}{c} \quad (4.2)$$

Multiplicidad caracteriza el número promedio de repeticiones de cada peso de objeto y ayuda a identificar tendencias o picos en la distribución de frecuencias de pesos. La Ecuación 4.3 define este índice, m_i es el número de objetos con peso s_i , donde S es un subconjunto de W que guarda los $\hat{n}$ diferentes pesos, sin repeticiones ($1 \leq \hat{n} \leq n$).

$$Multiplicidad = \frac{\sum_{i=1}^{\hat{n}} m_i}{\hat{n}} \quad (4.3)$$

Mayor_Multiplicidad representa la frecuencia máxima con la que un peso se repite en el conjunto de objetos. La Ecuación 4.4 define este índice, donde M guarda la frecuencia de aparición de cada diferente peso. La comparación de los valores de esta medida con los de *Multiplicidad*, puede sugerir la existencia de puntos de la distribución de pesos que posean una mayor acumulación objetos.

$$Mayor_Multiplicidad = \max(M) \quad (4.4)$$

Uniformidad mide el grado de uniformidad de la distribución de los pesos de los objetos, a partir de la división del rango de datos en cuatro segmentos de igual magnitud y las diferencias entre el número esperado y el número real de objetos en cada subrango. Sea

n = número de objetos

$W = \{w_i : w_i < w_{i+1}\}, 1 \leq \forall i \leq n$, arreglo de pesos de objetos en orden creciente

$R = \max(W) - \min(W)$, amplitud del rango de distribución de pesos de los objetos

$r_i = \min(W) + iR/4, 0 \leq i \leq 4$, marcador i de división en subrangos

El conjunto de objetos contenido en cada subrango j es:

$$B_j = \{w \in W : r_{j-1} < w \leq r_j\}, 1 \leq j \leq 4$$

La Ecuación 4.5 mide el grado de uniformidad de la distribución del conjunto de pesos. Un valor de *Uniformidad* cercano a uno, representa una distribución uniforme y mientras más alejado se encuentre este valor de uno, se observarán mayores diferencias en la frecuencia de aparición de cada uno de los pesos de los objetos.

$$Uniformidad = 1 - \frac{\sum_{j=1}^4 \left| \left(\frac{n}{4} - |B_j| \right) \right|}{n} \quad (4.5)$$

Caracterización del Comportamiento Algorítmico

El algoritmo objeto de estudio de este trabajo de investigación es un genético para BPP denominado HGGA-BP [Nieto07]. El Algoritmo 4.1 describe el procedimiento general.

ALGORITMO GENÉTICO HGGA-BP	
1	Inicio
2	Inicializar parámetros
3	para cada <i>generación</i>
4	para cada individuo <i>actual</i> de la población
5	<i>nuevo</i> = Generar Individuo
6	Calcular aptitud del individuo <i>nuevo</i>
7	si <i>generación</i> > 1 y <i>aleatorio</i> [0-1) <= 0.95
8	si <i>aptitud</i> (<i>nuevo</i>) > <i>aptitud</i> (<i>actual</i>)
9	<i>actual</i> = <i>nuevo</i>
10	fin si
11	si no
12	<i>actual</i> = <i>nuevo</i>
13	fin si
14	fin para
15	Seleccionar individuos a cruzar de forma proporcional
16	Aplicar Cruzamiento por Agrupación y generar <i>nuevos individuos</i>
17	Sustituir los peores individuos de la población por los <i>nuevos individuos</i>
18	Aplicar Mutación por Agrupación a los peores individuos de la población
19	Registrar la mejor solución
20	fin para
21	fin procedimiento

Algoritmo 4.1 Procedimiento general del algoritmo HGGA-BP.

En cada iteración, el proceso comienza generando una población inicial para lo cual se crea un conjunto de individuos en forma no determinista (Líneas 4-14). Después de la primera generación, la probabilidad de mantener al individuo de la generación previa es

proporcional a su aptitud y a la generación de un número aleatorio (Líneas 7-13). Dado un porcentaje de cruzamiento y considerando su aptitud, se selecciona un conjunto de individuos para aplicarles el operador de cruzamiento (Líneas 15-16). Posteriormente, según el porcentaje de mutación, los peores individuos sufren pequeñas alteraciones genéticas, con el objetivo de mejorar su aptitud (Línea 18). Al final de cada generación la mejor solución de la población es actualizada. La salida (resultado final) del algoritmo es el individuo más apto de todo el proceso evolutivo.

Para la *generación de individuos* se utilizan heurísticas simples para BPP, que manejan órdenes aleatorios del conjunto de objetos. Las estrategias de construcción de individuos inician con el cálculo de un límite inferior L_1 del número de contenedores de la solución óptima. Una vez calculado dicho límite, los L_1 objetos más pesados son almacenados cada uno en un contenedor y el resto de los objetos son acomodados, en orden aleatorio, con las heurísticas BF y B3F (descritas en la Sección 2.4). La Ecuación 4.6 define el límite L_1 , donde w_i es el peso del objeto i y c es la capacidad del contenedor.

$$\left\lceil \frac{\sum_{j=1}^n w_j}{c} \right\rceil \quad (4.6)$$

El *cálculo de la aptitud* de cada individuo es realizado utilizando la función de costo de Falkenauer [Falkenauer92b], que evalúa el promedio de llenado de los contenedores de una solución. El objetivo del algoritmo HGGA-BP es maximizar el valor de las aptitudes de los individuos que conforman la población. La Ecuación 4.7 define el cálculo de la aptitud, donde m es el número de contenedores utilizado en la solución, S_i es la suma de los tamaños de los objetos en el contenedor i y c es la capacidad del contenedor.

$$F_{BPP} = \frac{\sum_{i=1}^m (S_i / c)^2}{m} \quad (4.7)$$

Las *estrategias* incluidas en HGGA-BP *que producen cambios en las soluciones* y, por lo tanto, pueden encontrar la solución óptima, o una buena solución, de una instancia de BPP son enumeradas a continuación. Estas técnicas son tomadas en cuenta al llevar a cabo la caracterización del proceso algorítmico, para identificar los métodos que tienen mayor impacto en el desempeño.

1. *LB+BF_Aleatorio* (Construcción de individuos)
2. *LB+B3F_Aleatorio* (Construcción de individuos)
3. *Cruzamiento por Agrupación*
4. *Mutación por Agrupación*

Tanto los parámetros del algoritmo HGGA-BP, como cada una de las estrategias que lo conforman, influyen en el desempeño obtenido al solucionar una instancia de BPP. La formulación de índices adecuados que cuantifiquen estos factores, contribuirá con la obtención de explicaciones que permitan comprender el comportamiento algorítmico. En este trabajo se propone un conjunto de índices que permiten: observar el impacto de los principales parámetros de control del algoritmo; analizar la superficie de aptitudes de los individuos en cada generación; identificar la importancia y contribución de cada método incluido en el algoritmo; y evaluar la calidad de las soluciones que forman la población.

Para caracterizar el ***proceso de generación de individuos*** se definieron tres índices los cuales permiten evaluar el número y la calidad de los individuos nuevos, aceptados en cada generación, para formar parte de la población.

Nuevos representa el número promedio de individuos, de reciente creación, aceptados en cada generación. La Ecuación 4.8 define este índice, donde *nuevos_i* es el número total de individuos aceptados en la generación *i* y *g* indica el número de generaciones que el algoritmo iteró. Los individuos *Nuevos* sólo pudieron ser aceptados por dos razones: 1) porque el número aleatorio generado en el proceso de sustitución de individuos cayó en

el 5% de probabilidad de reemplazar el individuo actual; ó 2) debido a que su aptitud superaba la aptitud de un individuo actual (ver Algoritmo 4.1, Líneas 7-13).

$$Nuevos = \frac{\sum_{i=1}^g nuevos_i}{g} \quad (4.8)$$

Buenos cuantifica el número promedio de individuos nuevos, aceptados por generación, que superaron la aptitud del individuo que sustituyeron. La Ecuación 4.9 detalla este índice, donde *buenos_i* indica el número de individuos buenos que fueron aceptados en la generación *i* y *g* representa el número total de generaciones.

$$Buenos = \frac{\sum_{i=1}^g buenos_i}{g} \quad (4.9)$$

Malos mide el número promedio de individuos nuevos, aceptados por generación, cuya aptitud era menor que la del individuo que sustituyeron. La Ecuación 4.10 define la función de caracterización, *malos_i* indica el número de individuos malos aceptados en la generación *i* y *g* es el número de generaciones. La única posibilidad de que un individuo malo sea aceptado es que el número aleatorio generado en el proceso de sustitución de individuos haya caído entre 0.95 y 1 (ver Algoritmo 4.1, Líneas 7-13).

$$Malos = \frac{\sum_{i=1}^g malos_i}{g} \quad (4.10)$$

La **calidad de las soluciones** que conforman la población generada por el algoritmo es evaluada mediante cuatro índices. Estas medidas permiten registrar las aptitudes de los individuos que conforman la población en cada generación y posibilitan el análisis del proceso evolutivo recorrido por el algoritmo, observando en qué puntos se alcanzan las peores y las mejores soluciones.

$mejor_i$ y $peor_i$ registran, respectivamente, la aptitud del mejor y el peor individuo de la población en la generación i . Los índices son definidos por medio de la Ecuación 4.11 y la Ecuación 4.12, donde P es un conjunto que incluye las aptitudes de los individuos que forma parte de la población y $\max(P)$ y $\min(P)$ son funciones que devuelven la aptitud máxima y mínima, respectivamente.

$$mejor_i = \max(P) \quad (4.11)$$

$$peor_i = \min(P) \quad (4.12)$$

$aptitud_i$ cuantifica el promedio de la aptitud de todos los individuos de la población en la generación i . La Ecuación 4.13 define este índice, donde p_j es la aptitud del individuo j y t indica el tamaño de la población.

$$aptitud_i = \frac{\sum_{j=1}^t p_j}{t} \quad (4.13)$$

Mejor mide la aptitud promedio de las mejores soluciones obtenidas en cada generación. La Ecuación 4.14 define la medida, donde $mejor_i$ es la aptitud del mejor individuo de la generación i (ver Ecuación 4.11) y g es el número de generaciones.

$$Mejor = \frac{\sum_{i=1}^g mejor_i}{g} \quad (4.14)$$

Otro factor importante a estudiar, para comprender el comportamiento algorítmico, es la **superficie de aptitudes** de los individuos que participan en el proceso evolutivo. Este factor, es caracterizado mediante un conjunto de cuatro índices que permiten analizar la variabilidad en los individuos que conforman la población, así como la asociación que puede existir entre las mejores soluciones. Valores de variabilidad bajos pueden indicar posibles características del problema y del algoritmo. Por ejemplo, podría tratarse de una instancia de BPP para la que el número de combinaciones entre objetos es muy limitado,

por lo que el algoritmo no muestra mucha desviación en las aptitudes de los individuos que genera. También podría ocurrir que el algoritmo se estancara en ciertas zonas del espacio de búsqueda, generando soluciones muy parecidas, lo que indicaría la necesidad de incluir heurísticas que le permitan escapar de dichas zonas.

$desv_aptitud_i$ mide la variabilidad en la aptitud de los individuos que conforman la población de la generación i . El índice es definido por la Ecuación 4.15, que calcula la desviación estándar de las aptitudes, donde p_j es la aptitud del individuo j , $aptitud_i$ es el promedio de las aptitudes de la población en la generación actual (Ecuación 4.13) y t indica el tamaño de la población. La ecuación 4.16 mide el promedio de las desviaciones ocurridas durante todo el proceso evolutivo, donde g es el total de generaciones.

$$desv_aptitud_i = \sqrt{\frac{\sum_{j=1}^t (aptitud_i - p_j)^2}{t}} \quad (4.15)$$

$$Desv_Aptitud = \frac{\sum_{i=1}^g desv_aptitud_i}{g} \quad (4.16)$$

$Desv_Mejor$ representa la desviación en las aptitudes de los mejores individuos de cada generación. La Ecuación 4.17 define el índice, donde $mejor_i$ es la mejor aptitud de la generación i (ver Ecuación 4.11), $Mejor$ es el promedio de las aptitudes de los mejores individuos del proceso evolutivo (ver Ecuación 4.14) y g indica el total de generaciones.

$$Desv_Mejor = \sqrt{\frac{\sum_{i=1}^g (Mejor - mejor_i)^2}{g}} \quad (4.17)$$

La *Función de Auto-correlación*, ρ_1 , propuesta por Weinberger [Weinberger90], es propuesta para medir las asociaciones existentes entre las mejores soluciones. Para su cálculo, sea $mejor_i$ la mejor aptitud de la generación i , $Mejor$ el promedio de las

aptitudes de los mejores individuos del proceso evolutivo (ver Ecuación 4.14) y g el total de generaciones. La Ecuación 4.18 define este índice. Un valor absoluto de ρ_1 cercano a uno indica que las diferencias entre las aptitudes de cualquier par de soluciones son muy similares, lo que sugiere una superficie de aptitudes un tanto plana. Por otro lado, un valor de ρ_1 cercano a cero indica que las aptitudes son casi independientes, y por tanto la superficie puede presentar valles y picos. Una vez más esto puede estar relacionado con las características del problema de optimización.

$$\rho_1 = \frac{\sum_{i=1}^{g-1} (mejor_i - Mejor)(mejor_{i+1} - Mejor)}{\sum_{i=1}^g (mejor_i - Mejor)^2} \quad (4.18)$$

Finalmente, las estrategias que tienen mayor impacto en la obtención de las mejores soluciones del algoritmo son registradas a través del índice *Mejor_Heurística*. Este índice, obtiene la estrategia del algoritmo que permitió alcanzar el mayor número de mejores soluciones en r ejecuciones del algoritmo. La Ecuación 4.19 define la medida, donde H es el conjunto de heurísticas, s_i representa el número de veces que la heurística i generó la solución final del algoritmo y $\arg \max$ devuelve el índice de la heurística que presenta el mayor número de repeticiones.

$$Mejor_Heurística = \arg \max_{i \in H} (s_i) \quad (4.19)$$

Caracterización del Desempeño Final

El desempeño final del algoritmo se mide por tres tipos de información: calidad de la solución, la consistencia de los resultados y el tiempo que le llevó en encontrar la mejor solución, los cuales se describen a continuación.

La **calidad de las soluciones** obtenidas por el algoritmo es caracterizada mediante cuatro índices. Estos índices, registran las aptitudes de las soluciones finales alcanzadas en diferentes corridas del algoritmo, así como el número de contenedores usado en dichas

soluciones. Un valor de aptitud $F_{BPP} = 1$, significa que los contenedores que conforman la solución se encuentran completamente llenos, y por lo tanto, la solución es óptima.

Los índices $MejorF_{BPP}$ y $PeorF_{BPP}$ registran, respectivamente, la aptitud de la mejor y la peor solución obtenida para una instancia de BPP, en r corridas del algoritmo. Las ecuaciones 4.20 y 4.21 definen estos índices, donde S es un conjunto que incluye las aptitudes de r soluciones obtenidas en r ejecuciones del algoritmo y $\max(S)$ y $\min(S)$ son funciones que devuelve el mayor y el menor elemento del conjunto S .

$$MejorF_{BPP} = \max(S) \quad (4.20)$$

$$PeorF_{BPP} = \min(S) \quad (4.21)$$

$\overline{F}_{BPP}$ calcula el promedio de las aptitudes de las soluciones obtenidas, para una instancia de BPP, en r corridas del algoritmo. La Ecuación 4.22 precisa este índice, donde F_{BPPi} es la aptitud de la solución alcanzada en la corrida i y r es el total de ejecuciones.

$$\overline{F}_{BPP} = \frac{\sum_{i=1}^r F_{BPPi}}{r} \quad (4.22)$$

El *Radio_teorico* es otra medida de calidad, y representa la razón entre el número de contenedores usados por la mejor solución encontrada por el algoritmo y el número de contenedores de la solución óptima. Cuando la solución óptima no es conocida, en lugar de ésta, se puede usar el valor del mejor límite inferior (óptimo teórico) reportado en la literatura. La Ecuación 4.23 define este índice, que es una medida de error, donde z es el número de contenedores de la solución dada por el algoritmo y z_{opt} es el número de contenedores de la solución óptima.

$$Radio_teórico = \frac{z}{z_{opt}} \quad (4.23)$$

Para caracterizar la **estabilidad del algoritmo** se mide la desviación estándar de los resultados obtenidos. $Desv_Z$ cuantifica la variabilidad las soluciones encontradas para una instancia de BPP en diferentes ejecuciones del algoritmo. Esta medida es obtenida mediante la Ecuación 4.24, donde z_i es el número de contenedores que conforman la solución alcanzada en la corrida i y $\bar{z}$ es el promedio de las soluciones encontradas en r corridas. Un valor de desviación alto puede indicar que una instancia de BPP es difícil de resolver o que las estrategias aleatorias incluidas en el algoritmo lo hacen presentar un comportamiento poco robusto.

$$Desv_Z = \sqrt{\frac{\sum_{i=1}^r (\bar{z} - z_i)^2}{r}} \quad (4.24)$$

El factor **tiempo** es caracterizado a través de dos índices. El primero, mide el tiempo real que requirió la ejecución del algoritmo antes de encontrar la solución al problema. El segundo, cuantifica el número de generaciones que el algoritmo tuvo que iterar antes de encontrar la que finalmente fue la mejor solución.

La Ecuación 4.25 define el índice *Tiempo* que cuantifica el promedio del tiempo, en segundos, que el algoritmo tarda en encontrar una solución, para una instancia de BPP. Para r corridas, el tiempo utilizado en la corrida i es cuantificado por $tiempo_i$.

$$Tiempo = \frac{\sum_{i=1}^r tiempo_i}{r} \quad (4.25)$$

Generación representa la generación promedio en que el algoritmo encuentra la mejor solución de una instancia. Este índice es definido mediante la Ecuación 4.26, donde gen_i es la generación en la que se encontró la mejor solución de la corrida i .

$$Generación = \frac{\sum_{i=1}^r gen_i}{r} \quad (4.26)$$

4.2.2 ETAPA 2. REFINACIÓN DE CARACTERÍSTICAS

En la etapa anterior se propuso el uso de un conjunto de índices para la caracterización de BPP, del comportamiento del algoritmo HGGA-BP y del desempeño final. Una vez que dichos índices son aplicados, es necesario realizar el análisis de cada uno de ellos y de su aportación en la medición de las características significativas de los factores que afectan el desempeño algorítmico. El objetivo de la refinación es encontrar un conjunto mínimo de índices de calidad, que describan correctamente el proceso de optimización. Los estudios realizados en esta etapa pretenden: eliminar medidas redundantes; descartar índices incorrectos y poco relevantes; y validar los índices propuestos.

En el caso de la caracterización del problema, los índices propuestos permiten conocer la estructura de una instancia de BPP (ver Sección 4.2.1), midiendo los siguientes aspectos:

1. Tamaño del problema.
2. Tendencia central de los pesos de los objetos.
3. Dispersión de los pesos.
4. Forma de la distribución de frecuencia de los pesos.
5. Ubicación del rango de la distribución del conjunto de pesos.
6. Valores que dividen el conjunto de pesos en partes iguales (Posición).
7. Relaciones entre los pesos de los objetos y el tamaño del contenedor.
8. Multiplicidad de pesos de objetos.

La Tabla 4.1 detalla los 31 índices que permiten caracterizar estos factores, 26 índices han sido tomados de la literatura (ver Sección 3.2.4, Tabla 3.5 y Tabla 3.6) y cinco son propuestos en este trabajo (ver Sección 4.2.1). Cada instancia debe ser caracterizada en las mismas unidades, para poder realizar futuros análisis en búsqueda de relaciones de desempeño. Por lo anterior, algunas medidas deben ser estandarizadas, considerando los pesos de los objetos como porcentaje de la capacidad del contenedor.

Tabla 4.1 Índices de caracterización de BPP.

Medidas	Estadístico
Tamaño	p [Cruz04]
Tendencia Central	<i>Media aritmética</i> [Álvarez06]
	<i>Media geométrica</i> [Álvarez06]
	<i>Media armónica</i> [Álvarez06]
	<i>Mediana</i> [Álvarez06]
	<i>Moda</i> [Álvarez06]
	t [Cruz04]
Dispersión	<i>Rango</i> [Álvarez06]
	<i>Desviación media</i> [Álvarez06]
	<i>Varianza</i> [Álvarez06]
	<i>Desviación estándar</i> [Álvarez06]
	<i>Cuasivarianza</i> [Álvarez06]
	<i>Cuasidesviación estándar</i> [Álvarez06]
	<i>Coefficiente variación</i> [Álvarez06]
	<i>Error estándar</i> [Álvarez06]
	d [Cruz04]
Forma	<i>Asimetría Pearson</i> [Álvarez06]
	<i>Asimetría Pearson mediana</i> [Álvarez06]
	<i>Asimetría Pearson moda</i> [Álvarez06]
	<i>Asimetría Bowley</i> [Álvarez06]
	<i>Curtosis</i> [Álvarez06]
	<i>Uniformidad</i> [Este Trabajo]
Ubicación	<i>Menor</i> [Este Trabajo]
	<i>Mayor</i> [Este Trabajo]
Posición	<i>Cuartiles</i> [Álvarez06]
	<i>Deciles</i> [Álvarez06]
	<i>Percentiles</i> [Álvarez06]
Relaciones Objetos/Contenedor	f [Cruz04]
	b [Cruz04]
Multiplicidad	<i>Multiplicidad</i> [Este Trabajo]
	<i>Mayor Multiplicidad</i> [Este Trabajo]

A simple vista, parece ser que varios aspectos de la estructura del problema (BPP) son caracterizados mediante una gran cantidad de índices, por lo que es posible que algunos de ellos sean redundantes o innecesarios. El análisis exploratorio de las medidas ayudará a verificar lo anterior.

En los siguientes párrafos se describe el proceso de análisis exploratorio de datos llevado a cabo con el objetivo de seleccionar el conjunto final de índices de caracterización para el proceso de optimización de BPP con HGGA-BP. En primera instancia, el cálculo de la matriz de correlación y de otros estadísticos descriptivos, posibilitaron identificar las

asociaciones existentes entre los valores de los índices. Posteriormente, el estudio de gráficas de dispersión y de distribución ayudó a visualizar la *utilidad* y *consistencia* de cada medida. Por último, el uso de herramientas de análisis multivariado y de análisis causal, permitió *validar* el conjunto final de índices de caracterización.

Análisis de Correlación

El cálculo del coeficiente de correlación entre cada par de índices, permitió descubrir la existencia de *colinealidad* entre ciertas variables. La Tabla 4.2 muestra una parte de la matriz de correlación, donde se observa una relación lineal excesiva entre cinco de los índices de tendencia central. Si la matriz de correlación tiene colinealidad es posible que los análisis estadísticos basados en ésta no tengan sentido o que el resultado obtenido sea erróneo [Johnson00]. Para solucionar este problema, se tomó la decisión de seleccionar una sola variable para caracterizar el peso promedio de un objeto. El índice seleccionado fue *t* [Cruz04], pues es el único que no requiere una posterior estandarización, debido a que considera los pesos de los objetos como porcentaje de la capacidad del contenedor.

Tabla 4.2 Matriz de correlación de variables de tendencia central.

	<i>Media_</i> <i>aritmética</i>	<i>Media_</i> <i>geométrica</i>	<i>Media_</i> <i>armónica</i>	<i>Mediana</i>	<i>t</i>
<i>Media_aritmética</i>	1	0.9996	0.9978	0.9999	1
<i>Media_geométrica</i>	0.9996	1	0.9993	0.9996	0.9996
<i>Media_armónica</i>	0.9978	0.9993	1	0.9980	0.9978
<i>Mediana</i>	0.9999	0.9996	0.9980	1	0.9999
<i>t</i>	1	0.9996	0.9978	0.9999	1

El análisis de correlación mostró también la presencia de colinealidad entre las medidas que caracterizan la dispersión y la forma de la distribución de los pesos. Una vez más, algunos índices fueron descartados para eliminar la redundancia y evitar la presencia de problemas en análisis futuros. En general, se decidió conservar los índices que requerían menos cálculos. La Tabla 4.3 muestra el nuevo conjunto de índices, una vez eliminados los índices redundantes. El conjunto se redujo de 31 a 21 índices de caracterización.

Tabla 4.3 Índices de caracterización de BPP sin colinealidad.

Medidas	Estadístico
Tamaño	p [Cruz04]
Tendencia Central	$Moda$ [Álvarez06]
	t [Cruz04]
Dispersión	$Rango$ [Álvarez06]
	$Coeficiente_variación$ [Álvarez06]
	$Error_estándar$ [Álvarez06]
	d [Cruz04]
Forma	$Asimetría_Pearson$ [Álvarez06]
	$Asimetría_Pearson_moda$ [Álvarez06]
	$Asimetría_Bowley$ [Álvarez06]
	$Curtosis$ [Álvarez06]
	$Uniformidad$ [Este Trabajo]
Ubicación	$Menor$ [Este Trabajo]
	$Mayor$ [Este Trabajo]
Posición	$Cuartiles$ [Álvarez06]
	$Deciles$ [Álvarez06]
	$Percentiles$ [Álvarez06]
Relaciones Objetos/Contenedor	f [Cruz04]
	b [Cruz04]
Multiplicidad	$Multiplicidad$ [Este Trabajo]
	$Mayor_Multiplicidad$ [Este Trabajo]

Análisis de Aportación y Consistencia

Una vez identificado el nuevo conjunto de índices no colineales, es necesario analizar la contribución y consistencia de cada índice. El objetivo es identificar y eliminar aquellos índices que no aporten información significativa para la caracterización de las instancias de BPP. Los índices inconsistentes o incorrectos sólo agregan información errónea en la descripción de una instancia y no permiten discriminar entre instancias diferentes.

Después de analizado, el índice *Moda* fue descartado, dado que no permite caracterizar ningún aspecto importante del problema. Para una instancia de BPP, la información registrada por este índice es engañosa, pues muchos objetos suelen repetirse el mismo número de veces. Lo anterior se debe a que, en la mayoría de las instancias, los pesos de los objetos siguen una distribución más o menos uniforme (ver Sección 4.2.1, Figura 4.2 y Figura 4.3).

La Figura 4.4 muestra dos gráficas de dispersión para funciones de caracterización de tendencia central (*Moda* y *t*), dispersión (*d*) y forma (*Asimetría_Pearson*). Estos índices fueron aplicados en la caracterización de cinco conjuntos de instancias con diferente estructura (descritas en la Sección 3.2.2). En la primera gráfica, se utiliza el índice *Moda* como medida de tendencia central del conjunto de pesos de objetos y se puede observar que el índice no permite discriminar entre diferentes grupos de instancias. En la segunda gráfica, al sustituir *Moda* por *t*, se aprecia claramente la importancia de la información proporcionada por este índice, al relacionarse con los índices *d* y *Asimetría_Pearson*.

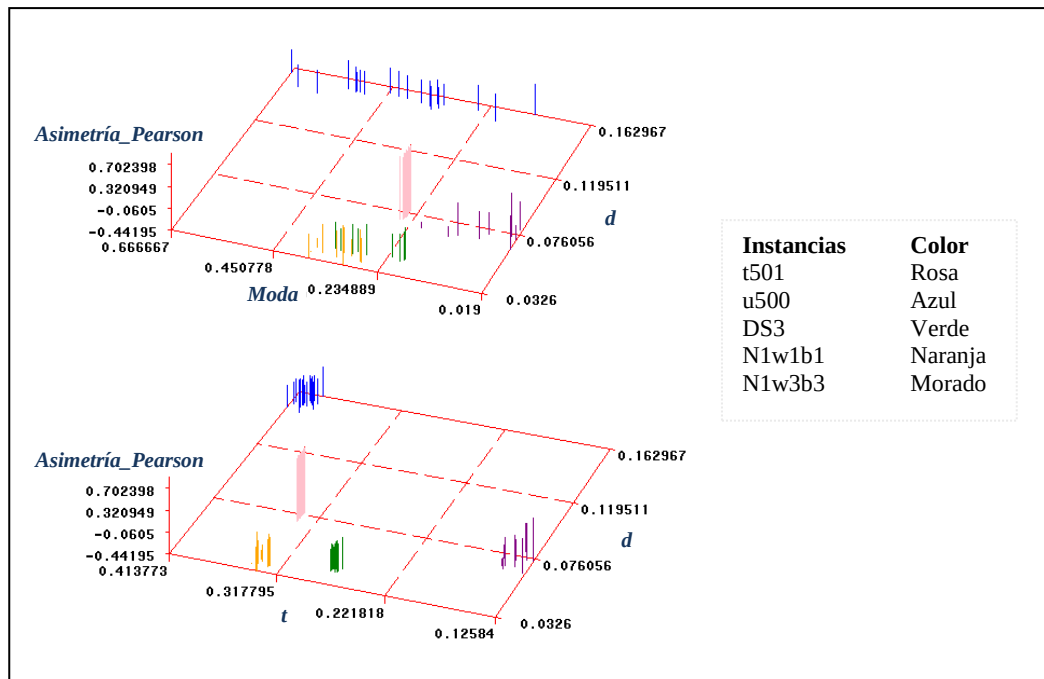


Figura 4.4 Gráficas de dispersión para *Moda*, *t*, *d* y *Asimetría_Pearson* en cinco conjuntos de instancias.

El análisis exploratorio también sugirió, que las medidas de posición: *Cuartiles*, *Deciles* y *Percentiles*, propuestas para caracterizar la uniformidad de la distribución del conjunto de pesos [Álvarez06], no aportaban información significativa para la caracterización de la estructura de una instancia de BPP. El estudio de las gráficas de distribución de las instancias demostró que los valores medidos por estos índices eran contradictorios y no correspondían con las características reales de las instancias. En las Figuras 4.5 y 4.6 se

presentan gráficas de distribución de dos instancias con diferente nivel de uniformidad. En la primera figura se muestra la estructura de la instancia de BPP menos uniforme; en la segunda se incluye un ejemplo de una instancia con una distribución uniforme; ambas pertenecen al conjunto Gau 1 (Sección 3.2.2). Los valores de los índices de posición, *Cuartiles*, *Deciles* y *Percentiles*, no permitieron diferenciar estas instancias entre sí, ni con respecto a otras instancias de mayor y menor uniformidad.

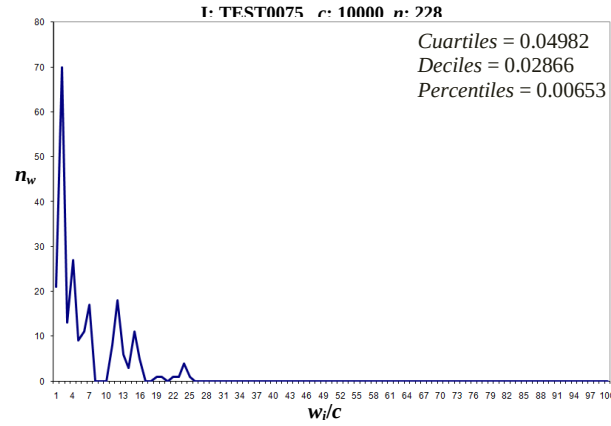


Figura 4.5 Gráfica de distribución de pesos para la instancia TEST0075.

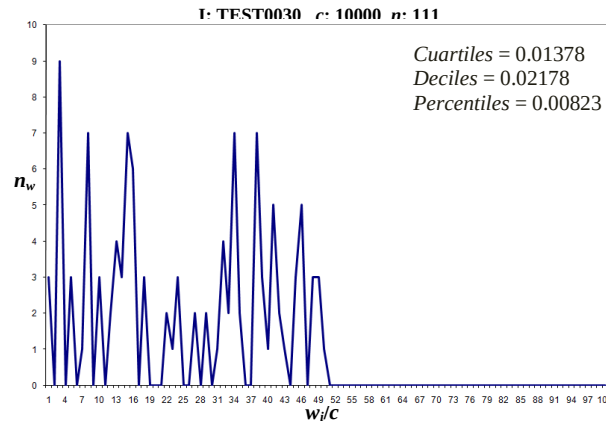


Figura 4.6 Gráfica de distribución de pesos para la instancia TEST0030.

La Figura 4.7 muestra tres gráficas de dispersión para los índices de posición (*Cuartiles*, *Deciles* y *Percentiles*), dispersión (*Rango*) y forma (*Asimetría_Pearson*), aplicados en la caracterización de seis conjuntos de instancias; las instancias TEST0075 y TEST0030 fueron marcadas con un diamante y con una cruz, respectivamente. Un valor alto en los índices de posición indica un bajo grado de uniformidad. En esta figura se aprecia la

existencia de contradicciones en los valores de los índices de posición. El grado de uniformidad asignado a las mismas instancias es diferente para los tres índices.

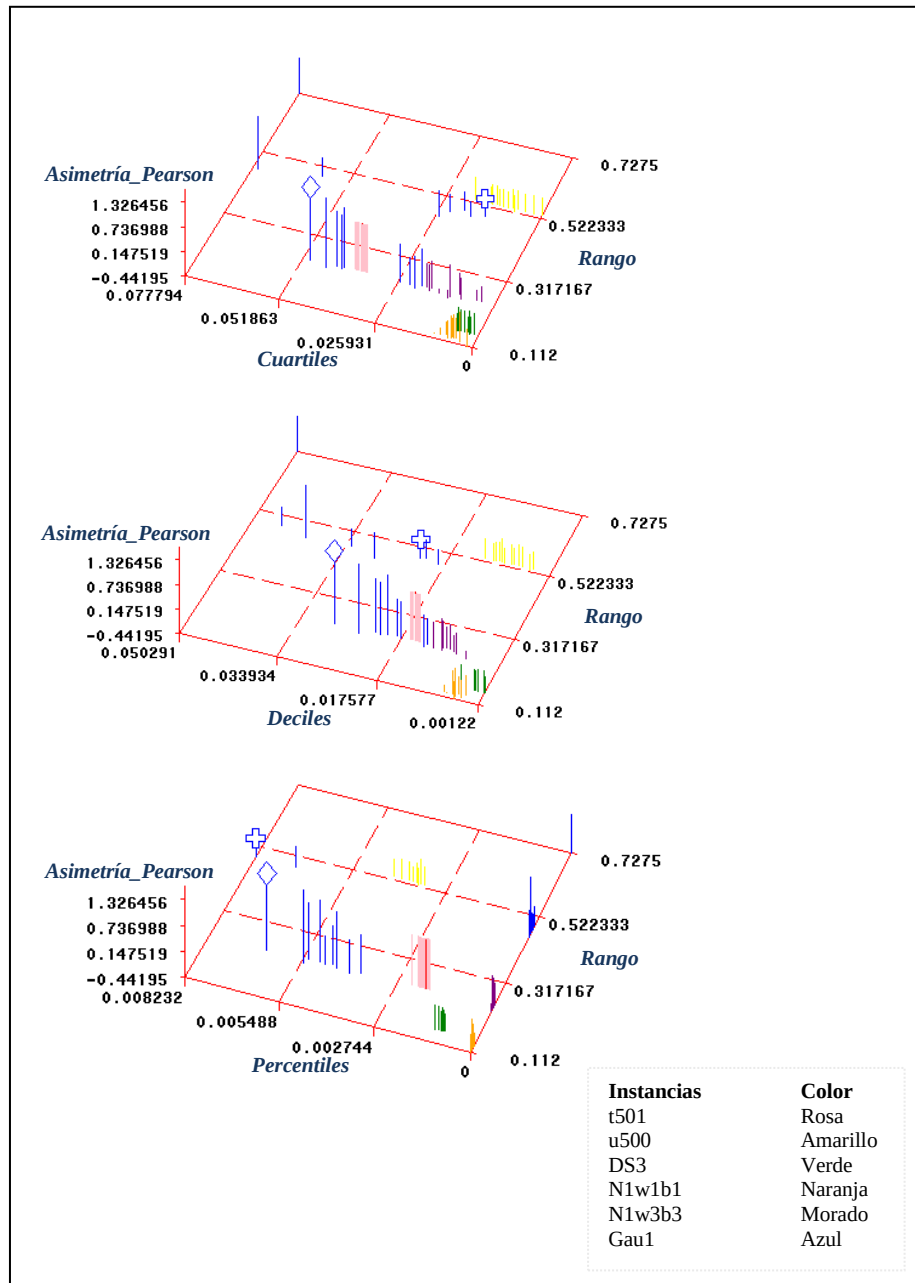


Figura 4.7 Gráficas de dispersión para índices de posición, *Rango* y *Asimetría_Pearson*.

La Figura 4.8 muestra una gráfica de dispersión similar a las presentadas en la Figura 4.7, pero haciendo uso del índice de caracterización *Uniformidad*, propuesto en este

trabajo (Sección 4.2.1, Ecuación 4.5). Las instancias TEST0075 y TEST0030 fueron marcadas, una vez más, con un diamante y una cruz. En esta figura se observa el poder discriminatorio y la consistencia del índice propuesto, el cual permite hacer una correcta distinción entre instancias con diferente grado de uniformidad.

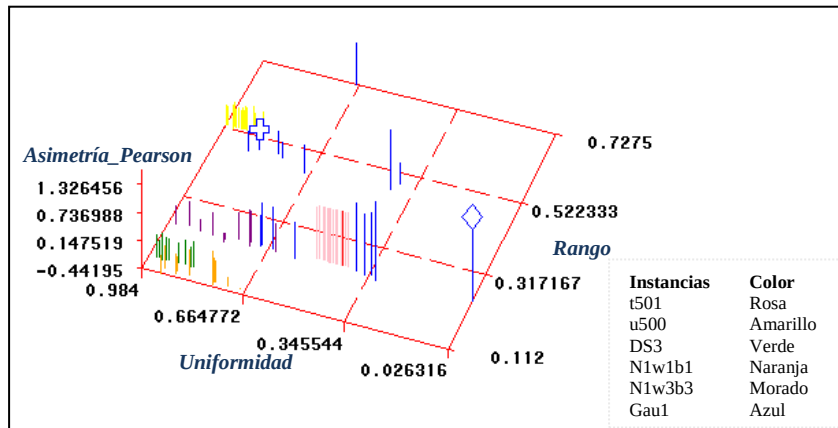


Figura 4.8 Gráfica de dispersión para las medidas de *Uniformidad*, *Rango* y *Asimetría_Pearson*.

Las conclusiones obtenidas en el estudio de aportación y consistencia de los índices de BPP llevaron a descartar los índices *Asimetría_Pearson_moda* y *Asimetría_Bowley*, que se basan, respectivamente, en los índices *Moda* y *Cuartiles*, cuya inconsistencia (para la caracterización de BPP) fue demostrada en párrafos anteriores. Por otro lado, el índice *Error_estándar* [Álvarez06], no tiene aplicación en la caracterización de la dispersión del conjunto de pesos de una instancia de BPP. El error estándar es una medida utilizada en muestreo estadístico, para calcular las desviaciones entre los estadísticos obtenidos de un conjunto de muestras aleatorias de una población. Por lo que, puede afirmarse que por definición este índice no es aplicable a la caracterización de BPP.

El análisis de los índices de caracterización del comportamiento algorítmico mostró que no existían índices incorrectos, redundantes o inconsistentes, pues cada índice captura aspectos importantes relativos a la estructura y el comportamiento del algoritmo HGGA-BP. Conclusiones similares fueron obtenidas al analizar las funciones de caracterización para el desempeño final.

Validación del Conjunto Final de Índices

El análisis exploratorio de los índices para BPP evidenció la contribución e importancia, de cada uno, en la caracterización del problema. Partiendo de un conjunto inicial de 31 índices, 10 fueron descartados debido a su redundancia y otros 7 fueron eliminados por su inconsistencia o nula aportación. El conjunto final de índices de caracterización es mostrado en la Tabla 4.4.

Tabla 4.4 Conjunto final de índices de caracterización de BPP.

Medidas	Estadístico
Tamaño	p [Cruz04]
Tendencia Central	t [Cruz04]
Dispersión	$Rango$ [Álvarez06]
	$Coefficiente_variación$ [Álvarez06]
	d [Cruz04]
Forma	$Asimetría_Pearson$ [Álvarez06]
	$Curtosis$ [Álvarez06]
	$Uniformidad$ [Este Trabajo]
Ubicación	$Menor$ [Este Trabajo]
	$Mayor$ [Este Trabajo]
Relaciones Objetos/Contenedor	f [Cruz04]
	b [Cruz04]
Multiplicidad	$Multiplicidad$ [Este Trabajo]
	$Mayor_Multiplicidad$ [Este Trabajo]

Un análisis de componentes principales ayudó a confirmar la importancia y aportación del conjunto final de índices. Este estudio fue realizado sobre los valores obtenidos al aplicar los índices a seis conjuntos de instancias de BPP (t501, u500, DS3, N3c1w1, Hard28 y Was 1). Las variables fueron agrupadas según tres componentes principales, que explican el 89.87% de la variabilidad total de los datos, cada componente captura un aspecto importante de la estructura de una instancia del problema.

El primer componente, *Variabilidad*, mide la variación en los pesos de los objetos y agrupa las medidas de tendencia central, dispersión y ubicación. El segundo, *Forma*, describe la distribución de los pesos, agrupando medidas de forma, ubicación y tamaño del problema. El último componente, *Multiplicidad*, mide las frecuencias de aparición de los pesos de los objetos. Estos tres componentes permiten resumir las características de

los cuatro grupos de instancias (ver Sección 3.2.2). La Figura 4.9 muestra una gráfica de dispersión para estos tres componentes en seis conjuntos de instancias. Como puede observarse, es posible distinguir claramente entre los seis tipos de instancias analizadas.

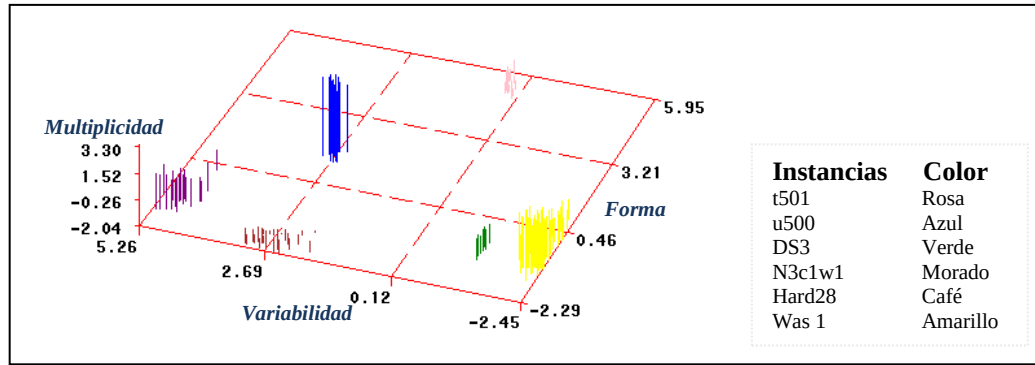


Figura 4.9 Gráfica de dispersión para los tres primeros componentes principales del conjunto de índices.

El análisis de las gráficas de distribución de frecuencia de pesos (definidas en la Sección 4.2.1) comprobó la exactitud y el poder discriminatorio de los índices propuestos, pues las similitudes y diferencias entre las características de las instancias pertenecientes a los seis conjuntos coincidieron con lo obtenido. Por ejemplo, la Figura 4.10, incluye dos gráficas representativas de la distribución de frecuencias de pesos en los conjuntos de instancias Was 1 y DS3, destacadas en color amarillo y verde en la Figura 4.9. En la Figura 4.10 puede observarse que, efectivamente, la forma de la distribución en este par de instancias es muy parecida y la diferencia entre ellas radica en el rango de valores de sus pesos y en el número de veces que un peso puede repetirse.

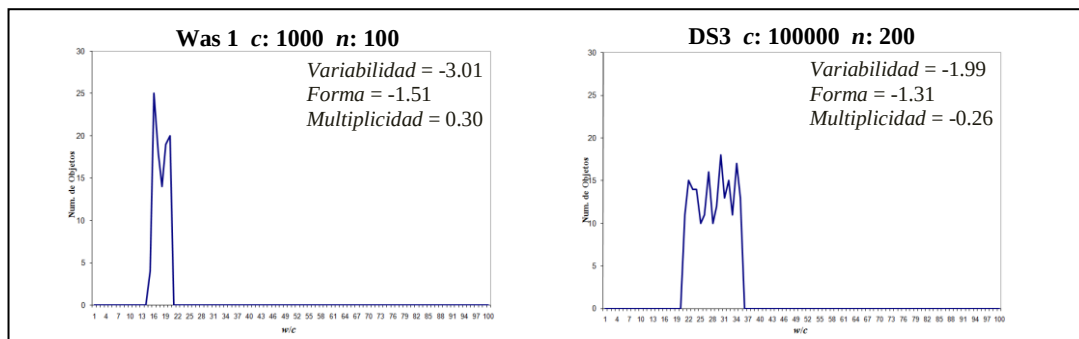


Figura 4.10 Gráficas de distribución de frecuencias de pesos para instancias Was 1 y DS3.

Por último, un análisis causal de las calificaciones de los componentes principales, que caracterizan la estructura de BPP, y de tres índices de caracterización de desempeño (ver Sección 4.2.1), confirmó que el conjunto final de índices permite describir factores que impactan el desempeño algorítmico. La Figura 4.11 muestra el grafo causal obtenido del análisis de los tres componentes, *Variabilidad*, *Forma* y *Multiplicidad*, y tres medidas del desempeño final, $\overline{F}_{BPP}$, *Radio_teorico* y *Generación*, para los seis conjuntos de instancias de la Figura 4.9.

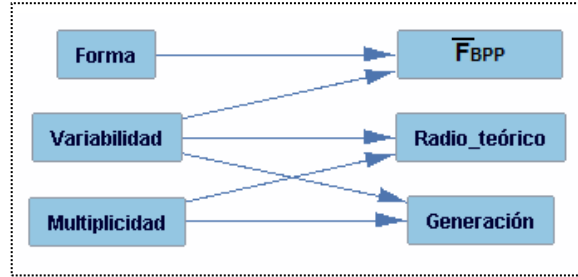


Figura 4.11 Relaciones causales entre las características del problema y desempeño de HGGA-BP.

Para los seis conjuntos de instancias analizados (t501, u500, DS3, N3c1w1, Hard28 y Was 1), el análisis causal sugirió que la calidad de las soluciones obtenidas HGGA-BP (*Radio_teorico*) y el número de iteraciones necesarias para encontrar la mejor solución (*Generación*), dependen de la variabilidad y la multiplicidad de los conjunto de pesos de las instancias. Así mismo, la aptitud promedio de la mejor solución ($\overline{F}_{BPP}$) parece estar relacionada con la forma y la variabilidad de la distribución de los pesos. Los resultados obtenidos mostraron la validez de los índices de caracterización de BPP aquí propuestos.

En el caso de los índices de caracterización del problema, una vez más, el análisis de componentes principales, permitió comprobar la exactitud discriminatoria de los índices. La Figura 4.12 muestra la gráfica de las calificaciones de los componentes, obtenidos al analizar las mediciones del comportamiento algorítmico al resolver cuatro conjuntos de instancias. Las instancias marcadas con una cruz son aquellas para las que el algoritmo no logra obtener la solución óptima.

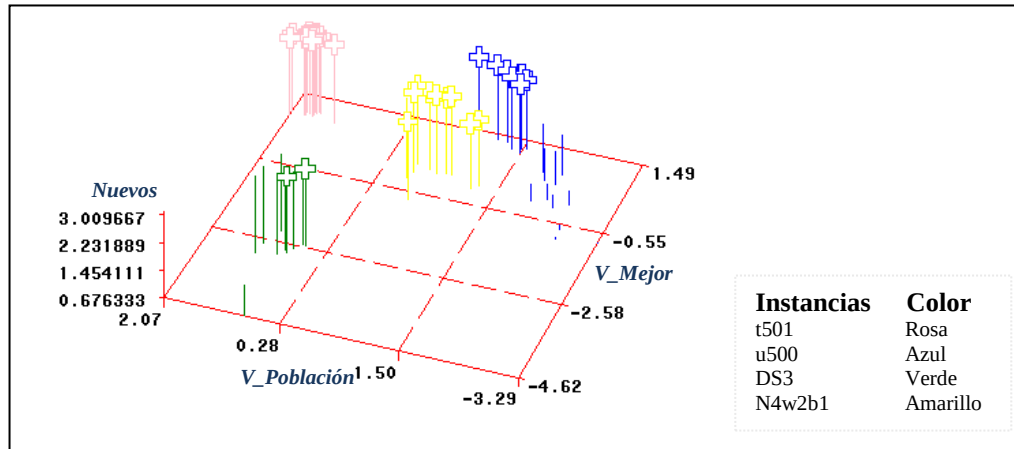


Figura 4.12 Gráfica de dispersión de los tres primeros componentes principales del conjunto de índices.

Para este grupo de instancias (t501, u500, DS3 y N4w2b1), el análisis agrupó los índices según tres componentes principales, que permiten explicar el 95% de la variabilidad total de los datos. El primero, *V_Población*, mide la variabilidad en la superficie de aptitudes de los individuos (soluciones de BPP) que conforman la población y agrupa índices de caracterización de calidad y superficie de aptitudes. El segundo, *V_Mejor*, describe la variabilidad en las aptitudes de los mejores individuos de cada generación e incluye índices de calidad de la población, así como de correlaciones y desviaciones existentes entre las mejores soluciones. El último componente, *Nuevos*, cuantifica el promedio de individuos que son aceptados en cada generación y agrupa los índices del proceso de generación de individuos. En la Figura 4.12 también se observa que hay algunas instancias para las que el valor del componente *Nuevos* es muy bajo. Esto ocurre porque el algoritmo encuentra la solución óptima de estas instancias en generaciones tempranas, terminando el proceso evolutivo cuando muy pocos individuos han sido creados.

El análisis causal de las calificaciones de los componentes principales, para el conjunto total de instancias, permitió comprobar que los índices propuestos caracterizan factores del comportamiento algorítmico que impactan en el desempeño final. La Figura 4.13 muestra el modelo causal obtenido en el estudio. En esta figura se observa que el componente *Nuevos* (destacado en color amarillo), parece no tener relación con el

desempeño final; este hecho puede deberse, más que a un error en el cálculo del índice, a una inconsistencia en el proceso algorítmico. La conjetura anterior será demostrada al analizar el comportamiento interno del algoritmo en la Sección 4.3.2.

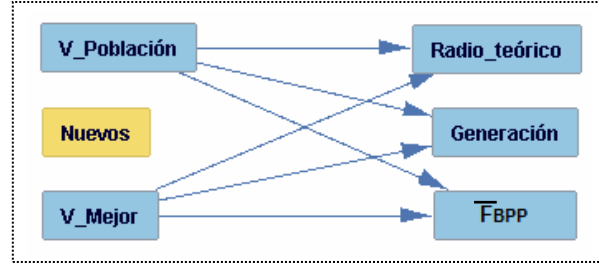


Figura 4.13 Relaciones causales entre el comportamiento y desempeño de HGGA-BP.

4.2.3 ETAPA 3. ESTUDIO DE RELACIONES

En esta última etapa, el conjunto final de índices de caracterización es estudiado, con la finalidad de descubrir relaciones de desempeño que detallen el proceso de optimización. El objetivo principal de este estudio es adquirir conocimiento que permita comprender el desempeño del algoritmo. Para este fin, los índices que caracterizan las tres etapas del proceso de optimización algorítmico son analizados en búsqueda de relaciones de tipo causa-efecto.

Para poder explicar el desempeño del algoritmo metaheurístico HGGA-BP es necesario descubrir y analizar las relaciones existentes entre los índices que definen la estructura de las instancias de BPP, los índices que describen el comportamiento del algoritmo y los índices que cuantifican el desempeño final. Las relaciones son obtenidas estudiando la matriz de correlación, analizando gráficas y aplicando inferencia causal. Inicialmente los análisis fueron realizados sobre el conjunto total de instancias (1,668 casos definidos en la Sección 3.2.2), para obtener información general de las relaciones de desempeño que definen el comportamiento del algoritmo HGGA-BP. El análisis de la matriz de correlación mostró que, principalmente, cuatro índices de caracterización del problema se encuentran asociadas con el índice *Función de Auto-correlación*, ρ_1 , que describe la asociación entre el conjunto de mejores soluciones generadas en el proceso evolutivo del

algoritmo. Estos índices son: t , d , $Rango$ y $Mayor$. La correlación existente entre cada índice es mostrada en la Tabla 4.5. Adicionalmente, en la Figura 4.14 se incluyen las gráficas de dispersión entre cada par de variables, así como el $Radio_teórico$ de las soluciones finales del algoritmo. En cada gráfica se aprecia la forma en que cada característica del problema parece afectar en la trayectoria generada por las mejores soluciones del algoritmo.

Tabla 4.5 Correlación entre ρ_1 y cuatro índices de caracterización de BPP.

Índices		Correlación
ρ_1	t	0.53043
ρ_1	d	0.46751
ρ_1	$Rango$	0.46663
ρ_1	$Mayor$	0.52167

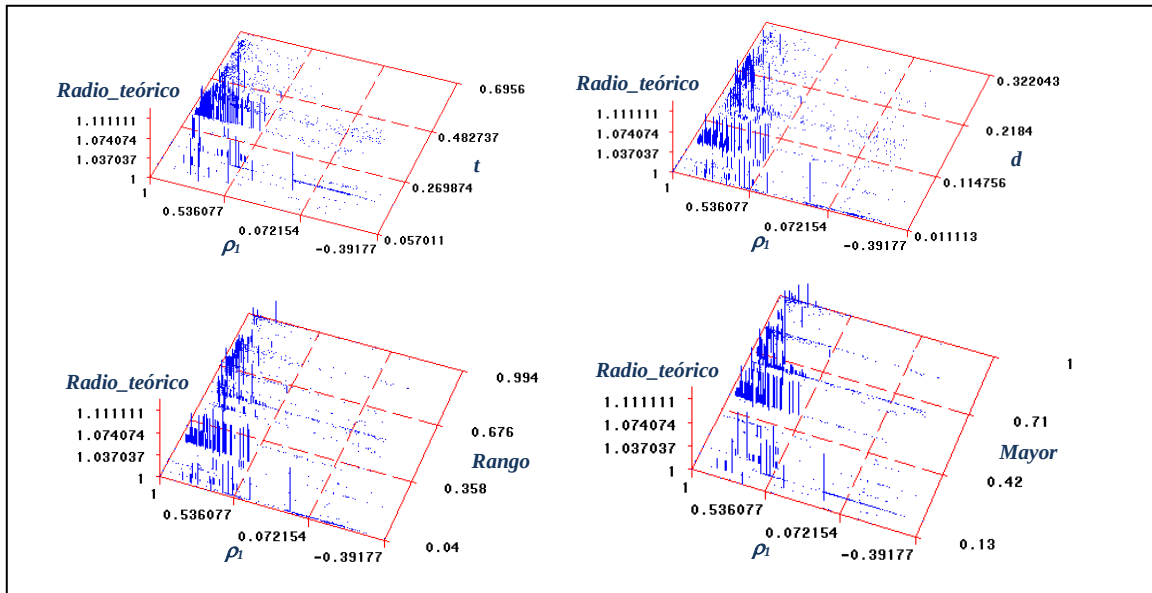


Figura 4.14 Gráficas de dispersión de índices de BPP y de comportamiento de HGGA-BP.

Las relaciones obtenidas muestran que, las asociaciones que puedan existir entre las mejores soluciones del algoritmo son causadas por la tendencia central y la variabilidad de los pesos de objetos, y que el tamaño de los objetos tiene un impacto importante en el comportamiento algorítmico. Del análisis de las gráficas se puede concluir que:

- Las instancias con *objetos más grandes* (un valor de t alto), parecen ser las más *fáciles* para el algoritmo y, para este tipo de instancias, la correlación entre el conjunto de soluciones generadas por el algoritmo es alta.
- Las instancias con *objetos más pequeños* (un valor de t bajo), parecen ser *fáciles*, aunque hay algunas instancias para las que el algoritmo no logra encontrar la solución óptima. Para este tipo de instancias, la mayoría de los casos, el óptimo es encontrado cuando las mejores soluciones generadas por el algoritmo no están muy asociadas entre sí.
- Las instancias con *objetos medianos* ($t \approx 0.4$), son más *difíciles*. Para este tipo de instancias, el desempeño del algoritmo parece ser pobre cuando la correlación entre las mejores soluciones es alta.
- Una mayor variabilidad en los pesos de los objetos (valores de d y *Rango* altos) parece beneficiar el desempeño del algoritmo y la correlación existente entre las mejores soluciones no parece impactar, en este caso, la solución obtenida.
- Las instancias con menor variabilidad (valores de d y *Rango* bajos) son más *difíciles*. El algoritmo obtiene buenas soluciones cuando la correlación entre las mejores soluciones es baja.

Los resultados obtenidos parecen revelar que para las instancias con tamaños de pesos medianos el algoritmo no alcanza buenos resultados cuando las mejores soluciones se encuentran muy asociadas entre sí, lo que indica que, posiblemente, el algoritmo se estanca en óptimos locales. El mismo comportamiento ocurre con algunas instancias que incluyen objetos pequeños. En ambos casos, las relaciones entre el mal desempeño y el mayor grado de asociación entre las mejores soluciones parecen acentuarse cuando la variabilidad en los pesos de los objetos es mediana y baja.

Por su parte, la variabilidad entre las aptitudes de las mejores soluciones generadas por el algoritmo durante el proceso de solución, caracterizadas por el índice *Desv_Mejor*

también están relacionadas con índices de tendencia central y dispersión. Las medidas que tienen mayor impacto sobre el grado de variación de las mejores soluciones son: t , d , $Rango$, $Coeficiente_variación$, $Mayor$ y b . La Tabla 4.6 y la Figura 4.16 muestran el grado y la forma de relación lineal existente entre cada par de variables.

Tabla 4.6 Correlación entre $Desv_Mejor$ y seis índices de caracterización de BPP.

	Índices	Correlación
$Desv_Mejor$	t	-0.60287
$Desv_Mejor$	d	-0.60631
$Desv_Mejor$	$Rango$	-0.60579
$Desv_Mejor$	$Coeficiente_variación$	-0.54876
$Desv_Mejor$	$Mayor$	-0.63186
$Desv_Mejor$	b	0.57900

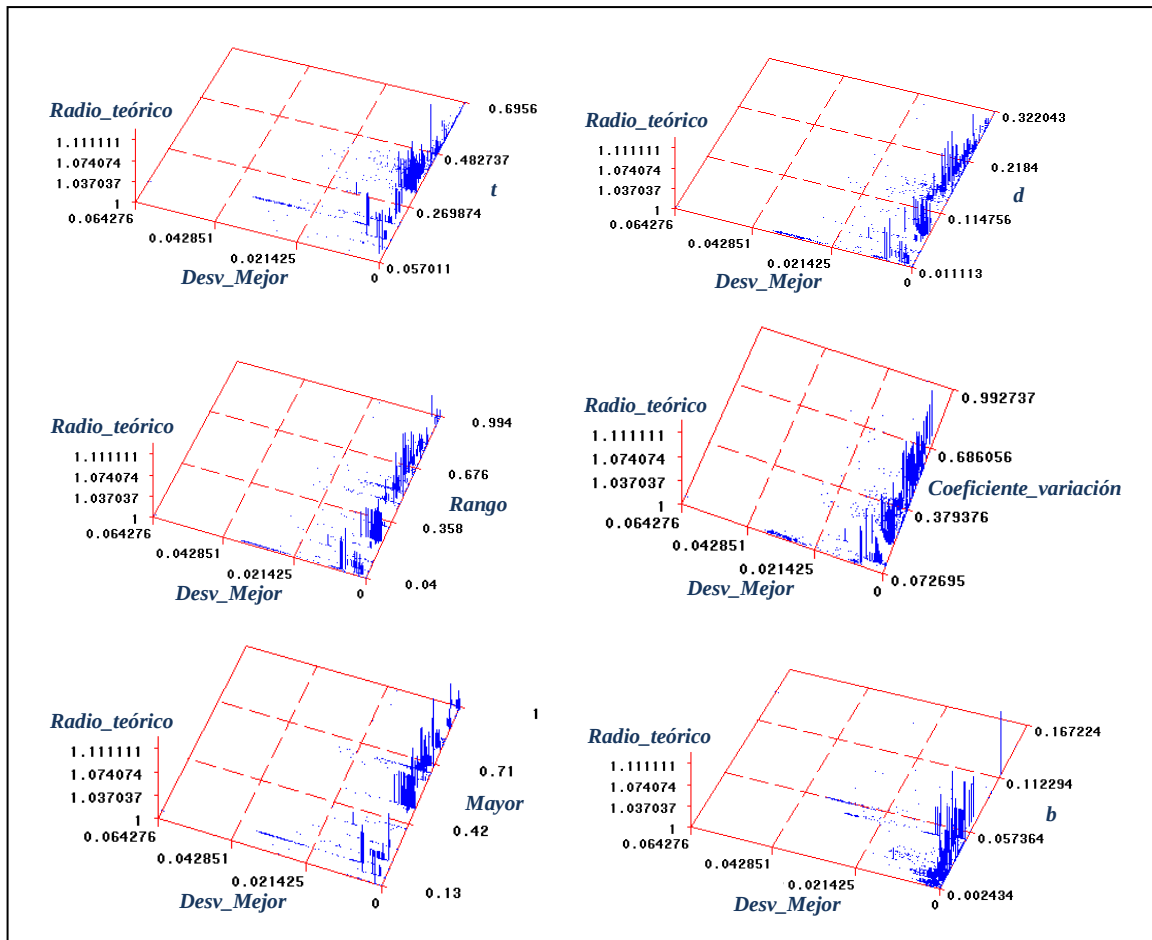


Figura 4.15 Gráficas de dispersión de índices de BPP y de comportamiento de HGGA-BP.

En la Figura 4.15 se puede apreciar la forma de cada relación entre la estructura de las instancias de BPP y la variabilidad en las mejores soluciones. Las conclusiones que pueden obtenerse de las relaciones encontradas son:

- Una vez más, puede observarse que los problemas más fáciles son los que presentan instancias con objetos más grandes y una mayor variabilidad en los pesos de los objetos.
- Las instancias difíciles son solucionadas, cuando la variabilidad entre las mejores soluciones generadas es mayor, lo que indica que se exploraron más zonas del espacio de búsqueda.
- Mientras más grande es el problema en relación al tamaño del contenedor, esto es, mientras más bajo el valor de b , la instancia es más fácil, pues los objetos son grandes y/o son muchos y el espacio de soluciones es más pequeño o plano. La variabilidad en las mejores soluciones es baja pues no existen tantas “buenas” combinaciones para objetos grandes.

El estudio de las relaciones entre estos índices permitió comprobar que en las instancias “difíciles” que no pudieron ser solucionadas (objetos con tamaños medianos y pequeños) el algoritmo parece estancarse en una zona del espacio de búsqueda. Lo anterior, sugiere que las estrategias incluidas en el algoritmo para generar y modificar soluciones, no permiten, en la solución de instancias “difíciles”, explorar y/o explotar diferentes zonas del espacio de búsqueda para salir de áreas con óptimos locales.

El análisis del índice de variabilidad de la población *Desv_Aptitud*, confirma que existen ciertas inconsistencias en el comportamiento del algoritmo HGGA-BP que revelan la falta de exploración del espacio de búsqueda. El promedio de variabilidad en la aptitud de los individuos de la población también depende de la tendencia central y variabilidad en los pesos de los objetos. Esta medida es impactada, principalmente, por los valores de

t , d , $Rango$, $Coeficiente_variación$ y $Mayor$. La Figura 4.16 y la Tabla 4.6 detallan la intensidad y la forma de cada relación.

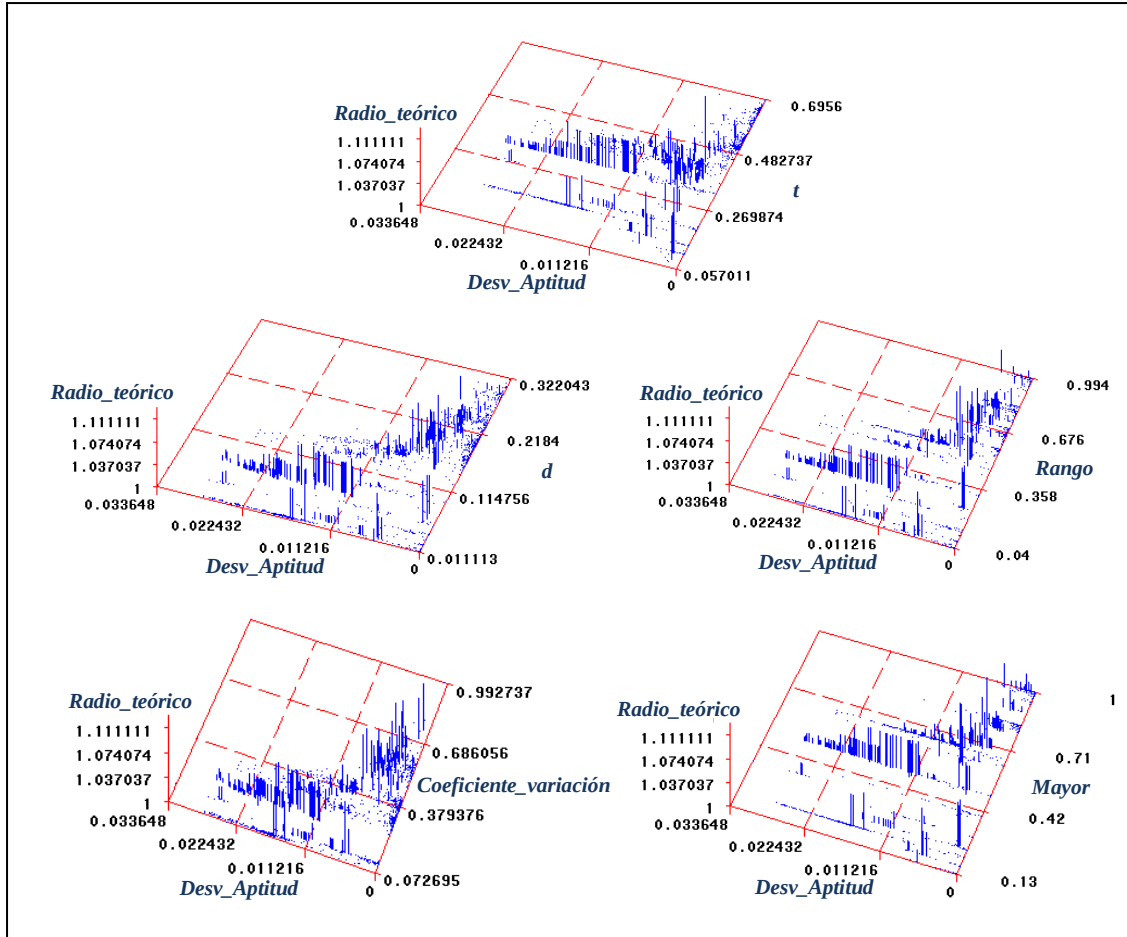


Figura 4.16 Gráficas de dispersión de índices de BPP y de comportamiento de HGGA-BP.

En la Figura 4.16, se confirma que, para el metaheurístico estudiado, las instancias con objetos grandes son las más fáciles. También, vuelve a observarse que, en algunos casos, la diversificación del conjunto de soluciones que conforman la población, ayuda al algoritmo a encontrar mejores soluciones. Las gráficas permiten identificar un detalle muy importante del comportamiento algorítmico: *Para las instancias “difíciles”, un mayor grado de variabilidad en la aptitud de los individuos que conforman la población no parece ayudar a obtener la solución óptima.* La observación anterior puede ser clave en la identificación de áreas de mejora en la estructura del algoritmo.

Tabla 4.6 Correlación entre *Desv_Aptitud* y cinco índices de caracterización de BPP.

	Índices	Correlación
<i>Desv_Aptitud</i>	<i>t</i>	-0.51759
<i>Desv_Aptitud</i>	<i>d</i>	-0.59671
<i>Desv_Aptitud</i>	<i>Rango</i>	-0.59886
<i>Desv_Aptitud</i>	<i>Coeficiente_variación</i>	-0.53074
<i>Desv_Aptitud</i>	<i>Mayor</i>	-0.56648

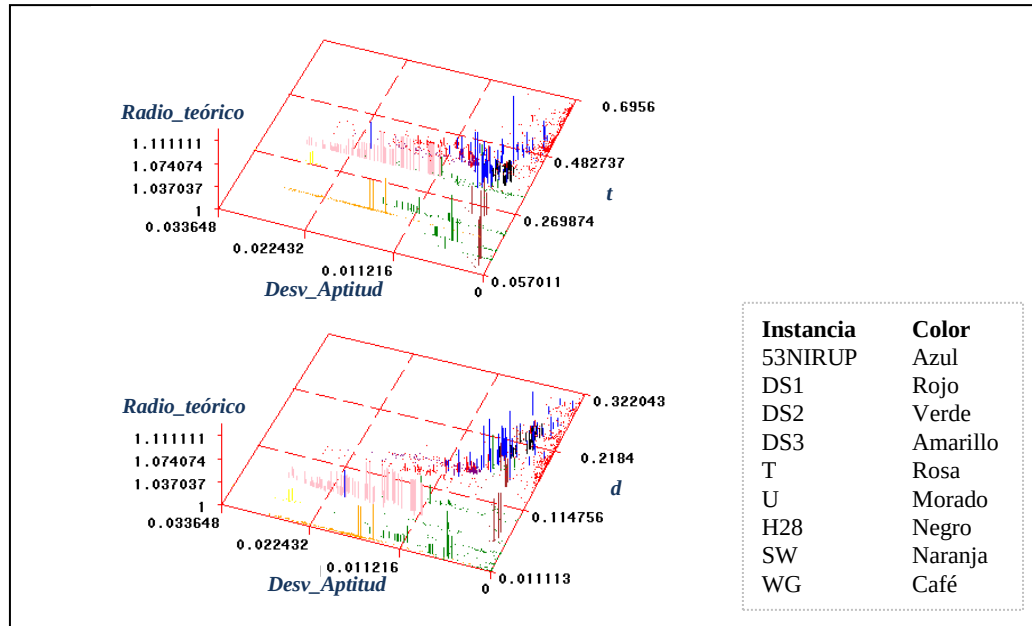


Figura 4.17 Gráficas de dispersión de índices de BPP y de comportamiento de HGGA-BP.

En la Figura 4.17 se muestran las gráficas de dispersión para los índices *Desv_Aptitud*, *t* y *d*, destacando los diferentes conjuntos de instancias. En ella puede observarse que los conjuntos de instancias que presentan mayor grado de dificultad para el algoritmo son las T, las H28, las DS3 y las 53NIRUP. Los demás conjuntos incluyen también algunas instancias que no pudieron ser resueltas de manera óptima. La dificultad de estas instancias podría ser causada por el tamaño del problema, la forma de la distribución de los pesos, la multiplicidad, o por otros factores que aún no han sido caracterizados. Lo anterior fue comprobado al realizar un análisis interno de las relaciones en cada conjunto de instancias.

Para *instancias uniformes* y con tamaños de *objetos medianos* (U, c3w2 y c3w4), con peso mayor cercano al 70% de la capacidad del contenedor ($Mayor \approx 0.7$ y $t \approx 0.4$), el aumento del número de objetos (p) y la *Multiplicidad* causan (ver Figura 4.18): 1) que la correlación del conjunto de mejores soluciones generadas por el algoritmo (ρ_1) aumente; 2) que la variabilidad en las aptitudes de las mejores soluciones (índice *Desv_Mejor*) disminuya; y 3) que la variabilidad en las aptitudes de la población (*Desv_Aptitud*) disminuya.

Para *instancias tripletas* (en promedio tres objetos llenan el contenedor), con tamaños medianos con peso promedio $t = 0.33$ (T, w1b1 y w1b2), el desempeño del algoritmo es afectado por el tamaño del problema, la multiplicidad y la forma de la distribución de las frecuencias de los pesos de los objetos (ver Figura 4.18). Un aumento en el tamaño del problema (p) causa: 1) un aumento en ρ_1 ; 2) una disminución en *Desv_Mejor*; y 3) un aumento en *Desv_Aptitud*. Por su parte, una menor *uniformidad* en los pesos de los objetos causa 1) que grado de correlación ρ_1 aumente; 2) que *Desv_Mejor* sea menor; y 3) que el valor de *Desv_Aptitud* aumente. Un aumento en la *asimetría* de la distribución de pesos (sesgo positivo indica que existen más objetos por debajo de la media, lo que caracteriza a las instancias del grupo T), causa: 1) que ρ_1 aumente; 2) que *Desv_Mejor* disminuya; y 3) que *Desv_Aptitud* aumente.

Para *instancias uniformes* y con objetos de *pesos pequeños* (w2b1, w2b2, w3b1, w3b2, w3b3, w4b1, w4b2, DS3, Was 1 y Was 2), con tendencia central $t < 0.3$, la correlación entre las mejores soluciones no parece ser afectada por ninguna propiedad del problema. Las características que tienen mayor impacto en el comportamiento algorítmico parecen ser el tamaño del problema (p) y la *Multiplicidad*. Un aumento en sus medidas ocasiona 1) que la desviación entre los valores de las mejores soluciones disminuya; y 3) que la variabilidad en las aptitudes de la población disminuya (ver Figura 4.18).

La Figura 4.18, muestra gráficas de dispersión que ilustran el comportamiento mostrado por el algoritmo con los tipos de instancias descritas en los párrafos anteriores. Una vez

más, se confirmó que para las instancias que no pueden ser resueltas, el algoritmo parece estancarse en ciertas zonas del espacio de búsqueda. En la figura, las instancias para las que se encontró el óptimo se marcaron en color azul y las más “difíciles” están marcadas con color negro. En el Capítulo 5 se profundiza en el estudio de las instancias más difíciles y se proponen dos nuevos índices de caracterización para BPP.

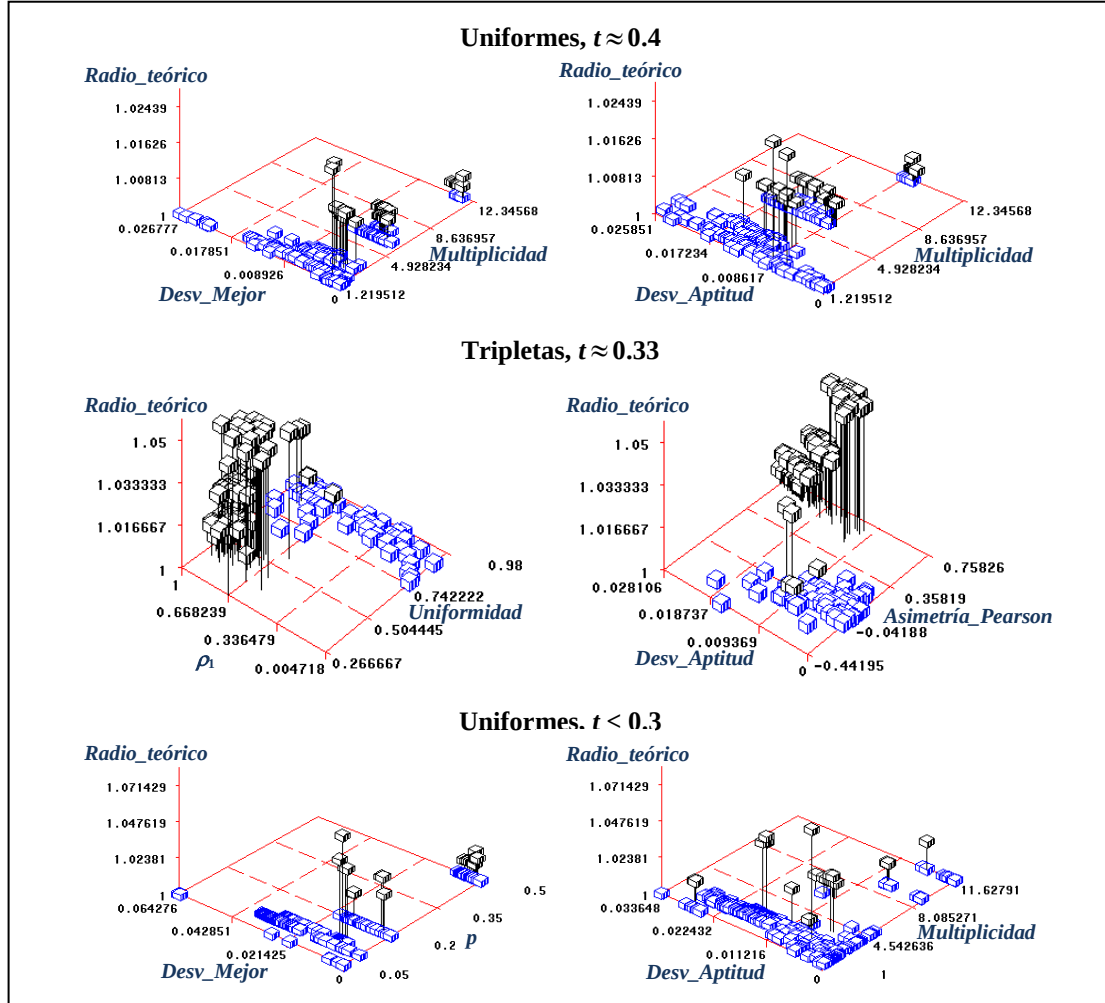


Figura 4.18 Gráficas de dispersión de índices de BPP y de comportamiento de HGGA-BP.

El estudio de las relaciones entre los índices de caracterización demostró que las principales características que reflejan el grado la dificultad de una instancia de BPP son la tendencia central y la variabilidad de los pesos de los objetos. Para las instancias que

presentan valores parecidos de estas medidas, son la forma y multiplicidad del conjunto de pesos los que definen la complejidad. El análisis del comportamiento algorítmico evidenció que las estrategias incluidas en el metaheurístico híbrido HGGA-BP, aunque presentan muy buenos resultados para algunas instancias (aquellas con objetos grandes) no parecen ser las adecuadas para tratar instancias con objetos medianos y pequeños. Para las instancias “difíciles”, las estrategias de generación de individuos no permiten agregar diversidad en los individuos de la población y los operadores genéticos de cruce y muta no conducen al algoritmo a nuevas regiones.

La Figura 4.19 muestra el modelo obtenido al realizar el análisis causal del proceso de optimización del algoritmo HGGA-BP. El análisis fue realizado sobre las calificaciones de los componentes principales para el problema y el comportamiento algorítmico (ver Sección 4.3.3) y los índices del desempeño final. En este grafo se confirma que los factores que tienen un impacto directo en la calidad de la solución final obtenida por el algoritmo son aquellos que definen la estructura de las instancias de BPP. Así mismo, se observa que el tiempo de ejecución se encuentra ligado directamente con el tamaño del caso a resolver; y que el número de generaciones requeridas por el metaheurístico para encontrar la mejor solución depende de las características del conjunto de individuos que conforman la población durante el proceso evolutivo.

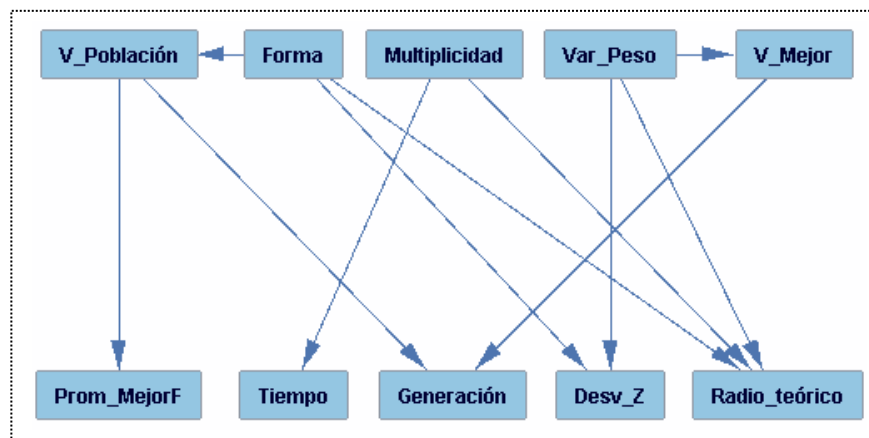


Figura 4.19 Diagrama causal del proceso de optimización de BPP con el algoritmo HGGA-BP.

4.3 EXPLICACIONES DEL COMPORTAMIENTO ALGORÍTMICO

En esta sección, las principales estrategias del algoritmo HGGA-BP son examinadas, identificando las causas del comportamiento algorítmico y proponiendo áreas de mejora. Los conocimientos obtenidos fueron utilizados para rediseñar la estructura del algoritmo y obtener una nueva versión que exhibe una contundente mejora en el desempeño. Los resultados obtenidos demuestran la aportación del enfoque de análisis experimental de algoritmos propuesto en este trabajo. La nueva versión del algoritmo fue reportada en el artículo “Heurísticas de Agrupación Híbridas Eficientes para el Problema de Empacado de Objetos en Contenedores” publicado en la revista *Computación y Sistemas*, año 2012, vol. 16, no. 3, p. 349-360 [Cruz-Reyes12].

En la sección anterior, las relaciones de desempeño obtenidas del análisis experimental del algoritmo HGGA-BP evidenciaron que las estrategias incluidas en él, para creación de soluciones y exploración del espacio de búsqueda, presentaban ciertas inconsistencias y no le permitían salir de óptimos locales y explorar otras zonas del espacio de búsqueda (ver Sección 4.2.3). A continuación se muestran las principales conclusiones obtenidas del análisis de las estrategias que definen la estructura y comportamiento del algoritmo y se resumen los cambios realizados en los procedimientos que tienen mayor impacto en el proceso de exploración del algoritmo: 1) generación de individuos y 2) sustitución y conservación de soluciones. Para un mayor detalle sobre la nueva versión del algoritmo ver el artículo mencionado en el párrafo anterior.

4.3.1 GENERACIÓN DE INDIVIDUOS

Los métodos construcción de individuos de HGGA-BP (detallados en la Sección 4.2.1) consisten en: 1) calcular el tamaño inicial de la solución $LB = L_1$, 2) acomodar los LB objetos más pesados en diferentes contenedores y 3) almacenar el resto de los objetos con las heurísticas BF y B3F (descritas en la Sección 2.4).

Al analizar el comportamiento del algoritmo, así como la aportación de los métodos para generación de individuos, se observó que las dos heurísticas utilizadas con este propósito ($LB+BF_Aleatorio$ y $LB+B3F_Aleatorio$) limitaban el espacio de búsqueda a cierta zona, al establecer desde el principio que los $LB = L_1$ objetos más grandes debían acomodarse en contenedores separados. Aunque en la mayoría de los casos los operadores evolutivos de cruzamiento y mutación permitían explorar otras zonas del espacio de búsqueda, para las instancias que resultaron “difíciles”, no existía la suficiente exploración para salir de los óptimos locales.

Las características de las instancias “difíciles” (objetos medianos y pequeños) revelaron que en muchas de ellas, separar los L_1 objetos más “grandes” es contraproducente, dado que los tamaños de estos objetos son en realidad “pequeños”, tomando en cuenta su relación con la capacidad del contenedor c . Por ejemplo, para una instancia de la clase Triplets, con $n = 501$, $c = 100$ y pesos distribuidos entre $[25, 50]$, el límite inferior de contenedores es $L_1 = 167$, por lo que en todas las soluciones generadas los 167 objetos más grandes eran acomodados cada uno en un contenedor. Como puede observarse, para esta instancia, separar los 167 objetos no tiene ningún sentido; estos objetos pueden combinarse entre sí, y algunos tal vez deberían estar juntos en la solución óptima. El estudio de las características de las instancias permitió concluir que la idea de separar los L_1 objetos más grandes no era una técnica de reducción apropiada, pues en lugar de beneficiar el desempeño del algoritmo, limitaba su poder de exploración.

La estrategia de generación de individuos fue modificada cambiando el valor del límite inicial de contenedores LB , por el valor de un nuevo límite, $LB = \tilde{n}$, que es igual al número de objetos de tamaño mayor que la mitad de la capacidad del contenedor. Con esto se garantiza acomodar los objetos realmente grandes y no se limita la exploración del espacio de búsqueda. En el Capítulo 5, se analiza el impacto del número de objetos grandes $\tilde{n}$ en la dificultad de las instancias de BPP y se propone un nuevo índice de caracterización relacionado con la estructura de las soluciones de las instancias. En el Capítulo 6 se propone un nuevo método de generación de individuos que incorpora

conocimiento sobre la estructura del problema para mejorar la efectividad de un nuevo algoritmo genético para BPP.

4.3.2 SUSTITUCIÓN Y CONSERVACIÓN DE INDIVIDUOS

En cada generación de HGGA-BP se construyen tantos individuos como el tamaño de la población y estos individuos tienen dos posibilidades de ser aceptados y formar parte de la población; una es que su aptitud sea mejor que la del individuo que ocupa esa posición y otra es que el valor de un número aleatorio $U[0, 1]$ sea mayor que 0.95 (ver Sección 4.2.1 y Algoritmo 4.1).

En la Sección 4.2.2 se mostró un diagrama causal en el que parecía que el número de individuos *Nuevos* que el algoritmo HGGA-BP agrega a la población en cada generación no tenía ninguna relación con el desempeño algorítmico. Lo anterior pudo comprobarse al analizar los índices que caracterizan el proceso de generación de individuos (ver Sección 4.2.1). La Figura 4.20 presenta una gráfica del comportamiento del algoritmo durante el proceso evolutivo; esta figura muestra, para cada generación (eje horizontal), el número de individuos *Nuevos*, *Buenos* y *Malos* que son agregados a la población (eje vertical). Este comportamiento se presenta, en general, para todas las instancias de BPP.

En la Figura 4.20 se observa que, la primera generación (en la cual se crea la población inicial) es la única que agrega diversidad a la población. En las generaciones posteriores, de los pocos individuos que son aceptados, la mayoría son considerados *Malos* pues su aptitud es peor que la de los individuos que sustituyen. Por lo que se puede concluir que el método de generación individuos no permite introducir soluciones con características variadas, que lleven al algoritmo a explorar diversas regiones del espacio de búsqueda. Los pocos individuos que sí son introducidos en cada generación, son aquellos que caen en el 5% de probabilidad de ser aceptados sin considerar su aptitud (ver Algoritmo 4.1).

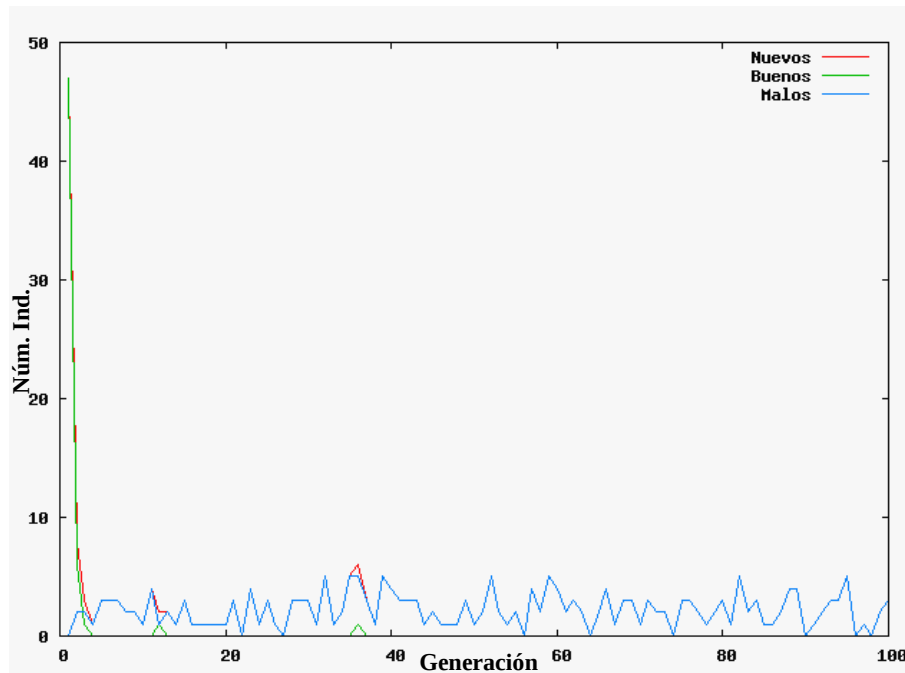


Figura 4.20 Individuos nuevos, agregados a la población en cada generación por HGGA-BP.

El análisis del comportamiento algorítmico reveló que la creación de nuevos individuos en cada generación era un trabajo adicional que aportaba muy poca ganancia a la calidad de las soluciones y desempeño del algoritmo. De los nuevos individuos creados en cada generación, en promedio, menos del 5% eran incluidos en la población, significando una pérdida de tiempo la generación del resto de los individuos. Por otra parte, el estudio de relaciones entre las aptitudes de los individuos en cada generación reveló que, después de las primeras cinco generaciones de evolución de HGGA-BP, la población comenzaba a incluir una gran cantidad de individuos con aptitudes repetidas; esta falta de diversidad conducía al algoritmo a estancarse en óptimos locales.

La estrategia de sustitución y conservación de individuos fue modificada para generar una población completa de individuos nuevos solamente en la primera generación del proceso evolutivo. Así mismo, para agregar diversidad a la población, al final de cada generación se introducen nuevos individuos que sustituyen los individuos con aptitud repetida.

4.3.3 COMPARACIÓN DE HGGA-BP ORIGINAL Y REDISEÑADO

La aplicación del conocimiento obtenido en el análisis de las características del proceso de optimización de BPP con HGGA-BP, permitió obtener una mejora significativa en el desempeño de dicho algoritmo. Con el fin de demostrar la robustez de HGGA-BP, para cada instancia de prueba se realizaron 30 corridas del algoritmo con diferentes semillas iniciales de números aleatorios. Los valores reportados para cada instancia representan los promedios de las 30 soluciones generadas. Los resultados finales son mostrados en la Tabla 5.2, donde se incluyen el número de óptimos, tiempo promedio (en segundos) y el número de generaciones iteradas por el algoritmo para encontrar la mejor solución. En esta tabla se presenta, además, el número de óptimos obtenido por los algoritmos HI-BP [Alvim04] y Perturbation-SAWMBS [Fleszar11], que son los mejores algoritmos del estado del arte.

Tabla 5.2 Resultados mejorados para el algoritmo GGA-BP.

Instancias	Inst.	HI-BP	Pert.- SAWMBS	HGGA-BP Original			HGGA-BP Mejorado		
				Óptimos	Tiempo	Gen	Óptimos	Tiempo	Gen
U	80	80	79	52	8.63	47	79	1.67	13
T	80	80	80	0	1.90	69	80	5.14	53
Data Set 1	720	720	720	692	2.43	43	718	2.67	19
Data Set 2	480	480	480	450	0.57	10	480	0.90	6
Data Set 3	10	10	10	8	1.75	8	9	8.10	77
Was 1	100	100	100	99	0.05	6	100	0.04	3
Was 2	100	100	100	98	0.34	28	100	0.99	32
Gau 1	17	12	16	9	0.59	54	15	1.92	40
Hard28	28	5	5	5	1.63	90	8	6.75	87
Total	1615	1587	1590	1413	2.01	44	1589	3.13	42
Efectividad		0.982	0.984	0.874			0.983		

Los resultados experimentales obtenidos por la nueva versión de HGGA-BP demuestran la validez y aportación del conocimiento obtenido producto del enfoque de análisis del proceso de optimización propuesto. La efectividad de HGGA-BP fue mejorada de un 87.4% a un 98.3%. Así mismo, fue posible superar los resultados de los dos mejores algoritmos del estado del arte en la clase de instancias Hard28, al encontrar la solución

óptima de 3 instancias de prueba para las que estos algoritmos utilizan un contenedor extra: hBPP640, hBPP531 y hBPP814.

4.4 CONCLUSIONES DEL ESTUDIO

El estudio presentado en este capítulo demostró la importancia de la caracterización y análisis del proceso de optimización de BPP, los conocimientos ganados permitieron explicar la dificultad de las instancias de prueba, así mismo fue posible comprender y mejorar el comportamiento del algoritmo objeto de estudio. Es importante destacar que la calidad y aplicabilidad del conocimiento obtenido dependen principalmente del poder de caracterización y de la validez de los índices usados en el análisis. La caracterización del proceso de optimización de algoritmos heurísticos es un problema desafiante, que requiere un amplio dominio del problema, del algoritmo y de las estrategias de análisis utilizadas, todo esto, combinado con una gran creatividad y compromiso por parte del investigador.

Por otro lado, aún existen instancias de BPP, como las Hard28, para las cuales no ha sido posible explicar el comportamiento de los algoritmos de solución, por lo que se considera que es necesario diseñar nuevos índices de caracterización, que permitan capturar otras propiedades, capaces de discriminar las instancias con mayor grado de dificultad. En el Capítulo 5, las instancias más difíciles son analizadas y se proponen tres nuevos índices de caracterización, luego se generan conjuntos de instancias con un alto grado de dificultad. Finalmente, en el Capítulo 6, todo el conocimiento adquirido en esta investigación es reunido para diseñar un algoritmo genético para BPP que incorpora estrategias de solución inteligentes y propiedades del dominio del problema, superando la efectividad de los algoritmos del estado del arte.

Capítulo 5

DIFICULTAD EN INSTANCIAS DE BPP: NUEVOS ÍNDICES Y CASOS DE PRUEBA

En este capítulo se analizan las instancias de BPP que han resultado ser más difíciles para los algoritmos del estado del arte. Con el objetivo de identificar las propiedades que impactan en el grado de dificultad de dichas instancias, se generan nuevos conjuntos de instancias de prueba y se analizan en detalle sus características y las de sus soluciones. Como producto de este estudio, se definen tres nuevos índices de caracterización para la estructura del problema y de las soluciones óptimas. Los índices propuestos permiten describir las instancias más difíciles de BPP. Finalmente, se sugieren nuevos conjuntos de instancias de prueba, para evaluar el desempeño de los algoritmos del estado del arte.

5.1 INSTANCIAS ESTÁNDAR DIFÍCILES

Con el objetivo de identificar las características que marcan el grado de dificultad de las instancias de BPP, se evaluó el desempeño de las heurísticas deterministas FFD, BFD y WFD (descritas en la Sección 2.4), así como el procedimiento WABP [Loh08] (descrito en la Sección 3.2.1) y la nueva versión del metaheurístico HGGA-BP sobre el conjunto de instancias estándar. Para cada instancia se realizó solamente una corrida del algoritmo HGGA-BP Mejorado, que de aquí en adelante será llamado HGGA-BP2, con la semilla inicial de números aleatorios fijada en 1.

La Tabla 5.1 muestra, el porcentaje de soluciones óptimas encontradas en cada clase de instancias para cada algoritmo de solución. De la tabla puede observarse, que las clases que incluyen a las instancias que coinciden en ser difíciles para todos los algoritmos son: Triplets, Gau 1 y Hard28. Las instancias Data Set 3, Was 1 y Was 2 fueron sugeridas por su alto grado de dificultad para las heurísticas deterministas, las cuales presentan un bajo desempeño sobre instancias con objetos de pesos distribuidos entre el 15% y 50% de la capacidad del contenedor [Schwerin97], por lo que no es extraño que ninguna de las tres heurísticas pueda resolverlas de manera óptima.

Tabla 5.1 Resultados experimentales de heurísticas deterministas, HGGA-BP2 y WABP.

Clase	HGGA-BP2 %opt.	WABP %opt.	FFD %opt.	BFD %opt.	WFD %opt.
Uniform	0.98	0.88	0.07	0.07	0.02
Triplets	0.75	0	0	0	0
Data Set 1	0.99	0.97	0.75	0.75	0.68
Data Set 2	1.00	0.97	0.49	0.49	0.46
Data Set 3	0.90	0.90	0	0	0
Was 1	1.00	1.00	0	0	0
Was 2	1.00	1.00	0	0	0
Gau 1	0.94	0.70	0.11	0.11	0.11
Hard28	0.35	0.17	0.17	0.17	0.07
Total	0.97	0.90	0.49	0.49	0.44

Adicionalmente, en la Tabla 5.2 se presentan los resultados detallados obtenidos por el algoritmo HGGA-BP2 sobre el conjunto Hard28. Esta tabla incluye, para cada instancia, el número de contenedores de la solución óptima y las soluciones mínima, máxima y promedio obtenidas por HGGA-BP2 en 30 corridas con diferentes semillas de números aleatorios. Se ha resaltado en color gris las cinco instancias que son resueltas de manera óptima por todos los procedimientos del estado del arte (hBPP14, hBPP359, hBPP716, hBPP119, hBPP175); en color azul las tres instancias en las que HGGA-BP2 supera a dichos algoritmos (hBPP640, hBPP531, hBPP814); así mismo se han resaltado en color púrpura tres instancias para las cuales el HGGA-BP2 logra obtener la solución óptima en alguna de las 30 corridas (hBPP360, hBPP742 y hBPP47).

Tabla 5.2 Resultados obtenidos por el algoritmo HGGA-BP2 en el conjunto Hard28.

Instancias	Óptimo	Min	Max	Prom
hBPP14	62	62	62	62
hBPP832	60	61	61	61
hBPP40	59	60	60	60
hBPP360	62	62	63	63
hBPP645	58	59	59	59
hBPP742	64	64	65	65
hBPP766	62	63	63	63
hBPP60	63	64	64	64
hBPP13	67	68	68	68
hBPP195	64	65	65	65
hBPP709	67	68	68	68
hBPP785	68	69	69	69
hBPP47	71	71	72	72
hBPP181	72	73	73	73
hBPP359	76	76	76	76
hBPP485	71	72	72	72
hBPP640	74	74	75	74
hBPP716	76	76	76	76
hBPP119	77	77	77	77
hBPP144	73	74	74	74
hBPP561	72	73	73	73
hBPP781	71	72	72	72
hBPP900	75	76	76	76
hBPP175	84	84	84	84
hBPP178	80	81	81	81
hBPP419	80	81	81	81
hBPP531	83	83	83	83
hBPP814	81	81	82	81

Al analizar los resultados obtenidos por HGGA-BP2 sobre las instancias del conjunto Hard28 (ver Tabla 5.2), es posible hacer una clasificación de su grado de dificultad. De esta forma se observó que existen: cinco casos “*muy fáciles*” (resaltados en color gris) resueltos de manera óptima por la mayoría de los algoritmos del estado del arte; tres casos de “*mediana dificultad*” (resaltados en color azul) que HGGA-BP2 resuelve de manera óptima en la mayoría de las corridas, superando a los mejores algoritmos; tres casos “*difíciles*” (resaltados en color púrpura) para los cuales HGGA-BP2 logra obtener la solución óptima en alguna de las 30 corridas, aunque no muestra un comportamiento robusto; las 17 instancias restantes son consideradas “*muy difíciles*” pues el algoritmo HGGA-BP2 no logra obtener la solución óptima en ninguna de las 30 corridas.

Como puede advertirse en los resultados experimentales, el desempeño de los algoritmos indica que existe una diferencia en el grado de dificultad en las instancias Hard28. Sin embargo, el conjunto de 14 índices para la caracterización de BPP, que fue estudiado en el Capítulo 4, no permite diferenciar entre este conjunto de 28 instancias (ver Tabla 4.4 y Figura 4.9) ¿Cuáles serán las características que marcan la diferencia en el grado de dificultad dentro de este conjunto de instancias? Para responder esta pregunta se realizó un análisis de la estructura de las soluciones óptimas de estas instancias.

5.2 NUEVO ÍNDICE DE CARACTERIZACIÓN DE BPP: *Gap*

El estudio de la estructura de las soluciones óptimas reveló que, para las instancias más difíciles, los contenedores de la solución están llenos casi al 100% de su capacidad. Esta propiedad puede caracterizarse mediante el índice *Gap_óptimo*, que mide qué tan llenos deben estar los contenedores en la solución óptima. La Ecuación 5.1 define este índice, donde m es el número de contenedores en la solución óptima, S es la suma de los pesos de todos los objetos y c es la capacidad de los contenedores.

$$Gap_óptimo = 1 - \frac{S}{mc} \quad (5.1)$$

El valor del índice *Gap_óptimo* puede variar entre $[0, 0.5)$ y determina cuánto espacio quedará libre en los contenedores de la solución óptima. El estudio experimental del desempeño de los algoritmos de solución de BPP reveló que: las instancias con valores altos de *Gap_óptimo* son fáciles de resolver, en contraste con las instancias con valores bajos para este índice, para las cuales el desempeño de los algoritmos es menos robusto.

La Figura 5.1 muestra gráficas de la relación entre los valores del índice *Gap_óptimo* y el desempeño de los algoritmos HGGA-BP2 y WABP sobre la clase Hard28. De manera complementaria, la Figura 5.2 muestra un marcador diferente para cada instancia Hard28 según el grado de dificultad que presenta para el algoritmo HGGA-BP2 (ver Tabla 5.2): cinco casos *muy fáciles* graficados con un rombo gris, tres casos de *mediana dificultad*

graficados con un círculo azul, tres casos *difíciles* graficados con un cuadro púrpura y 17 casos *muy difíciles* graficados con una línea roja.

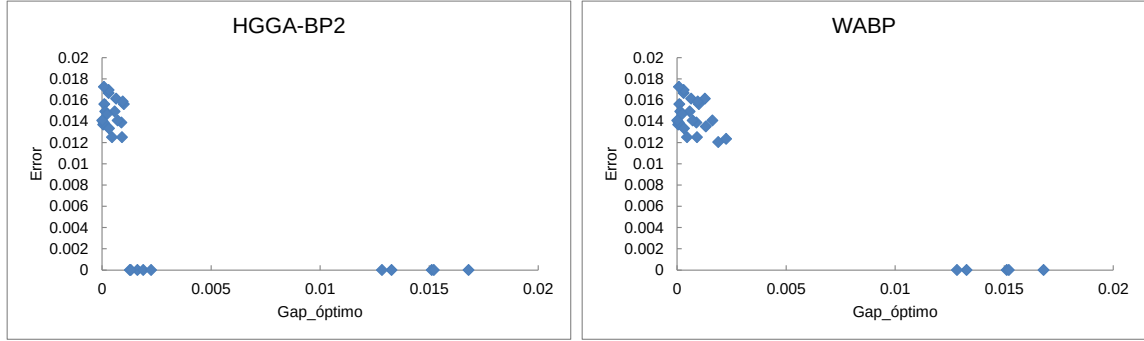


Figura 5.1 Desempeño de heurísticos y valor del índice $Gap_{\text{óptimo}}$ en instancias Hard28.

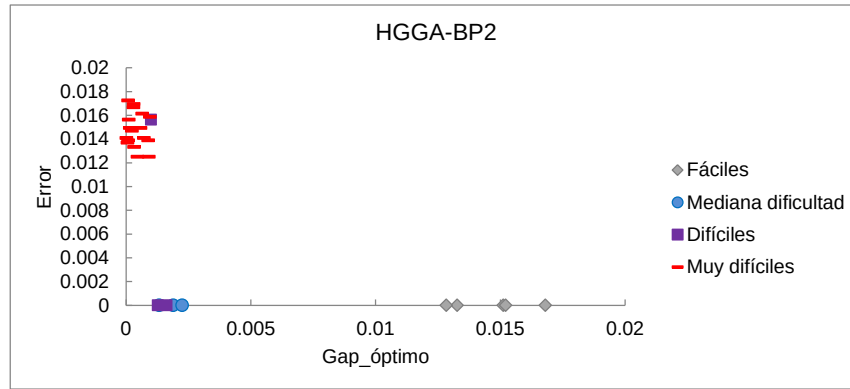


Figura 5.2 Desempeño de HGGA-BP2 y valor del índice $Gap_{\text{óptimo}}$ en instancias Hard28.

Al examinar los valores de $Gap_{\text{óptimo}}$, para todos los casos de prueba estándar, se observó que dicho índice permite diferenciar claramente la dificultad de las instancias de BPP. La Figura 5.3 muestra la gráfica de dispersión entre el valor del índice $Gap_{\text{óptimo}}$ y el error obtenido por los algoritmos HGGA-BP2 y WABP, para el conjunto total de 1615 instancias estándar. Los resultados experimentales obtenidos por los algoritmos del estado del arte de BPP confirman que las instancias con $Gap_{\text{óptimo}}$ cercano a cero son las más difíciles de resolver y que la dificultad de las instancias disminuye conforme los valores de $Gap_{\text{óptimo}}$ aumentan.

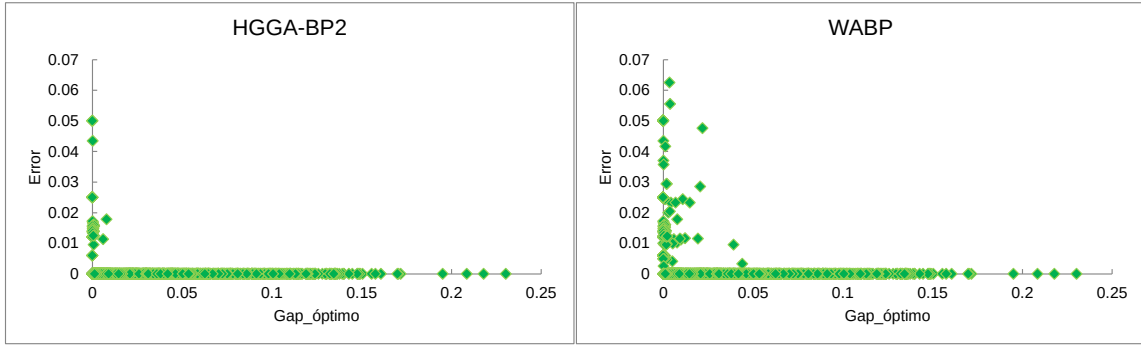


Figura 5.3 Desempeño de heurísticos y valor del índice *Gap_óptimo* en todas las instancias estándar.

Para una instancia nueva con un óptimo desconocido, se puede calcular el *Gap* esperado en la solución óptima, utilizando un límite inferior el número de contenedores, en este caso se propone utilizar el límite L_2 de Martello y Toth [Martello90b]. La Ecuación 5.2 define este índice, donde L_2 es el número de contenedores esperado a ser utilizado en la solución óptima, S es la suma de los pesos de todos los objetos y c es la capacidad de los contenedores. Debido al alto grado de exactitud del límite L_2 , en general, no se observan diferencias significativas en el comportamiento de los algoritmos para los valores de ambos índices.

$$Gap = 1 - \frac{S}{L_2 c} \quad (5.2)$$

5.3 NUEVAS INSTANCIAS DIFÍCILES TIPO HARD28

Con el objetivo de perfeccionar el proceso de caracterización de las instancias difíciles se generaron 9999 instancias con características similares a las Hard28. Cada instancia con $n = 160$ objetos, capacidad del contenedor $c = 1000$ y objetos con pesos w_i entre 1 y 800. La dificultad de las instancias fue evaluada al resolverlas con el algoritmo HGGA-BP2 en 30 corridas con diferentes semillas iniciales de números aleatorios. Dentro de este nuevo conjunto de instancias, fue posible identificar un grupo de 955 instancias para las que la solución del algoritmo no coincide con el óptimo teórico (límite inferior L_2), por lo que por ahora no puede comprobarse si el algoritmo logra o no obtener la solución

óptima. Las otras 9044 instancias se agruparon en cuatro clases para los que el algoritmo mostró un desempeño diferente:

1. Un grupo de 72 instancias *muy difíciles* para las cuales el algoritmo HGGA-BP2 no muestra un comportamiento robusto.
2. Un conjunto de 108 instancias *difíciles* cuya solución óptima es encontrada en la mayoría de las corridas del algoritmo.
3. Un grupo de 1724 instancias *fáciles* que son resueltas de manera óptima en todas las corridas del algoritmo.
4. Un conjunto de 7140 instancias *muy fáciles* que son resueltas óptimamente en la primera generación del algoritmo.

La Figura 5.4 muestra la gráfica de dispersión entre el valor del índice *Gap* y el número de soluciones óptimas obtenido para cada una de las 9044 instancias en 30 corridas del algoritmo HGGA-BP2. Una vez más puede comprobarse el poder de caracterización del índice propuesto, que permite identificar las instancias de BPP difíciles de resolver. La Figura 5.5 incluye una gráfica de dispersión entre el valor del índice *Gap* y el error obtenido por el algoritmo WABP para el nuevo conjunto de instancias. En ella puede evidenciarse la superioridad del algoritmo HGGA-BP2 sobre la estrategia WABP. Así mismo, una vez más se observa cómo la dificultad de las instancias tiende a aumentar cuando el valor del índice *Gap* disminuye.

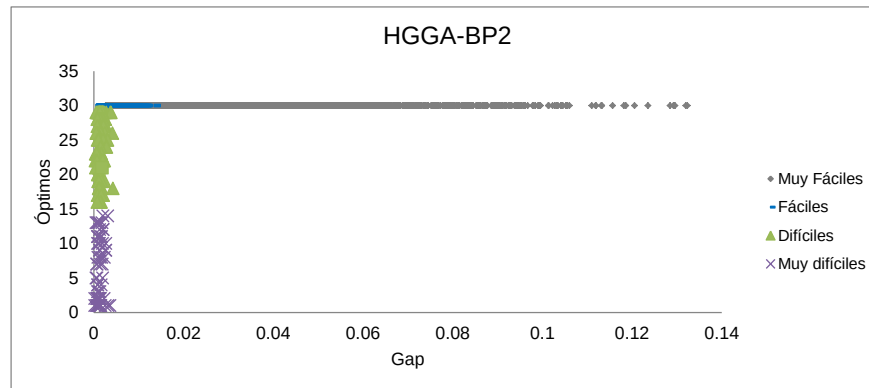


Figura 5.4 Desempeño de HGGA-BP2 y valor del índice *Gap* en 9044 instancias tipo Hard28.

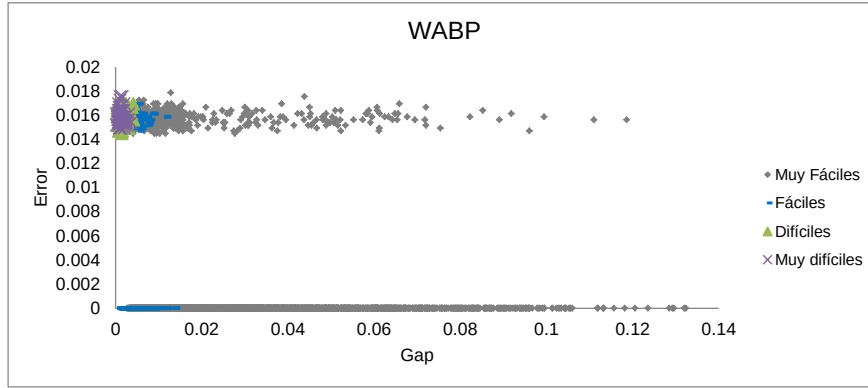


Figura 5.5 Desempeño de WABP y valor del índice *Gap* en 9044 instancias tipo Hard28.

Para demostrar que el comportamiento de los mejores algoritmos del estado arte con el nuevo conjunto de instancias, es similar al observado con la clase estándar Hard28, se seleccionaron de manera aleatoria, 13 instancias de cada uno de los cuatro grupos antes descritos. Las instancias fueron resueltas con el algoritmo HI_BP [Alvim04], obteniendo los siguientes resultados: 1) ningún óptimo en las 13 instancias *muy difíciles*; 2) ningún óptimo en las 13 instancias *difíciles*; 3) cuatro óptimos en las 13 instancias *fáciles*; y 4) todos los óptimos en las 13 instancias *muy fáciles*.

El análisis realizado permitió comprobar la superioridad del algoritmo HGGA-BP2 ante los mejores algoritmos del estado del arte al solucionar instancias similares a las Hard28, así como el poder de caracterización del índice *Gap*. Es evidente que los algoritmos del estado del arte no incluyen las estrategias adecuadas para resolver este tipo de instancias. La caracterización y análisis del proceso de optimización de dichos algoritmos podría ayudar a explicar el desempeño alcanzado por los mismos. En el Capítulo 6 se presenta un nuevo algoritmo que incorpora el conocimiento del impacto de la estructura de la solución óptima en la dificultad de las instancias de BPP.

5.4 NUEVO ÍNDICE DE CARACTERIZACIÓN DE BPP: *Pct_ñ*

Al analizar el comportamiento del algoritmo HGGA-BP (ver Sección 4.3.1) se concluyó que la estrategia de reducción utilizada por la heurística de generación de individuos

limitaba la exploración del espacio de búsqueda al establecer que los $LB = L_1$ objetos más grandes debían acomodarse en contenedores separados. Así mismo, se rediseñó el método de generación de individuos, cambiando el límite inicial de contenedores LB , por el valor de un nuevo límite, $LB = \tilde{n}$, que es igual al número de objetos de tamaño mayor que la mitad de la capacidad del contenedor. Con este cambio se logró mejorar el desempeño del algoritmo y se observó la importancia de considerar las características de la estructura las soluciones óptimas para perfeccionar el desempeño de los algoritmos.

Con el objetivo de caracterizar y analizar el impacto del número de objetos grandes $\tilde{n}$ en la dificultad de las instancias de BPP se propone el índice $Pct_{\tilde{n}}$. La Ecuación 5.3 define este índice, donde N incluye todos los objetos a acomodar, n es el número de objetos de N y $\tilde{n}$ es el número de objetos con peso $w_i > c/2$, $\forall i \in N$.

$$Pct_{\tilde{n}} = \frac{\tilde{n}}{n} \quad (5.3)$$

La Figura 5.6 muestra el desempeño de las heurísticas FF y FFD, las cuales consideran los objetos en orden aleatorio y decreciente de sus pesos, respectivamente. En la gráfica se puede observar como la dificultad de las instancias aumenta cuando $Pct_{\tilde{n}}$ disminuye y como FFD comienza a mostrar un peor desempeño que una estrategia aleatoria, debido a que cuando no existen objetos con peso mayor que $c/2$ el considerar a los objetos en orden decreciente de sus pesos, no parece beneficiar el desempeño algorítmico.

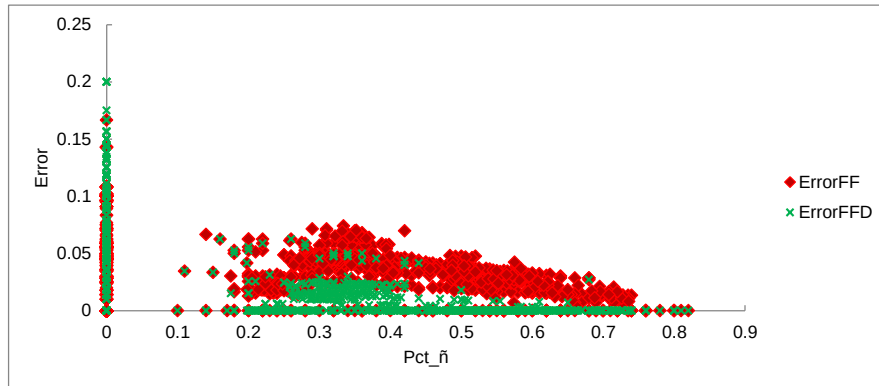


Figura 5.6 Desempeño de FF y FFD contra el valor del índice $Pct_{\tilde{n}}$ en instancias estándar.

La Figura 5.7 muestra el error de los algoritmos HGGA-BP2 y WABP contra los valores de $Pct_{\tilde{n}}$ de las instancias estándar. Aunque estos algoritmos muestran un desempeño más robusto que las heurísticas FF y FFD, una vez más puede observarse la tendencia a incrementar la dificultad cuando disminuye el porcentaje de objetos grandes. En esta figura, se ha destacado dentro de un círculo azul un grupo de instancias que incluyen a la clase Hard28, para la que se ha demostrado que su dificultad es causada principalmente por el valor bajo del índice $Gap_{\text{óptimo}}$. Los resultados experimentales confirman que un porcentaje bajo de objetos grandes ($Pct_{\tilde{n}} \leq 0.5$) suele incrementar la dificultad de las instancias.

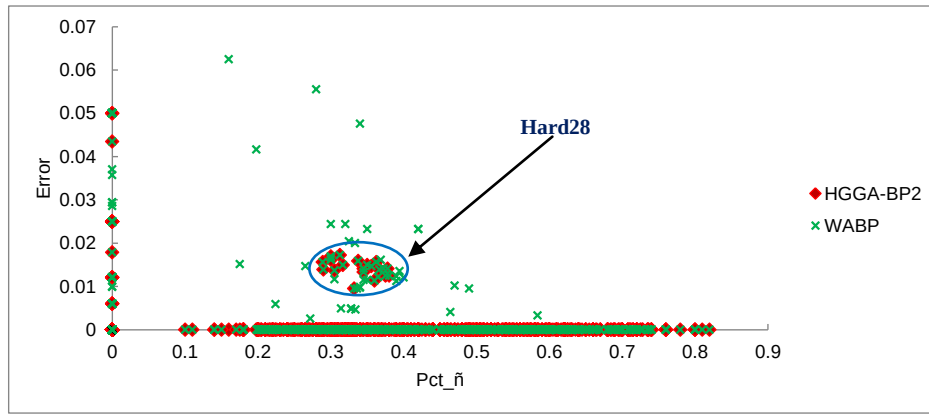


Figura 5.7 Desempeño de HGGA-BP2 y WABP contra el valor del índice $Pct_{\tilde{n}}$ en instancias estándar.

5.5 NUEVAS INSTANCIAS DIFÍCILES u-w1-wn

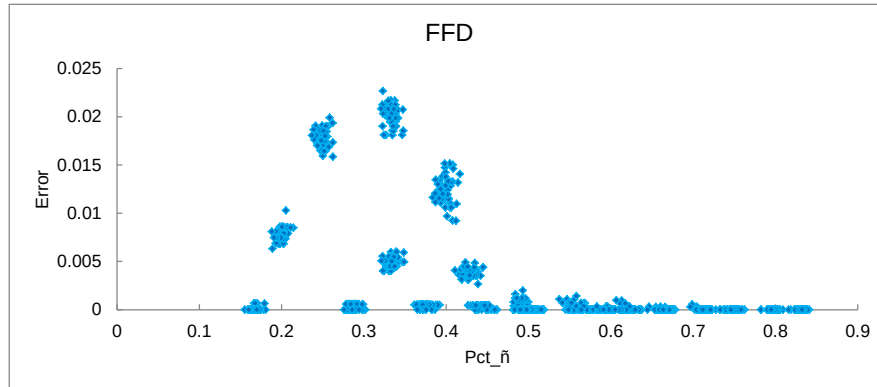
Para comprobar el impacto del porcentaje de objetos grandes $Pct_{\tilde{n}}$ en la dificultad de las instancias de BPP se generaron 25 clases de instancias, denominadas u-w1-wn, cada clase incluye 50 casos. Para todos los 1250 casos, el número de elementos es $n = 5000$ y la capacidad del contenedor es $c = 1000$. Para cada clase, dado un peso menor w_1 y un peso mayor w_n , los pesos de los objetos se generaron de manera uniforme en el intervalo $[w_1, w_n]$. La Tabla 5.3 muestra las características utilizadas para generar estas instancias, así como el número de *mejores soluciones* conocidas para cada clase de instancias, por la heurística FFD. Las *mejores soluciones* fueron calculadas considerando los resultados de los algoritmos HGGA-BP2, WABP, FFD, BFD y WFD.

Tabla 5.3 Características del conjunto u-w1-wn y número de mejores soluciones obtenidas por FFD.

$c = 1000$	w_n									
	600		700		800		900		1000	
w_1	$Pct_ñ$	FFD	$Pct_ñ$	FFD	$Pct_ñ$	FFD	$Pct_ñ$	FFD	$Pct_ñ$	FFD
1	0.167	45	0.286	28	0.376	9	0.444	17	0.499	38
100	0.2	0	0.333	0	0.428	0	0.499	36	0.554	35
200	0.25	0	0.399	0	0.499	37	0.57	32	0.624	36
300	0.333	0	0.498	45	0.598	47	0.665	46	0.713	45
400	0.497	50	0.664	50	0.748	50	0.797	50	0.831	50
Total		95		123		143		181		204

La Figura 5.8 muestra una gráfica de dispersión del desempeño de la heurística FFD para los diferentes valores de $Pct_ñ$ de las nuevas instancias. El error fue obtenido considerando la mejor solución conocida de instancia. De esta figura se puede observar, que la dificultad de las instancias parece aumentar para porcentajes de objetos grandes entre $0.2 \leq Pct_ñ \leq 0.5$. La Figura 5.9 muestra el valor del índice Gap y su relación con el error de FFD. Una vez más se comprueba el poder de caracterización del índice Gap y la importancia de utilizar estrategias que permitan llenar al máximo cada contenedor.

En las Secciones 6.1.3 y 6.3.3 del siguiente capítulo, se demuestra la importancia del conocimiento sobre el impacto del porcentaje de objetos grandes en la dificultad de las instancias. Este conocimiento es aplicado para diseñar una nueva heurística inteligente para BPP. Así mismo, las instancias u-w1-wn son utilizadas para analizar el efecto de la heurística de empaqueo en el desempeño final de un nuevo algoritmo genético.


Figura 5.8 Desempeño de FFD contra el valor del índice $Pct_ñ$ en instancias u-w1-wn.

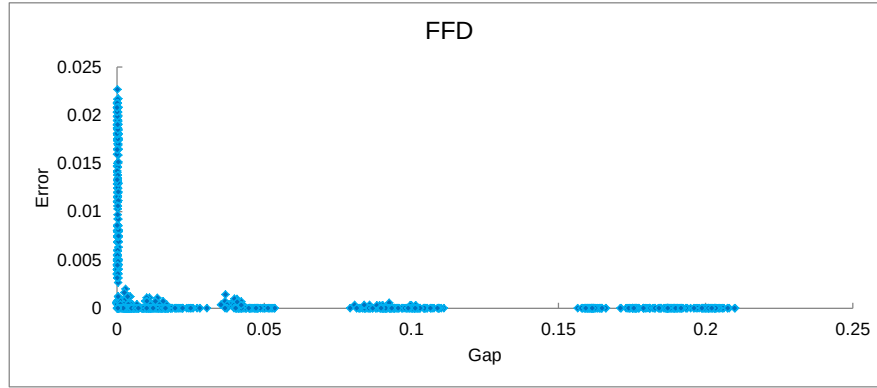


Figura 5.9 Desempeño de FFD contra el valor del índice *Gap* en instancias *u-w1-wn*.

5.6 NUEVAS INSTANCIAS DIFÍCILES TIPO UNIFORM

Con el objetivo de evaluar el desempeño de los algoritmos del estado del arte de BPP sobre casos de prueba de gran escala se crearon instancias de mayores magnitudes con características similares a la clase *uniform* (descritas en la Sección 3.2.2): capacidad del contenedor $c = 150$ y pesos distribuidos uniformemente entre 20 y 100. Los tamaños de los casos generados varían entre $n = 1500$ y $n = 1000000$.

La Figura 5.10 muestra algunas de las gráficas de desempeño de las heurísticas simples para BPP (descritas en la Sección 2.4) partiendo de permutaciones aleatorias de objetos. Cada gráfica presenta, el número promedio de contenedores que se obtuvieron mediante la generación de 100 soluciones aleatorias con cada heurística para 10 instancias de cada tamaño n . En ella puede observarse que, contrario al comportamiento observado para instancias de tamaño pequeño ($n = 250$), donde FF y BF muestran un mejor desempeño que la estrategia WF, al aumentar el tamaño del problema ($n > 1500$), el rendimiento de los algoritmos es opuesto, sobresaliendo ahora los resultados obtenidos por WF.

Los resultados observados en las gráficas de desempeño permiten identificar diferentes comportamientos en los algoritmos heurísticos simples y cuestionar la aplicabilidad de los métodos heurísticos, así como el alcance de las instancias de prueba propuestas en la literatura. Parece ser que los casos de prueba estándar, utilizados para evaluar a los

algoritmos del estado del arte, cuyo tamaño mayor es $n = 1000$, no son representativos de verdaderas instancias difíciles y no permiten validar el desempeño de los algoritmos a gran escala. Por lo que se recomienda, ampliar los bancos de prueba de BPP agregando instancias con mayores números de objetos, que posibiliten el estudio del impacto del tamaño del problema en el comportamiento de los algoritmos del estado del arte.

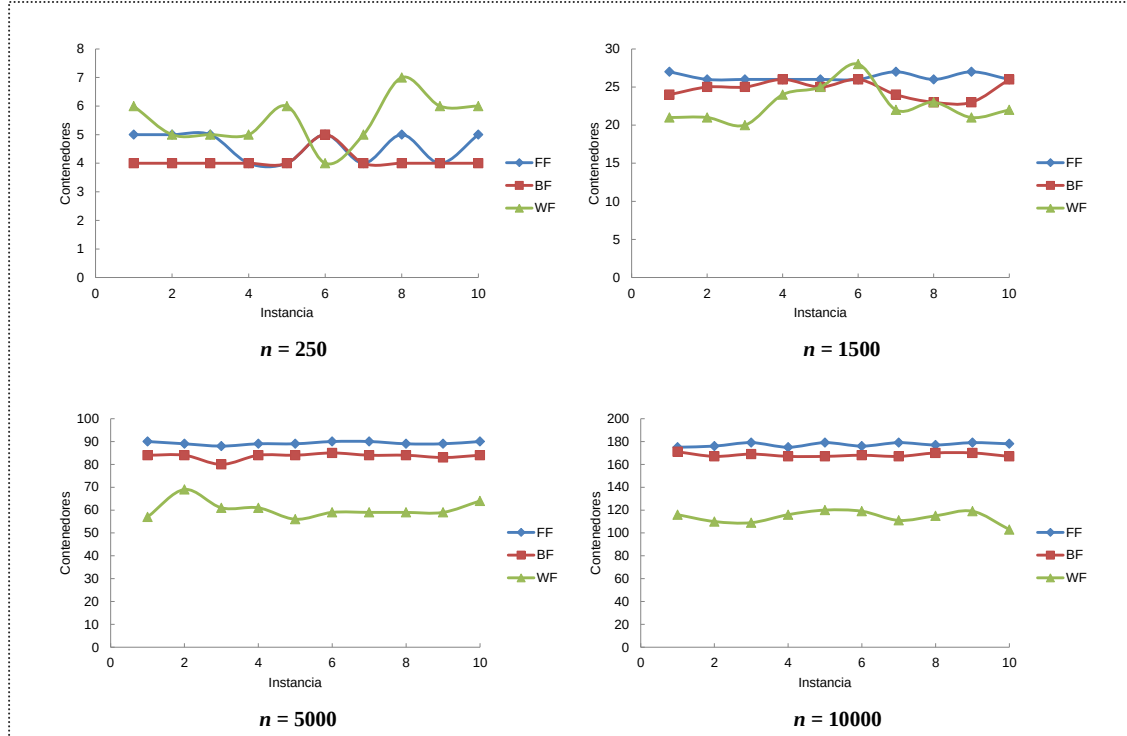


Figura 5.10 Desempeño de heurísticos de BPP en instancias tipo *uniform* grandes.

5.7 CONCLUSIONES DEL ESTUDIO

El análisis del proceso de optimización de BPP y de sus algoritmos de solución, reveló la importancia de conocer las características del problema que impactan el desempeño algorítmico, así como de generar instancias representativas de dichas características. Las conclusiones obtenidas sobre los factores que causan la dificultad de las instancias de BPP, producto de los estudios presentados en el Capítulo 4 y Capítulo 5 de esta tesis son resumidas a continuación. Este conocimiento ha sido aplicado para diseñar un nuevo algoritmo genético para BPP, el cual es presentado en el Capítulo 6.

1. La característica que tiene un mayor impacto en la dificultad de las instancias de BPP está relacionada con la estructura de las soluciones óptimas. Las instancias cuya solución óptima presenta muy *poco espacio libre en los contenedores* son las más *difíciles*.
2. Las instancias con *objetos grandes* (promedio de los pesos de los objetos mayor que el 50% de la capacidad del contenedor) parecen ser las más *fáciles* para la mayoría de los algoritmos del estado del arte de BPP, debido a que las técnicas heurísticas pueden obtener ventaja al acomodar los objetos en orden decreciente de sus pesos.
3. Las instancias con *objetos pequeños* (promedio de los pesos de los objetos menor que el 30% de la capacidad del contenedor) son *fáciles* para los algoritmos que incluyen estrategias heurísticas aleatorias.
4. Las instancias con *objetos medianos* (promedio de los pesos de los objetos entre el 30% y el 50% de la capacidad del contenedor) son *difíciles*. Y los algoritmos que obtienen los mejores resultados son estrategias híbridas que incorporan diferentes heurísticas.
5. Las instancias con una *alta variabilidad* en los pesos de los objetos son *fáciles*, pues incluyen objetos grandes y pequeños que pueden combinarse entre sí sin problemas.
6. Los algoritmos de solución de BPP deben incluir estrategias que aprovechen las propiedades de las instancias de BPP. El objetivo principal de los algoritmos debe ser explorar el espacio de búsqueda explotando al máximo el espacio de cada contenedor de la solución y dando tratamientos diferentes al acomodo de objetos grandes, medianos y pequeños.
7. Los bancos de prueba de BPP deben ser actualizados, incluyendo instancias de mayor escala, así como los casos de prueba propuestos en este trabajo de tesis.

Capítulo 6

UN ALGORITMO GENÉTICO PARA BPP CON TRANSMISIÓN DE GENES CONTROLADA

En este capítulo, se presenta un nuevo algoritmo de solución de BPP diseñado a partir del conocimiento obtenido del análisis experimental del proceso de optimización BPP. Este algoritmo es denominado GGA-CGT (Grouping Genetic Algorithm with Controlled Gene Transmission), *Algoritmo Genético de Agrupación con Transmisión de Genes Controlada*. El algoritmo propuesto es comparable con los mejores algoritmos de estado del arte, superando los resultados publicados para las instancias de la clase Hard28, que son, a la fecha, las instancias más difíciles para los algoritmos de solución de BPP.

GGA-CGT promueve la transmisión de los mejores genes en los cromosomas sin perder el equilibrio entre la presión selectiva y diversidad de la población. La transmisión de los mejores genes (los contenedores más llenos) se realiza por medio de un nuevo conjunto de operadores genéticos de agrupación, mientras que el proceso evolutivo es equilibrado por medio de una nueva técnica de reproducción que controla la exploración del espacio de búsqueda y evita la convergencia prematura del algoritmo.

La Figura 6.1 presenta el algoritmo GGA-CGT (los operadores genéticos y la técnica de reproducción son descritos en la siguientes secciones). El proceso inicia generando una población P de individuos con la heurística de empaqueo FF- $\tilde{n}$ (Línea 1). Luego, durante un máximo de max_gen generaciones, algunos individuos de P son seleccionados para ser recombinados y mutados en dos fases. En la primera fase (Líneas 3-5), n_c individuos son seleccionados con la estrategia Selección_Controlada para aplicarles el operador de cruce Cruzamiento_Nivel_Gen+FFD; en seguida, la estrategia Reemplazo_Controlado es utilizada para introducir la descendencia. En la segunda fase (Líneas 6-8), de acuerdo con una max_edad predefinida, los individuos del grupo élite son clonados y los mejores n_m individuos son seleccionados para aplicarles el operador Mutación_Adaptativa+RP; posteriormente, los clones son introducidos a la población por medio de la estrategia Reemplazo_Controlado. El algoritmo se detendrá antes de alcanzar el número máximo de generaciones max_gen en caso de encontrar una solución cuyo tamaño coincida con el límite inferior L_2 de Martello y Toth [Martello90b]. El resultado final del algoritmo es el individuo con la mayor aptitud de todo el proceso evolutivo.

ALGORITMO GENÉTICO DE AGRUPACIÓN CON TRANSMISIÓN DE GENES CONTROLADA	
Procedimiento GGA-CGT	
1	Generar una población inicial P con FF- $\tilde{n}$;
2	mientras $generación < max_gen$ y $Tamaño(mejor_solución) > L_2$
3	Seleccionar n_c individuos para cruzar por medio de Selección_Controlada;
4	Aplicar Cruzamiento_Nivel_Gen+FFD a los n_c individuos seleccionados;
5	Aplicar Reemplazo_Controlado para introducir las crías;
6	Seleccionar n_m individuos y clonar soluciones élite por medio de Selección_Controlada;
7	Aplicar Mutación_Adaptativa+RP a los mejores n_m individuos;
8	Aplicar Reemplazo_Controlado para introducir los clones;
9	actualizar $mejor_solución_global$;
10	fin mientras;
fin GGA-CGT	

Figura 6.1 Procedimiento general del algoritmo GGA-CGT.

El algoritmo GGA-CGT ha sido diseñado utilizando el conocimiento aprendido durante la caracterización del proceso de optimización de BPP. Se identificó que la estructura de las soluciones óptimas es un factor crítico en la dificultad de las instancias, pues en los casos más difíciles se debe utilizar la capacidad de los contenedores casi al 100%; en este

sentido, el algoritmo preserva inteligentemente los contenedores que hacen mejor uso de su capacidad (los mejores genes de los cromosomas), explorando el espacio de búsqueda mediante nuevos operadores genéticos que explotan el espacio de los contenedores de la solución. Por otro lado, se identificó que el peso de los objetos es un factor que impacta de manera importante el desempeño de las técnicas de empaçado; debido a lo anterior, GGA-CGT incorpora nuevas estrategias de acomodo que dan tratamientos diferentes al acomodo de objetos grandes, medianos y pequeños.

6.1 COMPONENTES DE GGA-CGT

En esta sección se describen las heurísticas agrupación inteligentes que se utilizan para la generación y evolución de las soluciones en el algoritmo GGA-CGT. Como principal aportación, se propone lo siguiente: una nueva heurística de empaçado inteligente que simplifica y mejora el empaçado de los objetos, un nuevo procedimiento de reacomodo que permite la exploración y explotación del espacio de búsqueda y un nuevo conjunto de operadores genéticos de agrupación que promueven la transmisión de los mejores genes en los cromosomas.

6.1.1 CODIFICACIÓN GENÉTICA

La codificación de una solución de un problema de optimización en un cromosoma es un punto clave para el buen desempeño de un algoritmo genético. En general, para BPP se han propuesto tres esquemas de codificación: representación basada en contenedores, representación basada en objetos y representación basada en grupos. En la *codificación basada en contenedores*, un cromosoma tiene una longitud igual al número de objetos y cada gen representa un objeto e indica el contenedor que almacena dicho objeto. En la *codificación basada en objetos*, un cromosoma representa una permutación de objetos y es necesario aplicar una función decodificadora para recuperar cada solución. En la *codificación basada en grupos*, los cromosomas pueden variar en longitud dependiendo del número de contenedores que se utilizan en cada solución; cada gen en el cromosoma representa un contenedor que almacena un grupo de objetos [Gen00].

En la búsqueda de un esquema de codificación para BPP, Falkenauer y Delchambre [Falkenauer92b] observaron que las codificaciones basadas en contenedores y basadas en objetos podrían ocasionar dificultades, ya que son altamente redundantes y no están adaptados a la función de costo de BPP; así mismo, demostraron que el esquema de codificación basado en grupos evita estos problemas. Por los motivos antes expuestos, se decidió representar las soluciones de GGA-CGT con el esquema de codificación basado en grupos. De esta manera, los operadores genéticos trabajarán con los contenedores y la información sobre los objetos que pertenecen a cada contenedor.

6.1.2 FUNCIÓN DE APTITUD

Siendo el objetivo de BPP minimizar el número de contenedores usados para acomodar un conjunto de objetos, sería natural definir la función de aptitud igual al número de contenedores utilizados en cada solución. Sin embargo, esto daría lugar a un espacio de búsqueda muy poco amigable, en el que sería imposible evaluar la calidad del acomodo de los objetos y distinguir entre soluciones con diferentes características. El objetivo principal de GGA-CGT es trabajar a nivel de genes para promover la transmisión de los mejores contenedores; para contribuir con dicho objetivo, la evaluación de la aptitud se hace por medio de la función definida por la Ecuación 6.1 [Falkenauer92b], donde m es el número de contenedores que se utilizan en la solución, S_i es la suma de los pesos de los objetos en el contenedor i y c es la capacidad de los contenedores. Para esta función, es mejor tener algunos contenedores casi completos y algunos casi vacíos, en lugar de varios contenedores igualmente llenos; esto se debe a que los contenedores casi vacíos permitirán acomodar más fácilmente objetos adicionales, que de otro modo podría ser demasiado grandes para caber en cualquiera de los contenedores medio llenos. El objetivo de GGA-CGT es maximizar los valores de la aptitud de los individuos de la población.

$$F_{BPP} = \frac{\sum_{i=1}^m (S_i / c)^2}{m} \quad (6.1)$$

6.1.3 POBLACIÓN INICIAL

En la mayoría de los enfoques evolutivos para BPP, la población inicial es generada de manera aleatoria mediante la ejecución de una heurística de empaqueo de BPP sobre permutaciones aleatorias de los objetos [Falkenauer96, Stawowy08, Rohlfshagen10]. El proceso de generación de individuos de GGA-CGT, aunque también genera soluciones aleatorias, primero contribuye a la simplificación y mejora de empaqueo de los objetos; para lograr lo anterior, se utiliza una heurística que se basa en un criterio de dominación que, teniendo en cuenta el conjunto de n objetos, determina el número $\tilde{n}$ de objetos con pesos mayores que el cincuenta por ciento de la capacidad del contenedor. El valor de $\tilde{n}$ representa el número total de objetos grandes que no se podrían combinar con otros del mismo subconjunto. En este trabajo se propone la heurística FF- $\tilde{n}$, que aprovecha este conocimiento para reducir el número de pasos en el proceso de empaqueo y obtener contenedores con un mejor llenado:

First Fit con $\tilde{n}$ objetos pre-asignados (FF- $\tilde{n}$). En primer lugar, los $\tilde{n}$ objetos grandes son empacados en contenedores separados; posteriormente, los $n - \tilde{n}$ objetos restantes son empacados usando la heurística de acomodo FF sobre una permutación aleatoria de los objetos. Cada uno de los $n - \tilde{n}$ objetos es empacado en el primer contenedor que tiene suficiente capacidad disponible para almacenarlo; si ninguno de los contenedores puede almacenarlo, el objeto se almacenará en un nuevo contenedor vacío.

Resultados experimentales revelan una mejora significativa en la calidad de la población inicial P y la eficiencia de empaqueo al generar las soluciones de P con FF- $\tilde{n}$ en lugar de FF. Por ejemplo, la Figura 6.2 presenta el número promedio de contenedores que se obtuvieron mediante la generación de 100 soluciones aleatorias con las heurísticas de empaqueo FF y FF- $\tilde{n}$ para las 20 instancias de la clase uniforme u1000 (descritas en la Sección 3.2.2), donde la proporción promedio de objetos grandes es $\tilde{n} / n = 0.3$. En esta figura se puede observar que al acomodar primero los $\tilde{n}$ objetos grandes, la calidad promedio de las soluciones es mejorada entre 5 y 8 contenedores por instancia.

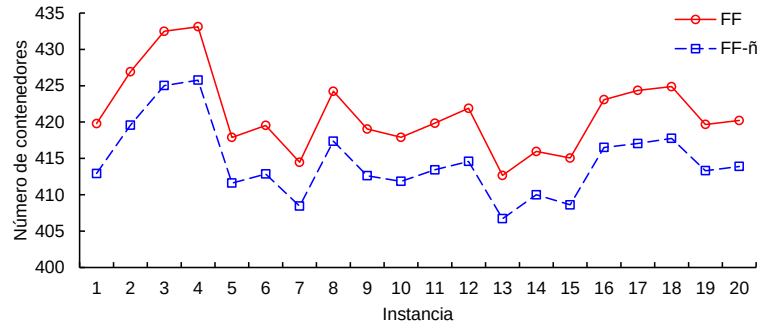


Figura 6.2 Resultados experimentales obtenidos por las heurísticas FF y FF-ñ sobre las instancias u1000.

Para las 1615 instancias estándar, cuando fue posible aplicar FF-ñ ($\tilde{n} > 0$), se observó una mejora promedio de 3.67 contenedores al generar las soluciones en la población con dicho método en lugar de FF; para instancias grandes ($n \geq 500$) cuando $\tilde{n} / n > 0.5$, la mejora promedio fue de 10.94 contenedores. Los resultados obtenidos demuestran que añadir inteligencia a la heurística de empaqueo mejora la calidad de las soluciones y la eficiencia de empaqueo. En la Sección 6.3.3 se muestra un análisis del desempeño del algoritmo GGA-CGT sobre instancias con diferentes proporciones de objetos grandes, mostrando el impacto de generar la población inicial con FF-ñ en lugar de FF.

En general, los $n - \tilde{n}$ objetos pueden ser acomodados utilizando cualquier heurística de empaqueo de BPP. En este caso, FF fue elegido después de compararlo con BF y WF, debido a que FF es el más simple de ellos, y luego de un análisis experimental, no se observaron diferencias significativas en la calidad promedio de la población generada por las tres heurísticas (estas heurísticas son descritas en la Sección 2.4).

6.1.4 OPERADOR DE CRUZAMIENTO POR AGRUPACIÓN

En general, un operador de cruzamiento toma dos soluciones padre (cromosomas) y las recombina para producir una solución hijo. Dados dos cromosomas, cruzarlos implica determinar qué genes deben ser heredados de padres a hijos. Esos genes que sobreviven el proceso evolutivo se caracterizan por ser dominantes sobre los otros. En muchos tipos de cromosomas, los patrones de genes que se encuentran unidos son significativos, y la preservación de un conjunto de los genes (un segmento del cromosoma) de una buena

solución tiende a conducir a otra buena solución. Sin embargo, cuando se utiliza un esquema de codificación basado en grupos, los patrones de genes que se encuentran unidos son mucho menos significativos, y la preservación de estos patrones no es una meta apropiada para un operador de cruzamiento [Brown93].

Un operador de cruzamiento de alta calidad debe tener la capacidad de propagar a la descendencia los patrones de los cromosomas que contribuyen a aumentar la aptitud de los padres. En el caso de una solución de BPP, codificada mediante el esquema basado en grupos, es fácil darse cuenta que, mientras mayor uso se dé a la capacidad de cada contenedor, se necesitarán menos contenedores para empacar los objetos. Así, en una solución óptima, se espera que la mayoría de los contenedores estén casi llenos; Por lo tanto, los contenedores bien llenos, propiedad aquí denominada como *patrón de llenado-eficiente*, son las características de los cromosomas que contribuyen a la aptitud de los individuos. Considerando el patrón de llenado-eficiente, se propone un nuevo método de cruzamiento para BPP que permite a los cromosomas de los padres contribuir a nivel de gen en lugar de a nivel de segmento, dando una mayor probabilidad de preservación a los mejores genes (los contenedores más llenos). La Figura 6.3 incluye un ejemplo de este operador.

Cruzamiento_Nivel_Gen. Dadas dos soluciones padre p_1 y p_2 , el operador propuesto se utiliza dos veces, para generar dos hijos, cruzando: p_1 con p_2 y p_2 con p_1 . Este operador considera los contenedores en orden descendente de su llenado, para aumentar la probabilidad de que los mejores contenedores sean heredados por los hijos. Comenzando con el mejor contenedor, contenedores individuales de ambos padres son comparados en paralelo, contenedor por contenedor. Para cada par de contenedores, el contenedor más lleno es el primero en ser heredado a la nueva solución, seguido de inmediato por el otro contenedor. En caso que ambos contenedores tengan el mismo llenado, se da preferencia al contenedor del primer padre. Si una solución incluye más contenedores, después de comparar y heredar todos los pares de contenedores, los contenedores restantes de la solución más grande son heredados directamente a la nueva solución. Los contenedores

con objetos duplicados son eliminados de la nueva solución, y los objetos que queden fuera de la solución son reinsertados con la heurística determinista FFD.

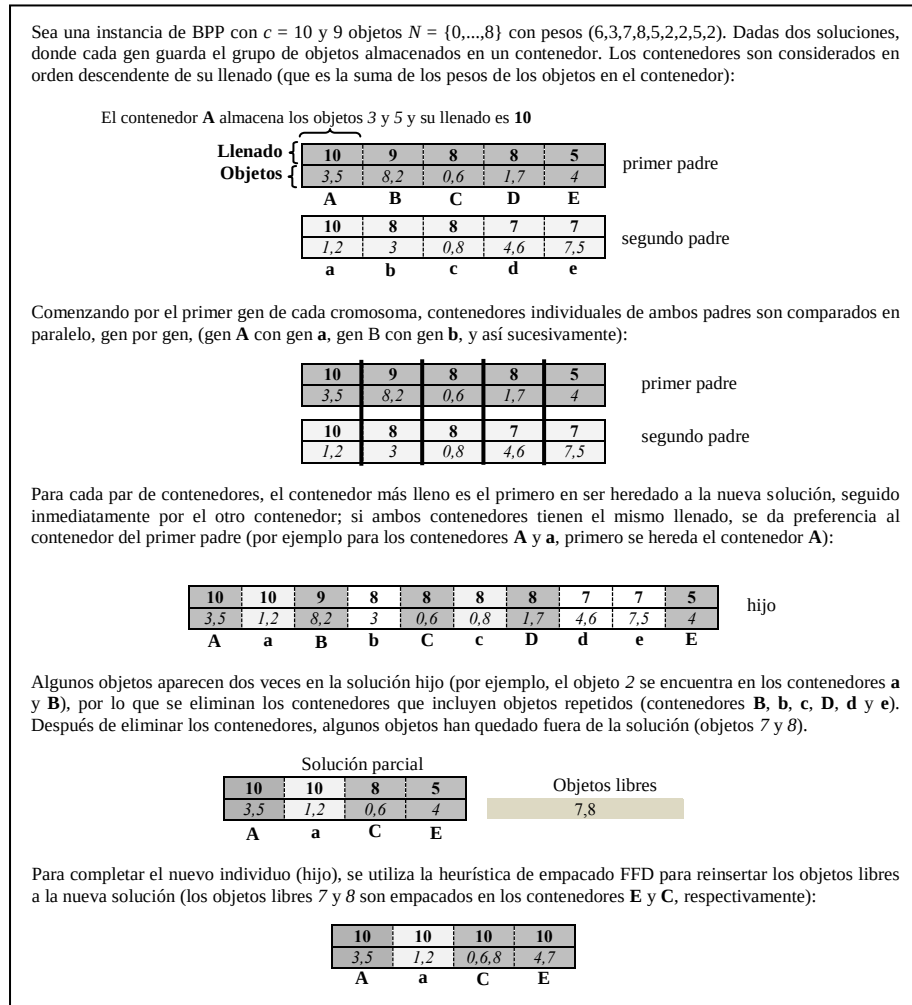


Figura 6.3 Ejemplo del operador Cruzamiento_Nivel_Gen que preserva el patrón de llenado-eficiente.

Para mostrar la efectividad del nuevo operador de cruzamiento a nivel de gen, se realizó un experimento comparativo con tres alternativas de cruzamiento: 1) cruzamiento a nivel de segmento [Falkenauer92b]; 2) cruzamiento a nivel gen sin acomodar los contenedores en orden descendente de su llenado; y 3) Cruzamiento_Nivel_Gen. Todos los algoritmos se ejecutaron bajo las mismas condiciones. Se utilizó una población de 100 individuos generados con FF-ñ. En cada experimento, para un máximo de 100 generaciones, los

100 individuos (seleccionados al azar) fueron cruzados y reemplazados por sus crías; los objetos libres fueron reinsertados a la solución con la heurística de empaqueo FFD. La Tabla 6.1 muestra los resultados que se obtuvieron en el experimento.

La Tabla 6.1 muestra primero el número de instancias de prueba (Inst.), seguido por el número de óptimos encontrado por cada alternativa de cruzamiento (opt.). Así mismo, para cada operador, se muestra la diferencia relativa total (dif.), definida como $(y - x) / x$, donde x es el número de contenedores óptimo e y es el número de contenedores obtenidos por el algoritmo. La última fila de la tabla muestra la efectividad, definida como o / t , donde o es el número de soluciones óptimas obtenidos por el algoritmo y t es el total de instancias utilizadas en el experimento.

Tabla 6.1 Resultados experimentales de tres alternativas de cruzamiento.

Clase	Inst.	Cruzamiento por Segmento		Cruzamiento a nivel gen			
				sin orden en llenado		con orden en llenado	
		opt.	dif.	opt.	dif.	opt.	dif.
Uniform	80	2	1.33	16	0.78	63	0.09
Triplets	80	0	7.90	0	6.64	0	2.00
Data Set 1	720	568	3.11	618	1.40	695	0.24
Data Set 2	480	284	7.63	322	5.83	439	0.65
Data Set 3	10	0	0.48	0	0.48	7	0.05
Was 1	100	8	5.11	48	2.88	89	0.61
Was 2	100	47	2.42	78	1.01	95	0.23
Gau 1	17	3	0.83	5	0.70	12	0.20
Hard28	28	5	0.33	5	0.33	5	0.33
Total	1615	917	29.18	1092	20.08	1405	4.44
Efectividad		0.56		0.67		0.86	

De la Tabla 6.1, se puede observar que se obtiene una mejora en la efectividad de 11% al utilizar un cruzamiento a nivel de gen en lugar de la de cruzamiento a nivel de segmento; y una mejora de 19% al dar una mayor probabilidad de ser heredados a los mejores contenedores. Estos resultados demuestran que la estrategia de preservar el patrón de llenado-eficiente (promoviendo la transmisión de los mejores genes) tiende a conducir a un mejor desempeño.

6.1.5 OPERADOR DE MUTACIÓN POR AGRUPACIÓN

El operador de mutación tiene la función de introducir pequeños cambios aleatorios en la población mediante la modificación de algunos de los cromosomas. En los algoritmos genéticos tradicionales, la mutación se aplica modificando cada gen del cromosoma con una pequeña probabilidad. El operador de mutación para la BPP, codificado con el esquema basado grupos, se ha definido de manera diferente [Falkenauer92b]: dada una solución de BPP (cromosoma), algunos contenedores son eliminados y los objetos que componían dichos contenedores son reinsertados a la solución con una heurística de reacomodo. Por lo general, la selección de los contenedores a eliminar se realiza al azar [Falkenauer96, Rohlfshagen10]; Sin embargo, se ha observado que, para aumentar la probabilidad de mejorar la calidad de las soluciones, es necesario eliminar al menos tres contenedores y siempre eliminar el contenedor menos lleno [Singh07, Falkenauer92b]; en este trabajo se propone hacer una combinación de estos dos enfoques.

Por otro lado, los operadores de mutación que han sido propuestos en la literatura son altamente perturbadores y pierden información genética importante al dar la misma probabilidad de eliminación a los genes buenos y malos; de esta manera, algunos de los contenedores más llenos son eliminados de la solución, perdiendo patrones que solían contribuir a la calidad de la población [Falkenauer96, Brown05]. Considerando el caso de que todos los cromosomas, sin importar su tamaño, tienen la misma probabilidad de mutación p_m (siendo p_m es el porcentaje de contenedores a vaciar); por ejemplo, cuando $p_m = 0.1$, para una solución de 20 contenedores, se esperaría eliminar aproximadamente 2 contenedores, que pueden ser muy pocos para obtener una mejora en la calidad de la solución; sin embargo, para una solución de 1000 contenedores, se eliminarían alrededor de 100 contenedores, lo cual ocasionaría la pérdida de una gran cantidad de información genética y representaría un trabajo sustancial para la heurística de reacomodo. Con el objetivo de introducir nuevo material genético y preservar el patrón de llenado-eficiente, se propone un nuevo operador de mutación que promueve la transmisión de los mejores

genes del cromosoma y optimiza el proceso evolutivo al adaptar la probabilidad de mutación p_m al tamaño de la solución.

Mutación_Adaptativa. Dada una solución de BPP, los contenedores son considerados en orden descendente de su llenado, eliminando los n_b contenedores menos llenos y reinsertando los objetos libres con una heurística reacomodo (ver Sección 6.1.6). En contraste con el enfoque adoptado en la literatura, se propone una función adaptativa para calcular el número de contenedores que deben ser eliminados de la solución. Para cada solución a mutar, el número de contenedores a eliminar n_b es una variable aleatoria que se calcula en función del tamaño del cromosoma y del número de contenedores incompletos (que no están llenos al 100%). Dada una solución BPP con m contenedores, de los cuales ι contenedores están incompletos ($\iota > 1$), el número de contenedores a eliminar es dado por la Ecuación 6.2, donde ε es la proporción de eliminación definida por la ecuación 6.3, p_ε es la probabilidad de eliminación definida por la Ecuación 6.4, y k es un parámetro que define la tasa de cambio de ε y p_ε con respecto a ι ($k > 0$). La Figura 6.4 incluye un ejemplo de este operador.

$$n_b = \lceil \iota \cdot \varepsilon \cdot p_\varepsilon \rceil \quad (6.2)$$

$$\varepsilon = \frac{2 - \frac{\iota}{m}}{\iota^{1/k}} \quad (6.3)$$

$$p_\varepsilon = 1 - U\left(0, \frac{1}{\iota^{1/k}}\right) \quad (6.4)$$

Dadas la Ecuación 6.3 y la Ecuación 6.4; la proporción de eliminación ε es inversamente proporcional al número de contenedores incompletos (ι) y al porcentaje de contenedores incompletos (ι / m). De esta manera, el porcentaje de contenedores para eliminar (n_b / m) es mayor cuando las soluciones son más pequeños y disminuye con soluciones más grandes. Alternativamente, la probabilidad de eliminación es directamente proporcional al número de contenedores incompletos, lo que permite, en la mayoría de los casos, más variaciones en ε cuando las soluciones son más pequeñas.

Sea una instancia de BPP con $c = 10$ y 10 objetos $N = \{0, \dots, 9\}$ con pesos $(6, 3, 7, 8, 4, 2, 2, 5, 2, 8)$. Dada una solución, donde cada gen guarda el grupo de objetos almacenados en un contenedor. Los contenedores son considerados en orden descendente de su llenado:

Llenado	10	10	9	8	8	4
Objetos	3,5	1,7	8,2	0,6	9	4
	A	B	C	D	E	F

Dado el número de contenedores de la solución $m = 6$, el número de contenedores incompletos $\iota = 4$, la tasa de cambio $k = 3$ y un número aleatorio $U(0, 0.62) = 0.5$. La proporción de eliminación (Ecuación 5.3) es $\varepsilon = 0.839$, la probabilidad de eliminación (Ecuación 5.4) es $p_e = 0.5$ y el número de contenedores a eliminar, definido por la Ecuación 5.2, es $n_b = 2$. Los dos contenedores menos llenos (contenedores E y F) son eliminados de la solución, y los objetos que componían esos contenedores (objetos 9 y 4) deben ser reinsertados con alguna heurística de reacomodo.

Solución parcial				Los contenedores menos llenos que fueron eliminados		⇒	Objetos libres
10	10	9	8	8	4		
3,5	1,7	8,2	0,6	9	4		9,4
A	B	C	D	E	F		

Para completar la solución, se utiliza un procedimiento de reacomodo para reinsertar los objetos libres. Si se eligiera utilizar alguna heurística determinista tal como FF o BF, la nueva solución sería la misma que la original, porque los contenedores de la solución parcial no tienen una cantidad suficiente de espacio para almacenar los objetos libres. En este caso, convendría utilizar un procedimiento de reacomodo que permitiera hacer intercambios entre objetos libres y objetos empacados (esta heurística se describe en la siguiente sección); así, se podría obtener la solución óptima (los objetos 0 y 6 del contenedor D se sustituyen con el objeto libre 9, ahora los objetos 0 y 6 se encuentran libres con el objeto 4, y pueden ser reinsertados en el contenedor D y en un nuevo contenedor E).

10	10	9	10	10
3,5	1,7	8,2	9,6	4,0
A	B	C	D	E

Figura 6.4 Ejemplo del operador de mutación adaptativa que preserva el patrón de llenado-eficiente.

6.1.6 HEURÍSTICAS DE REACOMODO PARA OPERADORES DE AGRUPACIÓN

En el proceso de solución, los operadores genéticos generan y modifican los individuos y algunos contenedores se eliminan y existen objetos libres que deben ser reinsertados a la solución (véase la Figura 6.3 y la Figura 6.4); para realizar este reacomodo de objetos libres, algunos autores han utilizado heurísticas simples de empaqueo como FFD o BFD [Falkenauer92b, Brown05], y otros han utilizado optimizaciones locales [Falkenauer96, Yesil12]. Se ha demostrado que el éxito de un algoritmo genético de agrupación (GGA) depende en gran medida de la heurística de reacomodo y que la adición de inteligencia a estas heurísticas mejora el desempeño de un GGA [Brown93]. El ejemplo de la Figura 6.4 evidencia que el uso de una heurística de reacomodo capaz de hacer intercambios entre objetos libres y objeto empacados tiene un impacto positivo en la efectividad del operador de mutación. En este trabajo se propone una heurística de reacomodo sencilla que trabaja con pares de artículos libres.

Reacomodo_por_Pares (RP). Esta heurística se compone de dos etapas: primero, se recorre cada contenedor en un intento de mejorar su llenado haciendo intercambios entre pares de objetos empacados y objeto libres; segundo, los objetos libres se introducen en la solución utilizando la heurística FF. La Figura 6.5 presenta la heurística propuesta, se denota por $S' = \text{Swap}(S, F, (p,s), i)$ a la solución obtenida de S mediante el intercambio de los objetos p y s del contenedor B_j con el objeto i del conjunto de objetos libres F . La función $\text{Factible}(S')$ devuelve VERDADERO si el intercambio no supera la capacidad libre de B_j y FALSO en caso contrario.

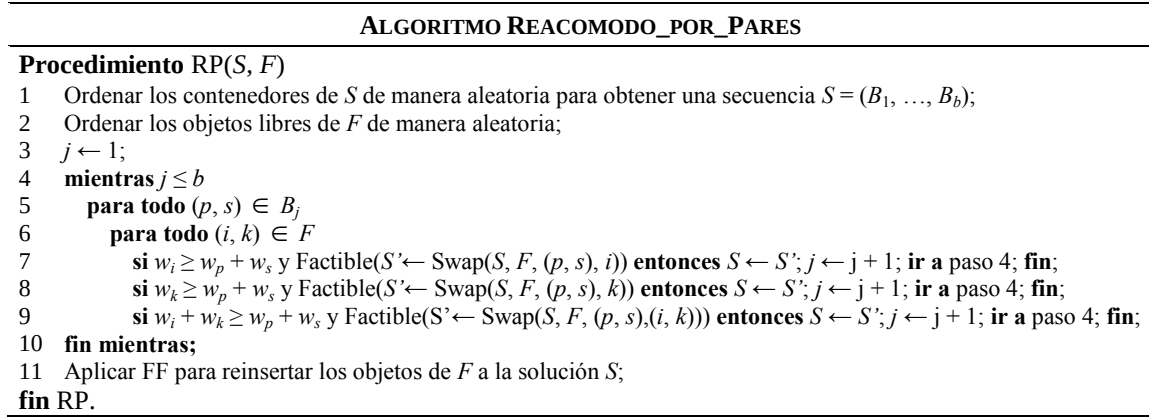


Figura 6.5 Procedimiento para reacomodo de objetos libres RP.

En la primera etapa (Líneas 1-10), dada una permutación aleatoria de los contenedores y una permutación aleatoria de los objetos libres, cada contenedor es considerado hasta que un reemplazo es posible. Para cada par de objetos en los contenedores, los objetos libres también se consideran en pares. Por cada dos pares de objetos, hay dos alternativas de reacomodo: a) si es posible, sustituir el par de objetos del contenedor con uno de los dos objetos libres de peso igual o superior que no exceda la capacidad del contenedor (Líneas 7-8); b) si el par de objetos libres puede llenar el contenedor tan bien o mejor que el par de objetos del contenedor sin sobrepasar la capacidad del contenedor, intercambiar el par de objetos del contenedor con el par de objetos libres (Línea 9). En la segunda etapa (línea 11), se aplica la heurística de FF para reinsertar a los objetos libres y completar la solución; una permutación aleatoria de los objetos libres es útil cuando el

peso del objeto libre más grande es menor del cincuenta por ciento de la capacidad del contenedor, en otro caso, es mejor tomar los objetos libres en orden descendente, para inducir el patrón de llenado-eficiente.

El procedimiento de reacomodo RP tiene consecuencias importantes en la explotación y exploración del espacio de búsqueda. Mejora la calidad del material genético, al obtener contenedores más llenos, cuando se realiza un intercambio que aumenta el llenado del contenedor. También permite la exploración de nuevos grupos de objetos, cuando objetos con pesos distintos o menores son liberados y reacomodados con FF; por esta razón, se realizan intercambios incluso cuando los objetos no cambian el llenado de los contenedores.

Para mostrar la contribución del operador Mutación_Adaptativa y de la heurística de reacomodo RP, se realizaron tres experimentos: 1) utilizando la mutación propuesta por Falkenauer con $p_m = 0.1$ [Falkenauer92b]; 2) utilizando la Mutación_Adaptativa; y 3) utilizando la Mutación_Adaptativa con la heurística RP. Todos los experimentos se llevaron a cabo en las mismas condiciones. Se utilizó una población de 100 individuos generados con FF-ñ; luego, para un máximo de 100 generaciones, los 100 individuos fueron mutados. En los dos primeros experimentos, se utilizó FFD para reinsertar los objetos libres.

Tabla 6.2 Resultados experimentales de la mutación de Falkenauer y la mutación adaptativa propuesta.

Clase	Inst.	Mutación_Falkenauer		Mutación_Adaptativa		Mutación_Adaptativa+RP	
		opt.	dif.	opt.	dif.	opt.	dif.
Uniform	80	3	1.06	32	0.33	78	0.01
Triplets	80	0	7.83	0	5.94	42	0.43
Data Set 1	720	537	3.10	623	1.00	694	0.25
Data Set 2	480	276	7.77	347	4.28	477	0.09
Data Set 3	10	0	0.42	0	0.21	8	0.03
Was 1	100	15	4.72	55	2.50	99	0.05
Was 2	100	61	1.79	87	0.60	99	0.04
Gau 1	17	2	0.92	5	0.70	12	0.20
Hard28	28	5	0.33	5	0.33	4	0.35
Total	1615	899	27.97	1154	15.93	1513	1.51
Efectividad		0.55		0.71		0.93	

La Tabla 6.2 muestra los resultados que se obtuvieron en cada experimento. Se puede observar que se obtuvo una mejora en la efectividad de 16% al utilizar la mutación adaptativa en lugar de la mutación Falkenauer. Estos resultados confirman que la estrategia de preservar el patrón de llenado-eficiente, promoviendo la transmisión de los mejores genes, tiende a ser más eficaz, y que el uso de la proporción de eliminación como una variable aleatoria hace una importante contribución al éxito de las mutaciones. Finalmente, se puede observar que la estrategia de reacomodo por pares (RP) mejora el rendimiento del operador de mutación adaptativa en un 22%, lo que demuestra que la adición de inteligencia a la heurística de reacomodo mejora el rendimiento de un GGA.

Es importante subrayar que, el operador Mutación_Adaptativa + RP introduce nuevo material genético sin perder el patrón de llenado-eficiente, pero esto no quiere decir que este operador sea un mecanismo de búsqueda local. Al no permitir reducir el llenado de los mejores contenedores se asegura preservar el patrón de llenado-eficiente; sin embargo, se promueve la creación de nuevo material genético, al permitir hacer intercambios entre los objetos de los mejores contenedores y los objetos libres; Por otro lado, después de la primera fase de la heurística reacomodo, los objetos libres se reinsertan mediante el uso de la heurística de empaqueo FF, por lo que la aptitud de la solución mutada puede ser peor que su aptitud original.

6.2 TÉCNICA DE REPRODUCCIÓN

El desempeño final de un algoritmo genético (GA) no sólo depende de la efectividad de los operadores genéticos, sino también de la técnica de reproducción utilizada para implementarlos. La *técnica de reproducción* define la forma en que los individuos deben ser elegidos para la reproducción y qué individuos sobrevivirán de una generación a la siguiente. No todos los algoritmos producen una población totalmente nueva en cada generación; existen técnicas de *reproducción elitista* que conservan los individuos más aptos de cada generación y los llevan de la población actual a la nueva población; otras técnicas de reproducción, denominadas de *estado fijo*, permiten que los individuos padre

coexistan con su descendencia y las crías compiten con los padres para reproducirse y sobrevivir. Una técnica de reproducción bien planificada impactará significativamente el rendimiento GA.

El equilibrio entre la exploración y la explotación del espacio de búsqueda en un GA es determinado principalmente por los mecanismos utilizados para seleccionar individuos. La *presión selectiva* da a los individuos con una mayor aptitud una mayor probabilidad de ser seleccionados para la reproducción, mutación y supervivencia; sin ella, el proceso de búsqueda se vuelve aleatorio, y las regiones prometedoras del espacio de búsqueda no se ven favorecidas por la explotación. Por otra parte, la *diversidad de la población* y la selección de individuos con características variadas permiten la exploración de nuevas regiones del espacio de búsqueda, lo que ayuda a producir mejores soluciones y evitar la convergencia prematura del algoritmo [Smith07, Lozano08]. En este trabajo se propone una nueva técnica de reproducción, denominada *reproducción controlada*, en la que la presión selectiva y la diversidad de la población son equilibradas para aumentar el desempeño del algoritmo.

6.2.1 ESQUEMA DE SELECCIÓN

Un *esquema de selección* decide qué individuos transmitirán sus genes a la siguiente generación, ya sea a través de la clonación, del cruzamiento o la mutación; su objetivo es mejorar la calidad media de la población al dar a los individuos de mayor calidad una mayor probabilidad de contribuir con la próxima generación. En general, la selección se realiza considerando la aptitud relativa de los individuos y los esquemas de selección de la literatura pueden ser divididos en tres clases: selección proporcional, selección por torneo y selección por ranking. En la selección proporcional, conforme la población se hace más homogénea es difícil mantener una presión selectiva adecuada porque hay menos variaciones en la aptitud. Alternativamente, cuando la selección por torneo y la selección por ranking son configuradas con una alta presión selectiva, un importante

número de soluciones son descartadas, disminuyendo la exploración del espacio de búsqueda [Blickle96, Zhang00].

En este trabajo se propone un nuevo esquema de selección, Selección_Controlada, que hace una diferenciación clara entre los individuos de buena y mala calidad, dando a todos los individuos la oportunidad de contribuir al proceso evolutivo, pero forzando la supervivencia de los mejores individuos. La estrategia utiliza una técnica elitista junto con dos rankings invertidos, que garantizan la diversidad de la población. Así mismo, para mantener una presión selectiva adecuada, la técnica elitista controla la conservación de un grupo élite B , que incluye los mejores individuos de la población P , modificando los individuos élite sólo cuando superan un tiempo de vida max_edad predefinido. En la Sección 6.3.3 se muestran resultados experimentales sobre el impacto del tamaño del grupo élite B en el desempeño global de GGA-CGT.

Selección_Controlada para cruzamiento. Al aplicar el operador de cruzamiento, n_c padres son seleccionados para generar n_c hijos. Los n_c padres son seleccionados en dos grupos de la siguiente forma: se generan los conjuntos de padres G y R , que incluyen individuos buenos y aleatorios, respectivamente; estos elementos se van a combinar por pares, cruzando G_i con R_i ($0 \leq i < n_c/2$). El conjunto G incluye $n_c/2$ individuos tomados al azar de los mejores n_c individuos de P con una probabilidad uniforme; el conjunto R incluye $n_c/2$ individuos elegidos al azar de $P \setminus B$ con una probabilidad uniforme. Es posible que un mismo individuo de $P \setminus B$ sea seleccionado tanto en G como en R , por lo que se prohíbe que el mismo individuo tenga la misma posición i en ambos conjuntos. En contraste con otros métodos de selección, en el esquema de selección controlada, los mejores individuos se ven obligados a reproducirse en cada generación, promoviendo la supervivencia de los mejores genes. Sin embargo, se prohíbe el cruzamiento entre los mejores individuos (grupo élite B) promoviendo la diversificación genética.

Selección_Controlada para la mutación. Dado un número de individuos para mutar n_m (con $n_m > |B|$), el operador de mutación se aplica a los mejores n_m individuos de P en dos

fases: clonación del grupo elite y mutación de los mejores. Siguiendo un enfoque elitista, la clonación es para individuos del conjunto B , cuya edad es inferior a un parámetro max_edad predefinido, esto es con el fin de explotar las mejores soluciones sin perder buenos patrones. Después de la fase de clonación, los individuos a mutar se toman de P dado el orden decreciente de su aptitud.

6.2.2 ESTRATEGIA DE REEMPLAZO

Una estrategia de reemplazo define los individuos de la población actual que se ven obligados a perecer para dejar sitio a la nueva descendencia. En general, las estrategias de reemplazo se pueden dividir en tres clases: basadas en la edad, basadas en la aptitud y basadas en aleatoriedad (eliminando los individuos más viejos, los de menor aptitud o individuos aleatorios). Se ha demostrado que el reemplazo de individuos más viejos y el reemplazo de individuos aleatorios disminuyen la presión selectiva, perdiendo algunas de las mejores soluciones; por otro lado, el reemplazo de los peores individuos puede conducir a la convergencia prematura del algoritmo [Smith07, Lozano08].

En este trabajo se propone una estrategia de reemplazo, Reemplazo_Controlado, que contribuye a la búsqueda de un equilibrio adecuado entre la exploración y la explotación del espacio de búsqueda. Para promover la exploración, los nuevos individuos están siempre autorizados a ingresar en la población. Las crías son introducidas utilizando una estrategia, que combina un enfoque elitista y una estrategia para la conservación de la diversidad de la población, reemplazando los peores individuos y los individuos con aptitud duplicada:

Reemplazo_Controlado para cruzamiento. Después de que se aplica el esquema de Selección_Controlada, se tiene un conjunto de soluciones de alta calidad G como los primeros padres y un conjunto de soluciones aleatorias R como los segundos padres. Posteriormente, el operador de cruzamiento genera un conjunto C de n_c hijos. Los primeros $n_c / 2$ hijos son introducidos en P sustituyendo a los individuos del conjunto de

padres aleatorios R . Los otros $n_c / 2$ hijos se introducen en $P \setminus (B \cup R)$ con las siguientes reglas: a) si hay individuos con aptitud duplicada, reemplazarlos con nuevas crías; b) si aún hay crías que no han sido insertadas en P , introducir las reemplazando las peores soluciones.

Reemplazo_Controlado para la mutación. Cuando se aplica el operador de mutación, algunos de los mejores individuos se clonan para preservar las mejores soluciones. Cada clon se puede introducir en la población de dos maneras: a) si hay soluciones con aptitud duplicada, entonces el clon reemplazará una de ellas; b) si la primera alternativa no es posible, entonces el clon reemplazará a la peor de las soluciones.

Es importante notar que si x individuos (soluciones) tienen el mismo valor de aptitud, la estrategia de reemplazo controlado intenta reemplazar $x - 1$ individuos con nuevas crías. El hecho de que dos individuos tengan la misma aptitud no implica necesariamente que sean soluciones duplicadas; sin embargo, dado que la búsqueda de soluciones duplicadas es demasiado costosa, la estrategia propuesta es una aproximación a este esquema.

6.2.3 COMPARACIÓN DE REPRODUCCIÓN CONTROLADA CON OTRAS TÉCNICAS

Para mostrar la contribución de técnica de reproducción controlada aquí propuesta, se realizó un experimento combinando tres esquemas de selección y dos estrategias de reemplazo con el objetivo de explorar el comportamiento de las diferentes técnicas de reproducción:

- *Selección basada en Ranking* (SR), donde los individuos a ser cruzados se seleccionan de entre los mejores 50 individuos con una probabilidad uniforme, y el operador de mutación se aplica a los mejores n_m individuos [Zhang00].
- *Selección proporcional* (SP), donde los individuos a ser cruzados se seleccionan usando con un procedimiento de ruleta, y el operador de mutación se aplica a los mejores n_m individuos [Zhang00].

- *Selección Controlada* (SC), donde la selección de individuos para el cruzamiento y la mutación se hace con esquema de selección propuesto, que promueve la supervivencia de los mejores genes con el uso de elitismo y la diversidad de la población con dos rankings invertidos.
- *Reemplazo del peor* (RP), donde los peores n_c individuos se sustituyen con las crías [Smith07].
- *Reemplazo controlado* (RC), donde la sustitución se realiza con la estrategia de reemplazo propuesta, que combina un enfoque elitista con el reemplazo de individuos con aptitudes duplicadas y con las peores aptitudes.

Los procedimientos se combinaron en seis técnicas de reproducción: SR + RP, SR + RC, SP + RP, SP + RC, SC + RP y SC + RC. Es importante señalar que, cuando se combinan SR o SP con RC, RC no es el método de reemplazo controlado completo, dado que la estrategia propuesta está vinculada a la selección controlada SC (con el grupo élite y el conjunto de los padres al azar R). Sin embargo, se utiliza la siguiente aproximación: a) si hay individuos con aptitud duplicada, reemplazarlos con nuevas crías; b) si hay crías que no han sido insertadas en P , entonces introducirlos mediante la sustitución de las peores soluciones. Todos los algoritmos se corrieron con la misma configuración, descrita en la Sección 5.3. Los dos primeros esquemas de selección (SR y SP) utilizan sólo los primeros cinco parámetros, ya que no incluyen un grupo élite. La Figura 6.6 muestra los resultados que se obtuvieron en el experimento. Para cada esquema de selección (SR, SP y SC), se presenta la diferencia relativa total que se obtiene cuando se combina con una de las dos estrategias de sustitución (RP y RC).

De la Figura 6.6, se puede observar que al utilizar la estrategia de reemplazo controlado RC en lugar de RP se obtiene una mejora en la efectividad, introduciendo diversidad en la población al reemplazar los individuos con aptitud repetida. También se puede ver la superioridad de la técnica de reproducción controlada propuesta (SC + RC), que mejora la efectividad de SR + RC y SP + RC en 198% y 463%, respectivamente. Los resultados

obtenidos confirman que la estrategia de reproducción controlada mediante la cual se equilibra la presión selectiva y la diversidad de la población tiene un fuerte impacto en el desempeño de GGA-CGT.

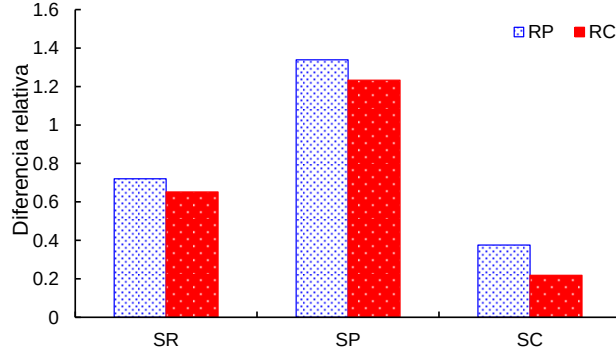


Figura 6.6 Resultados experimentales para tres técnicas de reproducción.

6.3 EXPERIMENTOS COMPUTACIONALES

El algoritmo GGA-CGT fue desarrollado en el lenguaje C++ y fue compilado utilizando Borland C++ 5.02 en modo Win32. Los experimentos se realizaron en un equipo con un procesador Intel Core 2 Duo E6300 1.86 GHz. Al igual que en la mayor parte del trabajo previo sobre BPP [Fleszar11, Alvim04], para cada instancia a resolver, se realizó una sola ejecución del algoritmo GGA-CGT, con la semilla para la generación de números aleatorios fijada en 1. En todos los experimentos, se utilizaron siguientes parámetros para el algoritmo:

- Tamaño de la población $|P| = 100$
- Número máximo de generaciones $max_gen = 500$
- Número de individuos a cruzar $n_c = 20$
- Número de individuos a mutar $n_m = 83$
- El número de contenedores a eliminar n_b en el operador de mutación para las soluciones no clonadas es calculado utilizando una tasa de cambio $k = 1.3$

- Número de individuos en el grupo élite $|B| = 10$
- El número de contenedores a eliminar n_b en el operador de mutación cuando la solución fue clonada es calculado utilizando una tasa de cambio $k = 4$
- Edad máxima para que un individuo pueda ser clonado $max_edad = 10$

Los valores de los primeros dos parámetros se establecieron considerando los valores utilizados por otros algoritmos genéticos para BPP; donde se utiliza un tamaño de la población de 100 individuos y un número máximo de generaciones entre 1000 y 10000 [Falkenauer96, Singh07, Rohlfshagen10]. Los valores de los demás parámetros fueron configurados por medio de un enfoque experimental [Rangel09] que se basa en covering arrays para obtener un conjunto mínimo representativo de las posibles configuraciones para evaluar el impacto de los parámetros del algoritmo. En la Sección 6.3.3, se analiza el impacto de algunos valores de los parámetros en el rendimiento global del algoritmo.

Una comparación entre un nuevo algoritmo y varios algoritmos previamente reportados no siempre se puede hacer directamente, debido a la variabilidad de los diferentes procesadores utilizados en las pruebas computacionales. En este trabajo se aborda este problema mediante el cálculo de factores de escala, que se utilizan para comparar el rendimiento de los algoritmos como si estuvieran ejecutándose en el mismo equipo. Para la determinación de los factores de escala de cada procesador, se calcula el promedio de velocidad del CPU de los datos reportados en el punto de referencia estándar de SPEC. Debido a que los procesadores que se utilizan en la comparación se presentan en diferentes puntos de referencia SPEC, se requiere un tratamiento preliminar para el escalado entre los puntos de referencia.

La Tabla 6.3 presenta los factores de escala para cada uno de los procesadores utilizados para ejecutar los algoritmos de referencia, que es una relación de la velocidad del CPU y la velocidad del procesador usado para ejecutar GGA-CGT (Velocidad del CPU = 12, según SPEC2006). Para los experimentos presentados en esta sección, las últimas filas

de las tablas comparativas muestran los tiempos de ejecución a escala, que son calculados multiplicando los tiempos reportados en la literatura por sus correspondientes factores presentados en la Tabla 6.3 (ver puntos de referencia de velocidad de CPU disponibles en <http://www.spec.org/results>).

Tabla 6.3 Factores de escala utilizados para comparar el desempeño de GGA-CGT con el estado del arte.

Algoritmo	Procesador	Velocidad del CPU	Factor
HGGA	R4000 Silicon Graphics workstation 50 MHz	0.097 (SPEC92→SPEC2006)	$0.0972/12 = 0.008$
HI_BP	Pentium IV 1.7 GHz	3.497(SPEC2000→SPEC2006)	$3.497/12 = 0.291$
Pert.-SAWMBS	Intel core2 Quad CPU Q8200 2.33 GHz	18.0 (SPEC2006)	$18/12 = 1.5$
GGA-CGT	Intel Core2 Duo processor E6300 1.86 GHz	12 (SPEC2006)	$12/12 = 1$

6.3.1 COMPARACIÓN DE GGA-CGT CON UN GENÉTICO DE AGRUPACIÓN

En la Sección 6.1, se compararon los operadores genéticos formulados en este trabajo con los propuestos en otros enfoques evolutivos para BPP, demostrando la superioridad de los nuevos métodos: FF-ñ, Cruzamiento_Nivel_Gen y Mutación_Adaptativa. En esta sección se compara el algoritmo aquí propuesto, GGA-CGT, con el algoritmo híbrido de agrupación para BPP presentado por Falkenauer, HGGA, el cual incluye una estrategia de optimización local que se inspira en el criterio de dominación [Falkenauer96]. La Tabla 6.4 resume los resultados de HGGA así como los obtenidos por GGA-CGT sobre las instancias propuestas por Falkenauer. Las Columnas 1 y 2 enumeran los conjuntos de instancias y el número de instancias en cada conjunto. Posteriormente se muestra, para cada GA: el número de casos en los que se obtiene una solución óptima (Columna *opt.*); el tiempo medio de ejecución en segundos (Columna *tiempo*), y el número máximo de generaciones para cada clase de instancias (Columna *max_gen*). La última fila de la tabla muestra los tiempos de ejecución a escala, calculados multiplicando los tiempos en la fila Total por sus correspondientes factores de escala, presentados en la Tabla 6.3.

De la Tabla 6.4, se puede observar que GGA-CGT es mucho más rápido y más eficaz que HGGA. A pesar de la simplicidad de GGA-CGT, el uso de operadores inteligentes y reproducción controlada permiten obtener la solución óptima para todas las instancias de Falkenauer mediante el uso de un número mínimo de generaciones. Cabe destacar que GGA-CGT es capaz de resolver de manera óptima todas las instancias de Falkenauer en menos de 250 generaciones, sin embargo, el parámetro *max_gen* ha sido configurado tomando en cuenta todas las instancias estándar.

Tabla 6.4 Comparación entre GGA-CGT con el algoritmo genético de Falkenauer [Falkenauer96].

Clase	Inst.	HGGA			GGA-CGT		
		opt.	tiempo (s)	<i>max_gen</i>	opt.	tiempo (s)	<i>max_gen</i>
u_120	20	18	381	2000	20	0.006	500
u_250	20	18	1337	2000	20	0.297	500
u_500	20	20	1015	5000	20	0.160	500
u_1000	20	20	7059	5000	20	0.583	500
t_60	20	18	47	1000	20	0.053	500
t_120	20	20	79	1000	20	0.147	500
t_249	20	20	728	2000	20	0.390	500
t_501	20	20	1663	2000	20	1.296	500
Total	160	154	1538.6	2500	160	0.366	500
Tiempo escalado			12.308			0.366	

En general, el algoritmo genético de agrupación, GGA, fue introducido por Falkenauer [Falkenauer92a]. Desde entonces, este enfoque ha sido aplicado con éxito a numerosos problemas de agrupación en diferentes campos [Agustín11, Höglund13]. En este trabajo se ha demostrado que un cruzamiento a nivel de genes es una mejor alternativa para un esquema de codificación basado en grupos (Sección 6.1.4); también se demostró que el uso de una proporción de eliminación adaptativa mejora la eficacia del operador de mutación (Sección 6.1.5). Finalmente, resultados experimentales comprobaron que el desempeño del algoritmo genético de agrupación propuesto, GGA-CGT, es mejor que el rendimiento del algoritmo de Falkenauer HGGA. Aunque los detalles exactos de implementación dependen del tipo de problema de agrupación, los operadores genéticos y la técnica de reproducción controlada propuestos son buenas alternativas para mejorar el desempeño de un algoritmo genético de agrupación, en general.

6.3.2 COMPARACIÓN DE GGA-CGT CON EL ESTADO DEL ARTE

Para investigar la eficacia de la heurística GGA-CGT, se compararon los resultados obtenidos con este algoritmo con los obtenidos por los mejores enfoques en la literatura: HI_BP [Alvim04] y Perturbation-SAWMBS [Fleszar11]. También se ejecutó el código (en C) de HI_BP suministrado por los autores sobre la clase Hard28. La tabla 6.5 muestra, para cada clase de instancias, el número de instancias y, para cada enfoque de solución, el número de instancias en las que el algoritmo obtiene una solución óptima (Columna opt.). La Tabla 6.5 también muestra el tiempo medio de ejecución, medido en segundos (Columna tiempo) y los tiempos de ejecución a escala (véase la Tabla 6.3). En todos los casos, cuando un algoritmo no logra encontrar una solución óptima, la solución encontrada tiene un contenedor más en relación a la solución óptima.

Tabla 6.5 Resultados obtenidos por los mejores algoritmos de solución de BPP (tiempo en segundos).

Clase	Inst.	HI_BP		Pert.-SAWMBS		GGA-CGT	
		opt.	tiempo (s)	opt.	tiempo (s)	opt.	tiempo (s)
Uniform	80	80	0.03	79	0.00	80	0.23
Triplets	80	80	0.98	80	0.00	80	0.41
Data Set 1	720	720	0.19	720	0.01	720	0.35
Data Set 2	480	480	0.01	480	0.00	480	0.12
Data Set 3	10	10	4.60	10	0.16	10	1.99
Was 1	100	100	0.02	100	0.00	100	0.00
Was 2	100	100	0.02	100	0.01	100	1.07
Gau 1	17	12	0.60	16	0.04	16	0.27
Hard28	28	5	--	5	0.24	16*	2.40
Total	1615	1587	0.77	1590	0.05	1602	0.35
Tiempo escalado			0.224		0.075		0.35

* GGA-CGT supera a los mejores algoritmos en la clase hard28

De la Tabla 6.5, puede observarse la superioridad del algoritmo propuesto GGA-CGT sobre la clase de instancias Hard28, que contiene las instancias más difíciles para los algoritmos conocidos. Es importante remarcar que la Perturbation-SAWMBS de Fleszar y Charalambous [Fleszar11] es un enfoque heurístico eficiente que logra resultados casi idénticos a GGA-CGT en todas las clases de instancias y en una fracción del tiempo. Sin embargo, GGA-CGT supera significativamente a Perturbation-SAWMBS en el conjunto Hard28. Para este grupo de instancias, Fleszar y Charalambous [Fleszar11] reportaron

que, incluso aumentando el número de iteraciones de Perturbation-SAWMBS de 2000 a 100000, no es posible mejorar las soluciones encontradas, haciendo hincapié en la dificultad de estas instancias y recomendando su uso para futuros estudios de heurísticas. Resultados similares fueron obtenidos al incrementar el tiempo de la búsqueda Tabú en el procedimiento HI_BP.

Para evaluar la estabilidad de GGA-CGT, de forma similar a Alvim et al. [Alvim04] y Fleszar y Charalambous [Fleszar11], se realizó una prueba de robustez. El experimento consistió en ejecutar, para cada una de las 1615 instancias, cinco corridas del algoritmo con diferentes semillas de números aleatorios (8075 corridas). GGA-CGT encontró las soluciones óptimas en 7997 corridas, perdiendo la solución óptima en sólo 78 casos; en contraste, HI_BP [Alvim04] y la Perturbation-SAWMBS [Fleszar11] fallaron en la obtención de las soluciones óptimas en 144 y 128 casos, respectivamente (los números fueron ajustados considerando el conjunto Hard28). Se puede concluir que el algoritmo GGA-CGT muestra una alta precisión y robustez.

6.3.3 ANÁLISIS DEL DESEMPEÑO DE GGA-CGT

Se realizó un análisis experimental para estudiar el desempeño global de GGA-CGT y observar el impacto de algunas de las principales estrategias que se incluyen en él. Se estudió el rendimiento global y la eficacia del algoritmo en una ejecución a largo plazo. También se analizó la influencia del grupo élite B y la tasa de cambio k utilizada por el operador de mutación. Los resultados experimentales obtenidos son presentados en los siguientes párrafos, en cada experimento, si no se especifica el valor de un parámetro, el parámetro tendrá el valor definido en la Sección 6.3.

Desempeño general de GGA-CGT. Se puede analizar el comportamiento promedio de GGA-CGT mediante la observación de la evolución de las mejores soluciones globales que se producen en cada generación, durante 20 corridas independientes del algoritmo sobre cada instancia. La Figura 6.7 muestra los perfiles de ejecución de dos instancias

hard28 (hBPP561 y hBPP645) que no pudieron ser resueltas por los algoritmos del estado del arte. Cada gráfico representa las aptitudes peor, media y mejor de las mejores soluciones globales de las 20 ejecuciones (ordenada) en cada generación de GGA-CGT (abscisa).

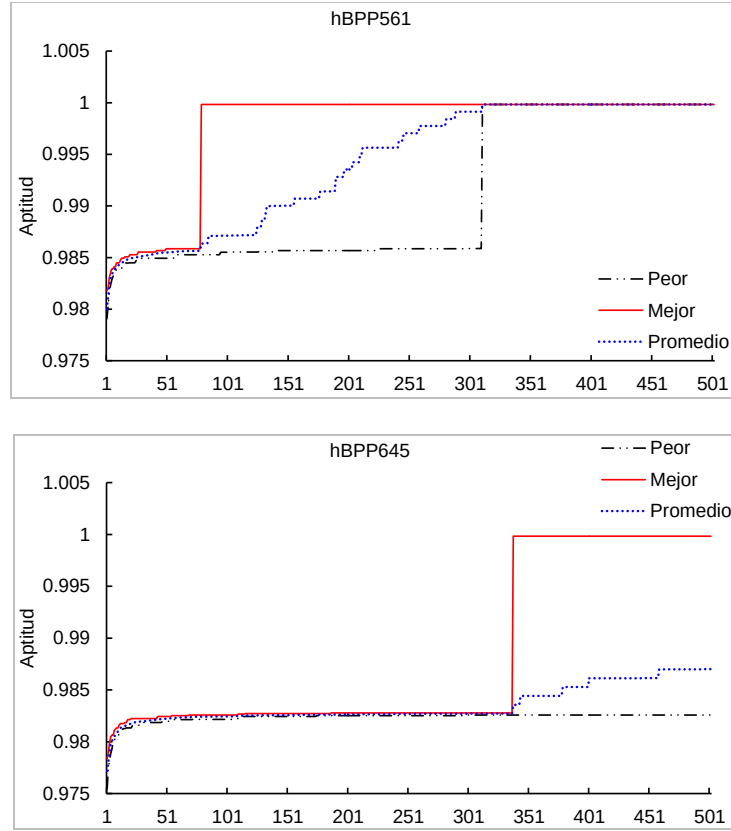


Figura 6.7 Perfiles de ejecución de GGA-CGT sobre dos instancias hard28: (a) hBPP561 y (b) hBPP645.

De la Figura 6.7, se puede observar que, antes de que alguna de las ejecuciones alcance la solución óptima ($Aptitud \approx 1$), existe una diferencia relativamente pequeña entre las peores y las mejores soluciones encontradas durante las ejecuciones de GGA-CGT. La pequeña variación en el comportamiento GGA-CGT en las diferentes ejecuciones es un indicador de la precisión y robustez del algoritmo. Es importante tener en cuenta que la fluctuación más grande en la aptitud es todavía pequeña, dada la naturaleza de la función de aptitud; un gran salto significa que el tamaño de la mejor solución se mejoró en un contenedor, que en este caso significa que la solución óptima fue encontrada después de

un período en el que los operadores genéticos trabajaron para mejorar el empaqueo de los contenedores para reducir el tamaño de las soluciones. Comportamientos similares se obtuvieron en todas las instancias de prueba.

Ejecución a largo plazo de GGA-CGT. Se investigó cómo los valores de *max_gen* influyen en el desempeño heurístico. La Tabla 6.6 resume los resultados; para cada clase de instancias y cada valor de *max_gen*, se muestra como aumenta la probabilidad de GGA-CGT de encontrar soluciones óptimas. La probabilidad media de encontrar la solución óptima aumenta de 97.39 a 99.62, cuando el número máximo de generaciones de GGA-CGT es aumentado de 50 a 10,000,000. Estos resultados demuestran que el algoritmo muestra un alto grado de precisión y robustez y es capaz de encontrar la solución óptima de una gran variedad de instancias de BPP.

Tabla 6.6 Probabilidad de encontrar soluciones óptimas con la heurística GGA-CGT.

<i>max_gen</i>	Porcentaje de soluciones óptimas para cada clase de instancias								
	U	T	Set 1	Set 2	Set 3	Was 1	Was 2	Gau 1	Hard28
50	97.50	90.00	99.72	100.00	90.00	100.00	100.00	88.23	3.57
100	97.50	96.25	99.72		100.00			88.23	28.57
500	100.00	100.00	100.00					94.11	57.14
5000								94.11	60.71
10000								94.11	64.28
100000								94.11	71.42
1000000								94.11	75.00
10000000								94.11	78.57

Influencia de la heurística FF-ñ. En la Sección 6.1.3, se mostró el efecto de explotar conocimiento sobre las características de BPP para obtener una población inicial de alta calidad. En esta sección, se muestra el impacto de la heurística FF-ñ en el desempeño del algoritmo GGA-CGT. Como primer experimento, se evaluó GGA-CGT en las instancias estándar generando la población inicial con FF en lugar de FF-ñ (véase Tabla 6.5); los resultados obtenidos mostraron que el uso de la heurística FF-ñ en lugar de FF permite que se obtenga una ventaja de 4 soluciones óptimas.

Con el objetivo de evaluar el potencial de la heurística FF- $\tilde{n}$, así como su impacto en el desempeño de GGA-CGT, se realizaron comparaciones experimentales sobre el grupo de instancias u-w1-wn (descritas en la Sección 5.5), que incluye casos de gran escala con diferentes proporciones de objetos grandes (con peso mayor que $c/2$). Para cada clase de instancias, dado un peso menor w_1 y un peso mayor w_n , los pesos de los objetos se distribuyen de manera uniforme en el intervalo $[w_1, w_n]$. La Tabla 6.7 muestra el desempeño de dos versiones del algoritmo GGA-CGT, donde la población se ha generado con FF- $\tilde{n}$ y FF, respectivamente. Cada fila indica el peso menor w_1 y cada columna indica el peso mayor w_n utilizados para generar las instancias de la clase correspondiente. Para cada clase de instancias $[w_1, w_n]$, la tabla muestra, para las dos versiones del algoritmo, el porcentaje medio de objetos grandes $\tilde{n} / n$, así como el número promedio de generaciones (Gen) utilizado por el algoritmo antes de cumplir con su criterio de paro (ver algoritmo de la Figura 6.1). Las últimas filas de la tabla muestran el número promedio de generaciones utilizados para obtener la mejor solución (Iter.) y la efectividad (Efec.), que es el porcentaje de las mejores soluciones obtenidas por cada versión del algoritmo.

Tabla 6.7 Impacto de las heurísticas FF- $\tilde{n}$ y FF en el desempeño final de GGA-CGT.

w_1	w_n														
	600			700			800			900			1000		
	$\tilde{n}/n$	Gen		$\tilde{n}/n$	Gen		$\tilde{n}/n$	Gen		$\tilde{n}/n$	Gen		$\tilde{n}/n$	Gen	
		FF- $\tilde{n}$	FF		FF- $\tilde{n}$	FF		FF- $\tilde{n}$	FF		FF- $\tilde{n}$	FF		FF- $\tilde{n}$	FF
1	0.16	0.3	0.52	0.28	1.04	2.02	0.37	2.64	4.76	0.44	39.6	44.6	0.49	75.6	87.9
100	0.2	4.88	5.36	0.33	9.02	12.3	0.42	67.1	74.6	0.49	76.4	86.9	0.55	26.6	40.4
200	0.25	48.9	58.9	0.39	100	100	0.49	48.8	77	0.57	15.5	36.1	0.62	24.2	39.7
300	0.33	49.7	51.6	0.49	12.8	40.0	0.59	6.4	29.4	0.66	6.5	29.2	0.71	12.1	27.6
400	0.49	1.2	13.7	0.66	1.28	12.7	0.74	1.18	14.4	0.79	0.1	19.5	0.83	0.9	17.2
Iter.		21.0	26.0		24.8	33.4		25.2	40.0		27.6	43.3		27.9	42.6
Efec.		1	0.99		1	0.99		1	1		1	0.99		1	0.98

De la Tabla 6.7, se puede observar que el desempeño de GGA-CGT mejora al cambiar de FF a FF- $\tilde{n}$ y que el impacto de la FF- $\tilde{n}$ es mayor cuando la proporción de objetos grandes aumenta. A pesar de su simplicidad, la estrategia de empacar primero los $\tilde{n}$ objetos grandes hace una importante contribución al desempeño del algoritmo.

Influencia del grupo elite B . El grupo elite es un elemento crítico para la técnica de reproducción propuesta, ya que controla la presión selectiva del algoritmo. A fin de examinar la influencia de este elemento en el desempeño de GGA-CGT, se realizaron algunas comparaciones experimentales utilizando diferentes tamaños para este conjunto. La Figura 6.8 muestra el número de soluciones óptimas obtenidas cuando se explora el impacto del tamaño de B para diferentes configuraciones del número de individuos a cruzar y a mutar (n_c y n_m , respectivamente).

La Figura 6.8 muestra el impacto de incluir un grupo elite en la técnica de reproducción de GGA-CGT; como puede observarse, es importante que el tamaño del grupo elite sea lo suficiente pequeño como para contribuir a la evolución sin impactar en la diversidad de la población. Estos resultados confirman la importancia de tener un balance justo entre la presión selectiva y la diversidad de la población.

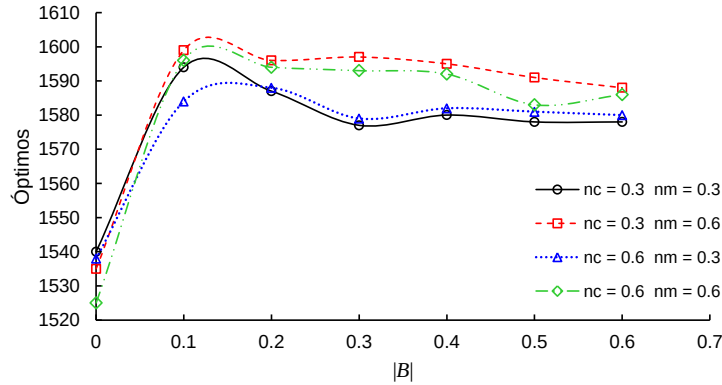


Figura 6.8 Efecto del tamaño de grupo elite $|B|$ sobre diferentes configuraciones de GGA-CGT.

Influencia de la tasa de cambio k utilizada por el operador de mutación. En este experimento, se explora el impacto de la proporción de contenedores a eliminar al aplicar el operador de mutación adaptativa. El operador de mutación se aplica a dos clases de individuos: soluciones normales y soluciones elite clonadas. Al estudiar el impacto de los parámetros GGA-CGT, se observó que la tasa de cambio k utilizada para calcular el número de contenedores a eliminar n_b en el operador de mutación debe estar configurado de forma diferente para los individuos normales y clonados. La Figura 6.9

incluye dos gráficas que muestran el impacto de diferentes valores de k cuando se aplica la mutación a individuos normales y clonados.

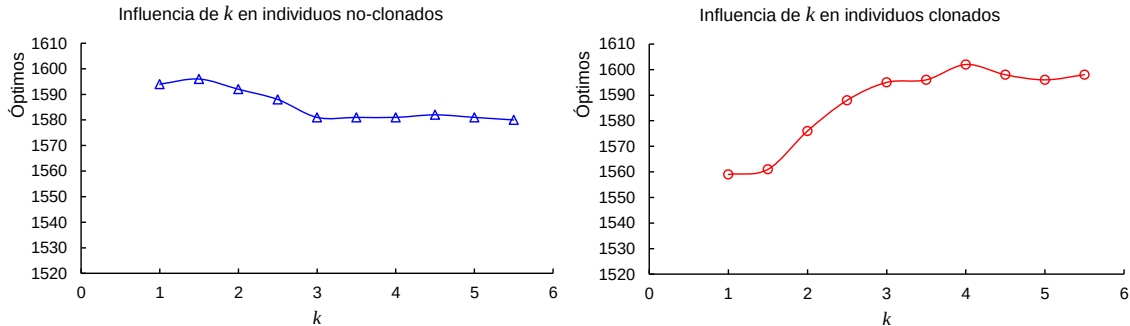


Figura 6.9 Efecto de la tasa de cambio de k en los individuos clonados y no-clonados.

De las gráficas, se observa cómo, en contraste con el caso de los individuos no-clonados, es mejor utilizar una proporción de eliminación más alta cuando se mutan soluciones clonadas. Es importante recordar que todas las soluciones clonadas provienen de soluciones élite; en este caso, una proporción más alta de eliminación en el operador de mutación permite escapar de un óptimo local y explorar otras zonas del espacio de búsqueda.

Los resultados experimentales presentados indican que GGA-CGT es un algoritmo de alta calidad que produce resultados buenos y robustos en una amplia gama de instancias de prueba con tamaños y estructuras variados, superando estrategias de solución más complicadas en un corto tiempo de ejecución. Como consecuencia de la utilización de heurísticas inteligentes que conservan las mejores características de las soluciones y una técnica de reproducción controlada, el algoritmo genético propuesto es una herramienta sencilla y eficaz para las diferentes clases de instancias de BPP.

Capítulo 7

CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se presentan las conclusiones y los resultados finales obtenidos como producto de la investigación desarrollada y se dan sugerencias para el desarrollo de trabajos futuros.

7.1 CONCLUSIONES

En el trabajo de investigación presentado en esta tesis se abordó el problema de explicar formalmente porqué un algoritmo sigue cierto comportamiento al resolver un conjunto de instancias de un problema mediante la caracterización y estudio del proceso de solución de problemas NP-duros. Se propuso un nuevo enfoque de análisis experimental de algoritmos que integra métodos de análisis multivariado de datos para obtener explicaciones sobre el desempeño observado por un algoritmo al resolver un conjunto de instancias de interés.

El enfoque propuesto fue evaluado al aplicarse en el análisis del algoritmo heurístico de alto desempeño HGGA-BP, que resuelve el problema clásico de empaqueo de objetos en contenedores, BPP. El conocimiento obtenido del estudio permitió identificar áreas de mejora en la estructura del algoritmo e incorporar nuevas estrategias que incrementaron

su desempeño. Los resultados experimentales obtenidos mostraron que, en las instancias de prueba más difíciles, el algoritmo rediseñado supera las soluciones de los mejores algoritmos del estado del arte.

La caracterización y análisis de BPP y de sus mejores algoritmos de solución, permitió identificar propiedades de las instancias que son más difíciles y caracterizarlas mediante nuevos índices. Se demostró que las características que impactan en mayor grado la dificultad de las instancias de BPP son la estructura de las soluciones óptimas y el peso promedio de los objetos (la Sección 5.7 presenta un resumen del conocimiento obtenido en el estudio del proceso de optimización de BPP). Así mismo, el conocimiento obtenido se utilizó para generar nuevos conjuntos de instancias de prueba de BPP con alto grado de dificultad. Finalmente, se diseñó un nuevo algoritmo de solución de BPP, que incorpora estrategias de solución eficientes (ver Secciones 6.1 y 6.2) para superar la efectividad de los mejores algoritmos de estado del arte.

El enfoque de análisis propuesto, los modelos de explicación generados y la aplicación de los conocimientos obtenidos confirmaron la validez de las hipótesis planteadas al inicio de esta investigación (ver Sección 1.3). El enfoque de análisis experimental (ver Sección 4.1) y su aplicación al proceso de optimización de BPP (ver Secciones 4.2, 5.2 y 5.4) permitieron comprobar las hipótesis H1 y H2. Posteriormente, los índices de caracterización propuestos fueron considerados para generar nuevos conjuntos de instancias de alto grado de dificultad (ver Secciones 5.3, 5.5 y 5.6), comprobando la hipótesis H3. Finalmente, el desempeño alcanzado por el nuevo algoritmo, GGA-CGT (ver Sección 6.3), permitió cumplir con la totalidad de los objetivos planteados y comprobar la hipótesis H4. Entre las principales aportaciones de esta investigación se encuentran:

1. Se diseñó un enfoque de análisis experimental para caracterizar y explicar el proceso de optimización algorítmico que permite comprender la dificultad de las instancias y el comportamiento de los algoritmos heurísticos.

2. Se realizó un estudio sobre la estructura del problema de empaçado de objetos en contenedores.
3. Se analizó y propuso un conjunto de índices de caracterización para el problema de empaçado de objetos en contenedores.
4. Se definió un conjunto de índices de caracterización para el comportamiento de un algoritmo heurístico que da solución a BPP. Debido a su generalidad, estos índices podrían ser aplicados a algoritmos que dan solución a otros problemas.
5. Se estudió y formuló un conjunto de índices de caracterización del desempeño que pueden ser aplicados a diferentes algoritmos heurísticos.
6. Se generaron modelos de desempeño que permitieron obtener explicaciones del comportamiento del algoritmo HGGA-BP en la solución de instancias de BPP con diferentes estructuras.
7. El conocimiento obtenido de los modelos de explicación permitió mejorar el desempeño del algoritmo HGGA-BP, incrementando su efectividad de un 87.4% a un 98.3% para un grupo de 1615 instancias estándar, superando la efectividad de los mejores algoritmos del estado del arte en la clase de instancias Hard28.
8. Se generaron tres nuevos conjuntos de instancias de prueba de BPP con un alto grado de dificultad para evaluar el desempeño de los algoritmos del estado del arte.
9. Considerando el impacto de la explotación de la capacidad de los contenedores y la importancia de dar tratamientos diferentes al acomodo de objetos grandes, medianos y pequeños para obtener una solución de calidad; se diseñó una nueva heurística de empaçado para generar la población de un algoritmo genético, optimizando el proceso de empaçado y generando soluciones de alta calidad.

10. Se diseñaron operadores genéticos de agrupación, que incorporan conocimiento de las propiedades de BPP para promover la transmisión de los mejores genes y la explotación del espacio de búsqueda.
11. Se diseñó un nuevo procedimiento de reacomodo simple para BPP que mejora las soluciones y promueve la exploración del espacio de búsqueda.
12. Se diseñó una nueva técnica de reproducción controlada que procura un balance entre la presión selectiva y la diversidad en la población.
13. Se diseñó un nuevo algoritmo genético de alto desempeño que promueve la transmisión de los mejores genes en los cromosomas, superando a la efectividad de los mejores algoritmos del estado del arte en los casos de prueba más difíciles.

7.2 TRABAJO FUTURO

Para dar continuidad al trabajo de investigación presentado se proponen los siguientes trabajos:

1. Incorporar a la caracterización del proceso de optimización avances de otras áreas, tales como: percolación y transición de fases.
2. Ampliar el conocimiento obtenido sobre la complejidad de las instancias de BPP al analizar el comportamiento de otros algoritmos y estrategias. Utilizar las explicaciones de desempeño para clasificación de instancias y pronósticos de desempeño.
3. Incorporar conocimiento del dominio del problema en los algoritmos del estado del arte de BPP para aumentar su efectividad y mejorar su desempeño.

4. Aplicar el enfoque análisis experimental en la caracterización de otra clase de problemas de optimización y otros algoritmos heurísticos.
5. Realizar un análisis que permita relacionar las características teóricas de un problema NP-duro con la estructura del espacio de búsqueda.
6. Reducir el número de parámetros que necesitan ser ajustados en el algoritmo GGA-CGT.
7. Desarrollar un nuevo esquema de codificación para el algoritmo GGA-CGT, para reducir el espacio de búsqueda generado por soluciones isomorfas; con el objetivo de mejorar el tiempo de ejecución del algoritmo.
8. Aplicar los operadores genéticos y la técnica de reproducción controlada aquí propuestos para solucionar otros problemas de agrupación.

REFERENCIAS BIBLIOGRÁFICAS

- [Achlioptas05] Achlioptas D., Naor A., Peres Y. *Rigorous location of phase transitions in hard optimization problems*. Nature, 435: 759-764, 2005.
- [Agustín11] Agustín-Blas LE, Salcedo-Sanz S, Ortiz-García EG, Portilla-Figueras A, Pérez-Bellido AM, Jiménez-Fernández S. *Team formation based on group technology: A hybrid grouping genetic algorithm approach*. Computers & Operations Research, 38(2): 484-495, 2011.
- [Alvarez06] Álvarez V. *Modelo para representar la Complejidad del problema y el desempeño de algoritmos*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México, 2006.
- [Alvim03] Alvim A. *Uma Heurística Híbrida de Melhoria para o Problema de Bin Packing e sua Aplicacao ao Problema de escalonamento de Tarefas*. Tesis Doctoral. Catholic University of Rio de Janeiro, Department of Computer Science, Rio de Janeiro, Brazil, 2003.
- [Alvim04] Alvim A., Glover F., Ribeiro C., Aloise D. *A hybrid improvement heuristic for the one-dimensional bin packing problem*. Journal of Heuristics, 10: 205-229, 2004.
- [Angel98] Angel E., Zissimopoulos V. *Autocorrelation Coefficient for the Graph Bipartitioning Problem*. Theoretical Computer Science, 191: 229-243, 1998.
- [Barr95] Barr S., Golden L., Kelly P., Resendez M., Stewart R. *Designing and Reporting on Computational Experiments with Heuristic*

Methods. Journal of Heuristics, 1(1): 9-32, 1995.

- [Basse98] Basse S. *Computer Algorithms, Introduction to Design and Analysis*. Editorial Addison-Wesley Publishing Company, 1998.
- [Beasley90] Beasley J. E. *OR-library: Distributing test problems by electronic mail*. Journal of the Operational Research Society, 41: 1069-1072, 1990. Disponible en internet via <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/binpackinfo.html>
- [Bejar01] Bejar R., Vetsikas I., Gomes C., Kautz H., Selman B. *Structure and phase transition phenomena in the VTC problem*. TASK PI Meeting Workshop, 2001.
- [Belov04] Belov G. *Problems, Models and Algorithms in One and Two Dimensional Cutting*. PHD thesis. Fakultät Mathematik und Naturwissenschaften der Technischen Universität Dresden, 2004.
- [Belov05] Belov G., Letchford A., Uchoa E. *A Node-Flow Model for 1D Stock Cutting: Robust Branch-Cut-and-Price*. University of Dresden, Lancaster University and Universidade Federal Fluminense, 2005.
- [Berberich10] Berberich E., Hagen M., Hiller B, Moser H. *Experiments*. Algorithm Engineering, Chapter 8. Lecture Notes in Computer Science, 5971: 325-388, Springer-Verlag, 2010.
- [Berzal02] Berzal F., Blanco I., Sánchez D., Vila M. *Measuring the accuracy and interest of association rules: A new framework*. Department of Computer Science and Artificial Intelligence, University of Granada, E.T.S.I.I, 2002.
- [Bhatia04] Bhatia A., Basu S. *Packing Bins Using Multi-chromosomal Genetic Representation and Better-Fit Heuristic*. Neural Information Processing, Lecture Notes in Computer Science, 3316: 181-186, 2004.
- [Birattari01] Birattari M, Paquete L., Stützle T., Varrentrapp K. *Classification of Metaheuristics and Design of Experiments for the Analysis of Components*. Technical Report AIDA-01-05, Intellektik, Technische Universität Darmstadt, Darmstadt, Germany, 2001.
- [Birattari09] Birattari M. *Tuning Metaheuristics: A machine learning perspective*. SCI 197. Springer, Berlin, Germany, 2009.

- [Blickle96] Blickle T, Thiele L. *A comparison of selection schemes used in evolutionary algorithms*. Evolutionary Computation, 4(4): 361-394, 1996.

- [Borenstein05] Borenstein Y. *Information landscapes*. Genetic And Evolutionary Computation Conference, 1515-1522, 2005.

- [Borenstein06] Borenstein Y., Poli R. *Structure and metaheuristics*. Proceedings of the 8th annual conference on Genetic and evolutionary computation, 1087-1094, 2006.

- [Borgs01] Borgs C., Chayes J., Pittel B. *Phase transition and finite-size scaling for the integer partitioning problem*. Random Structures and Algorithms 19: 247-288, 2001.

- [Brglez07] Brglez F., Osborne J. A. *Performance testing of combinatorial solvers with isomorph class instances*. Workshop On Experimental Computer Science, 2007.

- [Brown93] Brown EC, Sumichrast RT. *Impact of the replacement heuristic in a grouping genetic algorithm*. Computers & Operations Research, 30(11): 1575-1593, 2003.

- [Brown05] Brown EC, Sumichrast RT. *Evaluating performance advantages of grouping genetic algorithms*. Engineering Applications of Artificial Intelligence, 18(1): 1-12, 2005.

- [Burke02] Burke R., Gustafson S., Kendall G. *A Survey and Analysis of Diversity Measures in Genetic Programming*. Proceedings of the Genetic and Evolutionary Computation Conference, 716-723, 2002.

- [CaPaD] CaPaD 2014. Cutting and Packing at Dresden University, Benchmark data sets. Disponible en internet vía <http://www.math.tu-dresden.de/~capad/cpd-ti.html#pmp>

- [Caramanis03] Caramanis C. *Survey Propagation Iterative Solutions to Constraint Satisfaction Problems*. Expository Writing, 2003.

- [Carnegie09] Carnegie Mellon University. *Causal and Statistical Reasoning*. Open Learning Initiative, 2009. Disponible en internet vía

<http://oli.web.cmu.edu/openlearning/>

- [Chalmers99] Chalmers A. F. *What is this thing called Science?* Third Edition, Hackett Publishing Company, 1999.
- [Cheeseman91] Cheeseman P., Kanefsky B., Taylor W. *Where the really hard problems are*. Proceedings the 12th International Joint Conference on Artificial Intelligence (IJCAI-91), 331-337, 1991.
- [Chevalier06] Chevalier R. *Balancing the effects of parameter settings on a genetic algorithm for multiple fault diagnosis*. Artificial Intelligence, The university of Georgia, 2006.
- [Chiarandini07] Chiarandini M., Paquete L., Preuss M., Ridge E. *Experiments on metaheuristics: Methodological overview and open issues*. Technical Report DMF-2007-03-003, The Danish Mathematical Society, 2007.
- [Coffman97] Coffman E., Garey M., Johnson D. *Approximation algorithms for bin packing: a survey*. Approximation algorithms for NP-hard problems. D. Hochbaum, Boston, PWS Publishing, 46-93, 1997.
- [Cohen95] Cohen P. R. *Empirical Methods for Artificial Intelligence*. The MIT Press, Cambridge, Massachusetts, 1995.
- [Cook97] Cook S. Mitchell D. *Finding hard instances of the satisfiability problem: A survey*. Satisfiability Problem: Theory and Applications, vol. 35 of Dimacs Series in Discrete Mathematics and Theoretical Computer Science, 1-17, American Mathematical Society, 1997.
- [Corne03] Corne D., Oates M., Kell D. *Landscape state machines: tools for evolutionary algorithm performance analyses and landscape / algorithm mapping*. Lecture Notes In Computer Science, Proceedings of the 2003 international conference on Applications of evolutionary computing, pages: 187-198, Springer, 2003.
- [Cruz04] Cruz L. *Caracterización de Algoritmos Heurísticos Aplicados al Diseño de Bases de Datos Distribuidas*. Tesis de doctorado, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Morelos, México, 2004.
- [Cruz07] Cruz L., Nieto-Yáñez D., Rangel-Valdez N., Herrera J., González J.,

- Castilla G., J., Delgado-Orta F. *DiPro: An Algorithm for the Packing in Product Transportation Problems with Multiple Loading and Routing Variants*. MICAI 2007: Advances in Artificial Intelligence, Lecture Notes in Computer Science, 4827: 1078-1088, 2007.
- [Cruz-Reyes12] Cruz-Reyes, L., Quiroz, M., Alvim, A., Fraire H., Gómez, C., & Torres-Jiménez, J. Heurísticas de agrupación híbridas eficientes para el problema. *Computación y Sistemas*, 16(3): 349-360, 2012.
- [Czogalla09] Czogalla J., Fink A. *Fitness Landscape Analysis for the Resource Constrained Project Scheduling Problem*. Lecture Notes in Computer Science, Learning and Intelligent Optimization 5851: 104-118, Springer, 2009.
- [Denning89] Denning P., Comer D., Gries D., Mulder M., Tucker A., Turner A., Young P. *Computing as a discipline*. *Computer* 22(2): 63-70, 1989.
- [Denning05] Denning P. *Is computer science science?* *Communications of the ACM - Transforming China*, 48(4): 27-31, 2005.
- [Díaz96] Díaz A., González J., Laguna M., Mascato P., Tseng T., Glover F., Ghaziri M. *Optimización Heurística y Redes Neuronales*. Editorial Parainfo, España, 1996.
- [Dréo06] Dréo J., Pétrowski A., Siarry P., Taillard E. *Metaheuristics for Hard Optimization Methods and Case Studies*. Springer-Verlang Berling Heidelberg, 2006.
- [ESICUP] ESICUP 2014. Euro Especial Interest Group on Cutting and Packing, One Dimensional Cutting and Packing Data Sets. Disponible en internet vía http://paginas.fe.up.pt/~esicup/tiki-list_file_gallery.php?galleryId=1
- [Falkenauer92a] Falkenauer E. *The grouping genetic algorithm—widening the scope of the GAs*. *Belgian Journal of Operations Research, Statistics and Computer Science*, 33: 79-102, 1992.
- [Falkenauer92b] Falkenauer E., Delchambre A. *A Genetic Algorithm for Bin Packing and Line Balancing*. *Proc. of the IEEE International Conference on Robotics and Automation (RA92)*, 1992.
- [Falkenauer96] Falkenauer E. *A Hybrid Grouping Genetic Algorithm for Bin*

- Packing*. Journal of Heuristics 2: 5-30, 1996.
- [Feitelson06] Feitelson D. G. *Experimental Computer Science: The Need for a Cultural Change*. 2006. Disponible en internet vía <http://www.cs.huji.ac.il/~feit/papers/exp05.pdf>
- [Ferat07] Ferat E. *Formación Dinámica de Regiones de Dominación de Algoritmos Heurísticos*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México, 2007.
- [Fleszar02] Fleszar K., Hindi K. *New heuristics for one-dimensional bin-packing*. Computers & Operations Research, 29(7): 821-839, 2002.
- [Fleszar11] Fleszar K, Charalambous C. *Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem*. European Journal of Operational Research, 210(2): 176-184, 2011.
- [Frank95] Frank J., Martel C. *Phase transitions in the properties of random graphs*. Principles and Practice of Constraint Programming (CP-95), Cassis, France, 1995.
- [Gagliolo07] Gagliolo M., Schmidhuber J. *Learning Dynamic Algorithm Portfolios*. AI & MATH 2006, Special Issue of the Annals of Mathematics and Artificial Intelligence, 47(3-4): 295-328, presented at LION, 2007.
- [García04] García N. *Análisis de las Propiedades del Problema de la Distribución de Bases de Datos y su Influencia en el Desempeño de Algoritmos Heurísticos*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México, 2004.
- [García10] García R. *Hiper-heurístico para resolver el problema de cartera de proyectos sociales*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México, 2010.
- [Garey79] Garey M. R., Jonson D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company. A classic introduction to the field, 1979.
- [Gazis10] Gazis P., Levit C., Way M. *Viewpoints: A High-Performance High-Dimensional Exploratory Data Analysis Tool*. Publications of the Astronomical Society of the Pacific, 122, 1518. 2010.

- [Gent95] Gent I., Walsh T. *The number partition phase transition*. Technical Report RR-95-185, Department of Computer Science, University of Strathclyde, Glasgow G1 1XH, Scotland, 1995.
- [Gent96] Gent I., Walsh T. *Phase transitions and annealed theories: Number partitioning as a case study*. Proceedings of the 12th ECAI, 1996.
- [Gent99] Gent I. P., Walsh T. Paul R. *Book Review: Cohen's Empirical Methods for Artificial Intelligence*. Artificial Intelligence 113: 285-290, Elsevier Science B.V., 1999.
- [Gen00] Gen M, Cheng R. *Genetic Algorithms and Engineering Optimization*. New York: John Wiley & sons, 2000.
- [Gonzalez07] Gonzalez T. *Handbook of Approximation Algorithms and Metaheuristics*. Chapman & Hall/CRC, Taylor & Francis Group, 2007.
- [Gómez-Meneses09] Gómez-Meneses P., Randall M. *A Hybrid Extremal Optimisation Approach for the Bin Packing Problem*. Artificial Life: Borrowing from Biology, Lecture Notes in Computer Science, 5865: 242-251, 2009.
- [Gupta99] Gupta J., Ho J. *A new heuristic algorithm for the one-dimensional bin packing problem*. Production Planning & Control: The Management of Operations, 1366-5871, 10(6): 598-603, 1999.
- [Hartwig79] Hartwig F., Dearing B. E. *Exploratory Data Analysis*. Sage University Paper Series on Quantitative Research Methods, Vol. 16, Newbury Park, CA: Sage, 1979.
- [Hien06] Hien N., Hoai N. *A Brief Overview of Population Diversity Measures in Genetic Programming*. Proceeding of 11th Asia-Pacific Workshop on Intelligent and Evolutionary Systems, 128-139, 2006.
- [Hogg94] Hogg T., Williams C. *The hardest constraint problems: a double phase transition*. Artificial Intelligence 69(1-2): 359-377, 1994.
- [Höglund13] Höglund H. *Estimating discretionary accruals using a grouping genetic algorithm*. Expert systems with applications, 40(7): 2366-

2372, 2013.

- [Holland75] Holland J. H. *Adaption in Natural and Artificial Systems*. The University of Michigan Press, Ann Harbor, MI, 1975.
- [Hooker94] Hooker J. N. *Needed: An empirical science of algorithms*. Operations Research, 42(2): 201-212, 1994.
- [Hooker95] Hooker J. N. *Testing Heuristics: We have it all wrong*. Journal of Heuristics, 1: 33-42, 1995.
- [Hoos04] Hoos H., Smyth K., Stützle T. *Search Space Features Underlying the Performance of Stochastic Local Search Algorithms for MAX-SAT*. Parallel Problem Solving from Nature VIII, Lecture Notes in Computer Science, 3242: 51-60, 2004.
- [Hoos07] Hoos H. H., Stützle T. *Empirical Analysis of Randomized Algorithms*. Handbook of Approximation Algorithms and Metaheuristics. Chapman & Hall/CRC, Taylor & Francis Group, 2007.
- [Hutter05] Hutter F., Hamadi Y. *Parameter adjustment based on performance prediction: Towards an instance-aware problem solver*. Microsoft Research, Technical Report MSR-TR-2005-125, 2005.
- [Jansen01] Jansen T. *On classifications of fitness functions*. Natural Computing Series, Theoretical aspects of evolutionary computing, 371-385, 2001.
- [Johnson74] Johnson DS. *Fast algorithms for bin packing*. Journal of Computer and System Sciences, 8(3): 272-314, 1974.
- [Johnson00] Johnson D. *Método Multivariados Aplicados al Análisis de Datos*. Primera Edición. International Thomson Editores, 2000.
- [Jones95] Jones T. *Evolutionary algorithms, fitness landscapes and search*. Ph.D. thesis, The University of New Mexico, Albuquerque, New Mexico, 1995.
- [Judd09] Judd C., McClelland G., Ryan C. *Data analysis: A model comparison approach*. New York, NY, US: Routledge/Taylor & Francis Group,

2009.

- [Klein09] Klein R., Scholl A. *Bin Packing*. Disponible en internet vía <http://www.wiwi.uni-jena.de/Entscheidung/binpp/>
- [Landro04] Landero V. *Análisis de las Regiones de Dominación de Algoritmos Heurísticos Aplicados al Diseño de Bases de Datos Distribuidas*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México, 2004.
- [Landro08] Landero V. *Desarrollo de un Método Formal que Muestre la Interrelación entre las Características de un Conjunto de Casos y las de un Algoritmo que los Resuelve Eficientemente, para el Problema de Distribución de Objetos en Contenedores*. Tesis de doctorado, Dpto. de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor., 2008.
- [Lemeire07] Lemeire J. *Learning Causal Models of Multivariate Systems and the Value of it for the Performance Modeling of Computer Programs*. PhD thesis, Vrije Universiteit Brussel, 2007.
- [Levine04] Levine J., Ducatelle F. *Ant colony optimization and local search for bin packing and cutting stock problems*. Journal of the Operational Research Society, 55: 705-716, 2004.
- [Lewis09] Lewis R. *A general-purpose hill-climbing method for order independent minimum grouping problems: A case study in graph colouring and bin packing*. Computers & Operations Research, 36(7): 2295-2310, 2009.
- [Leyton-Brown02] Leyton-Brown K., Nudelman E., Shoham Y. *Learning the Empirical Hardness of Optimization Problems: the case of combinatorial auctions*. Principles and Practice of Constraint Programming (CP-02) 2470: 556-572, Ithaca, 2002.
- [Liu96] Liu X. *Intelligent Data Analysis: Issues and Challenges*. The Knowledge Engineering Review, 11: 365-371, 1996.
- [Loh08] Loh K., Golden B., Wasil E. *Solving the one-dimensional bin packing problem with a weight annealing heuristic*. Computers & Operations Research 35: 2283-2291, 2008.

-
- [Lozano08] Lozano M, Herrera F, Cano JR. Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Information Sciences*, 178(23): 4421-4433, 2008.
- [Madani09] Madani O., Raghavan H., Jones R. *On the Empirical Complexity of Text Classification Problems*. SRI AI Center Technical Report, 2009.
- [Martello90a] Martello S., Toth P. *Lower Bounds and Reduction Procedures for the Bin Packing Problem*. *Discrete Applied Mathematics*, 22: 59-70, North-Holland, Elsevier Science Publishers B.V., 1990.
- [Martello90b] Martello S., Toth P. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley & Sons Ltd., 1990.
- [Martin01] Martin O., Monasson R., Zecchina R. *Statistical mechanics methods and phase transitions in optimization problems*. *Theoretical Computer Science*, 265: 3-67, 2001.
- [McGeoch92] McGeoch C. C. *Analysis of algorithms by simulation: variance reduction techniques and simulation speedups*. *ACM Computing Surveys* 24(5): 195-212, 1992.
- [McGeoch02] McGeoch C.C. *Experimental Analysis of Algorithms*. Pardalos, P.M., Romeijn, *Handbook of Global Optimization*, 2: 489-513, 2002.
- [McGeoch07] McGeoch C.C. *Experimental Algorithmics*. *Communications of the ACM*, 50 (11): 27-31, 2007.
- [McKay01] McKay R., Abbass H. *Anti-correlation Measures in Genetic Programming*. *Proceedings of the Australasia-Japan Workshop on Intelligent and Evolutionary Systems*, pages: 45-51, University of Otago, Dunedin, New Zealand, 2001.
- [Menzies09] Menzies T., Mizuno O., Takagi Y., Kikuno T. *Explanation vs Performance in Data Mining: A Case Study with Predicting Runaway Projects*. *J. Software Engineering & Applications*, 2: 221-236, 2009.
- [Mertens06] Mertens S. *The easiest hard problem: Number partitioning*. In A.G. Percus, G. Istrate, and C. Moore, editors, *Computational Complexity and Statistical Physics*, Oxford University Press, 125-139, New York,

2006.

- [Merz00a] Merz P., Freisleben B. *Fitness Landscapes, Memetic Algorithms, and Greedy Operators for Graph Bipartitioning*. Evolutionary Computation 8(1): 61-91, 2000.
- [Merz00b] Merz P., Freisleben B. *Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem*. IEEE Transactions on Evolutionary Computation 4(4): 337-352, 2000.
- [Merz04] Merz P. *Advanced fitness landscape analysis and the performance of memetic algorithms*. Evolutionary Computation, Special issue on magnetic algorithms, 12(3): 303-325, 2004.
- [Messelis09] Messelis T., Haspeslagh S., Bilgin B., De Causmaecker P., Vanden Berghe G. *Towards prediction of algorithm performance in real world optimisation problems*. Proceedings of the 21st Benelux Conference on Artificial Intelligence, BNAIC, Eindhoven, pages 177-183, 2009.
- [Michalewicz04] Michalewicz Z., Fogel D. B. *How to Solve It: Modern Heuristics*. 2nd edition, Springer-Verlang, 2004.
- [Mitchell92] Mitchell M., Forrest S., Holland J. *The royal road for genetic algorithms: Fitness landscapes and GA performance*. Proceedings of the first European Conference on Artificial Life, (MIT Press, Cambridge, MA), 245-254, 1992.
- [Mitchell03] Mitchell B., Mancoridis S. *Modeling the search landscape of metaheuristic software clustering algorithms*. Lecture Notes in Computer Science, Proceedings of the 2003 international conference on Genetic and evolutionary computation: Part II, 2499-2510, Springer, 2003.
- [Moret02] Moret B. *Towards a discipline of experimental algorithmics*. Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges, M. H. Goldwasser, D. S. Johnson, C. C. McGeoch (Eds.), 197-213, 2002.
- [Morrison10] Morrison C., Snodgrass R. *Computer science can use more science*. University of Arizona, October 27, 2010. Disponible en internet vía <http://www.cs.arizona.edu/~rts/pubs/CACM2011.pdf>

- [Müller-Hannemann10] Müller-Hannemann M., Schirra S. *Challenges in Algorithm Engineering*. Algorithm Engineering, Chapter 10. Lecture Notes in Computer Science, 5971: 446-453, Springer-Verlag, 2010.
- [Nieto07] Nieto D. *Hibridación de Algoritmos Metaheurísticos para Problemas de Bin Packing*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México, 2007.
- [Norsys09] Norsys Software Corp. *Netica Application*. 2009.
- [Nudelman04] Nudelman E., Devkar A., Shoham Y., Leyton-Brown K. *Understanding random SAT: Beyond the clauses-to-variables ratio*. Principles and Practice of Constraint Programming CP'04, 3258: 438–452, 2004.
- [Ochoa09] Ochoa G., Qu R., Burke E. *Analyzing the landscape of a graph based hyper-heuristic for timetabling problems*. Proceedings of the 11th Annual conference on Genetic and evolutionary computation, 341-348, 2009.
- [Ott10] Ott L., Longnecker M. *An introduction to statistical methods and data analysis*. Brooks/Cole, Cengage Learning, 2010.
- [Pérez07] Pérez V. *Modelado Causal del Desempeño de Algoritmos Metaheurísticos en Problemas de Distribución de Objetos*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México, 2007.
- [Pérez08] Pérez J., Cruz L., Pazos R., Landero V., Reyes G., Zavala C., Fraire H., Pérez V. *A Causal Approach for Explaining Why a Heuristic Algorithm Outperforms Another in Solving an Instance Set of the Bin Packing Problem*. Lectures Notes in Computer Science, 4994: 591-598, Springer-Verlag, Berlin Heidelberg New York, 2008.
- [Piñol08] Piñol C. *CSP Problems as Algorithmic Benchmarks: Measures, Methods and Models*. Universitat de Lleida, Departament d'Informàtica i Enginyeria Industrial, 2008.
- [Rangel09] Rangel-Valdez N, Torres-Jimenez J, Bracho-Rios J, Quiz-Ramos P. Problem and algorithm fine-tuning - a case of study using bridge club and simulated annealing. Proceedings of the International Joint Conference on Computational Intelligence (IJCCI'09). Madeira:

Springer, 302–305, 2009.

- [Rohlfshagen10] Rohlfshagen P, Bullinaria JA. *Nature inspired genetic algorithms for hard packing problems*. Annals of Operations Research, 179: 393-419, 2010.
- [Santiago04] Santiago E. *Modelo Estadístico para la Caracterización de Casos del Problema del Diseño de la Distribución*. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México, 2004.
- [Schneider03] Schneider J. *Searching for Backbones—a high-performance parallel algorithm for solving combinatorial optimization problems*. Future Generation Computer Systems, 19(1): 121-131, 2003.
- [Schoenfield02] Schoenfield J. E. *Fast, exact solution of open bin packing problems without linear programming*. Draft, US Army Space & Missile Defense Command, Huntsville, Alabama, USA, 2002.
- [Scholl97] Scholl A., Klein R., Jürgens C. *BISON: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem*. Computers and Operations Research, 24: 627-645, 1997.
- [Schwerin97] Schwerin P., Wäscher G. *The bin-packing problem: A problem generator and some numerical experiments with FFD packing and MTP*. International Transactions in Operational Research, 4: 337-389, 1997.
- [Schwerin99] Schwerin P., Wäscher G. *A new lower bound for the bin-packing problem and its integration to MTP*. Pesquisa Operacional, 19: 111-129, 1999.
- [Singh07] Singh A, Gupta AK. Two heuristics for the one-dimensional bin-packing problem. OR Spectrum 2007; 29(4): 765-781.
- [Slaney01] Slaney J., Walsh T. *Backbones in optimization and approximation*. Proceedings the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), 254–259, 2001.
- [Smith07] Smith J. *On replacement strategies in steady state evolutionary algorithms*. Evolutionary Computation, 15(1): 29-59, 2007

- [Smith-Miles09] Smith-Miles K., James R., Giffin J., Tu Y. *Understanding the relationship between scheduling problem structure and heuristic performance using knowledge discovery*. In Learning and Intelligent Optimization, LION 3, 2009.
- [Smyth04] Smyth K. *Understanding Stochastic Local Search Algorithms: An Empirical Analysis of the Relationship Between Search Space Structure and Algorithm Behaviour*. Tesis de maestría. The University of British Columbia, Department of Computer Science, Vancouver, Canada, 2004.
- [Snodgrass10] Snodgrass R. *Ergalics: A Natural Science of Computation*. February 2010. Disponible en internet vía <http://www.cs.arizona.edu/projects/ergalics/whatis.html>
- [Stawowy08] Stawowy A. *Evolutionary based heuristic for bin packing problem*. Computers & Industrial Engineering, 55(2): 465-474, 2008.
- [Stützle04] Stützle T., Fernandes S. *New Benchmark Instances for the QAP and the Experimental Analysis of Algorithms*. Lecture Notes in Computer Science, volume 3004, Evolutionary Computation in Combinatorial Optimization, 199-209, Springer, 2004.
- [Sugihara97] Sugihara K. *A Case Study on Tuning of Genetic Algorithms by Using Performance Evaluation Based on Experimental Design*. Technical Report ICS-TR-97-01, Dept. of Information and Computer Sciences, Univ. of Hawaii at Manoa, Sugiharaz, 1997.
- [Thierens02] Thierens D. *Predictive measures for problem representation and genetic operator design*. Technical report UU-CS-2002-055, Utrecht University: Information and Computing Sciences, 2002.
- [Tukey77] Tukey J. W. *Exploratory Data Analysis*. Addison-Wesley, 1977.
- [Vandegriend98] Vandegriend B., Culberson J. *The $g_{n,m}$ phase transition is not hard for the hamiltonian cycle problem*. J. Artificial Intelligence Research 9: 219-245, 1998.
- [Wäscher96] Wäscher G., Gau T. *Heuristics for the one-dimensional cutting stock problem: A computational study*. OR Spektrum, 18: 131-144, 1996.
- [Watson03] Watson J., Beck J., Howe A., Whitley L. *Problem difficulty for tabu*

- search in job-shop scheduling*. Artificial Intelligence, 143: 189-127, 2003.
- [Weinberger90] Weinberger E. *Correlated and uncorrelated fitness landscapes and how to tell the difference*. Biological Cybernetics, 63: 325-336, 1990.
- [Wolpert97] Wolpert D. H., Macready W.G. *No Free Lunch Theorems for Optimizations*. IEEE Transactions on Evolutionary Computation, 1: 67-82, 1997.
- [Wright32] Wright S. *The roles of mutation, inbreeding, crossbreeding and selection in evolution*. Proceedings of the sixth international genetics 1: 356-366, 1932.
- [Yesil12] Yesil C, Turkyilmaz H, Korkmaz EE. *A new hybrid local search algorithm on Bin Packing problem*. Proceedings of the 12th International Conference on Hybrid Intelligent Systems (HIS'12). Pune: IEEE, 161-166, 2012.
- [Zeng09] Zeng G., Lu Y. *Survey on computational complexity with phase transitions and extremal optimization*. Proceedings of 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference Shanghai, P.R. China, December 16-18, 2009.
- [Zhang00] Zhang BT, Kim JJ. *Comparison of selection methods for evolutionary optimization*. Evolutionary Optimization, 2(1): 55-70, 2000.
- [Zhang04] Zhang W. *Configuration landscape analysis and backbone guided local search.: Part I: Satisfiability and maximum satisfiability*. Artificial Intelligence, 158(1): 1-26, 2004.
- [Zhang05] Zhang W., Look M. *A novel local search algorithm for the travelling salesman problem that exploits backbones*. Proceedings of the 19th International Joint Conference on Artificial Intelligence, pages: 343-350, 2005.