



**INSTITUTO TECNOLÓGICO
DE CD. MADERO**



DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



***“CARACTERIZACIÓN DE FACTORES DE DESEMPEÑO
DE ALGORITMOS DE SOLUCIÓN DE BPP”***

TESIS
PARA OBTENER EL GRADO DE
MAESTRA EN CIENCIAS EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:
I.S.C. MARCELA QUIROZ CASTELLANOS

DIRECTOR DE TESIS:
DRA. LAURA CRUZ REYES

CO-DIRECTOR DE TESIS:
DRA. ADRIANA CESÁRIO DE FARIA ALVIM

CD. MADERO, TAMPS., MEX.

NOVIEMBRE 2009

SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO



SECRETARÍA DE
EDUCACIÓN PÚBLICA

SEP

Cd. Madero, Tamps; a 19 de Noviembre de 2009.

OFICIO No.: U5.419/09
AREA: DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

**C. ING. MARCELA QUIROZ CASTELLANOS
P R E S E N T E**

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestría en Ciencias en Ciencias de la Computación, se acordó autorizar la impresión de su tesis titulada:

"CARACTERIZACIÓN DE FACTORES DE DESEMPEÑO DE ALGORITMOS DE SOLUCIÓN DE BPP"

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

ATENTAMENTE
"Por mi Patria y por mi Bien"

Ma. Yolanda Chávez Cinco
M. P. MARÍA YOLANDA CHÁVEZ CINCO
JEFA DE LA DIVISIÓN



c.c.p.: Archivo

MYCHC 'MICO 'aygc*

"2009, Año de la Reforma Liberal"

Ave. 10. De Mayo y Sor Juana I. De la Cruz, Col. Los Mangos, C.P. 89440 Cd. Madero, Tam.
Tels. (833) 3 57 48 24, Fax: (833) 3 57 48 20, Ext. 1002, email: itcm@itcm.edu.mx
www.itcm.edu.mx

RESUMEN

En muchas organizaciones existen un gran número de problemas de optimización complejos, en los cuales se busca una asignación eficiente de recursos con la finalidad de satisfacer los objetivos planteados por la entidad. Para contribuir en ello, es necesario resolver estos problemas 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, lo cual implica que no se conocen algoritmos eficientes para resolverlos de manera exacta en el peor caso.

Para la solución de problemas NP-duros, se ha propuesto una gran variedad de algoritmos aproximados, los cuales han mostrado un desempeño satisfactorio en la solución de problemas de optimización. A pesar de los esfuerzos de muchos investigadores en este ámbito, 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 parece ser la única opción. Un obstáculo que se presenta al analizar el comportamiento de este tipo de algoritmos es identificar cuáles estrategias hacen que un algoritmo muestre un mejor desempeño y bajo que condiciones lo obtienen.

Este trabajo plantea el estudio integral del proceso de optimización, con la finalidad de identificar relaciones inherentes entre los factores que afectan al desempeño algorítmico. Se espera que con el conocimiento obtenido en dicho análisis sea posible diseñar estrategias de mayor impacto, especialmente, en el desarrollo de algoritmos de solución de alto desempeño para el problema de empaqueo de objetos en contenedores (Bin Packing Problem, BPP).

CONTENIDO

RESUMEN I

CAPÍTULO 1	1
1.1 ANTECEDENTES	2
1.2 DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN	3
1.3 HIPÓTESIS	4
1.4 JUSTIFICACIÓN	5
1.5 OBJETIVOS	6
1.5.1 OBJETIVO GENERAL	6
1.5.2 OBJETIVOS ESPECÍFICOS	6
1.6 ALCANCES Y LIMITACIONES	7
1.7 ORGANIZACIÓN DEL DOCUMENTO.....	7
CAPÍTULO 2	9
2.1 PROBLEMA DE EMPACADO DE OBJETOS EN CONTENEDORES.....	10
2.2 ALGORITMOS APROXIMADOS PARA LA SOLUCIÓN DE BPP	10
2.2.1 ALGORITMOS DETERMINISTAS	12
2.2.2 ALGORITMOS METAHEURÍSTICOS	13
2.3 ANÁLISIS EXPERIMENTAL DE ALGORITMOS METAHEURÍSTICOS	16
2.4 CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN	19
2.4.1 PROCESO DE OPTIMIZACIÓN	19
2.4.2 FORMULACIÓN DE FUNCIONES DE CARACTERIZACIÓN	20
2.4.3 REFINACIÓN DE FUNCIONES DE CARACTERIZACIÓN	23
2.4.4 EXPLICACIÓN DEL DESEMPEÑO ALGORÍTMICO	25
2.5 RELACIONES DE DESEMPEÑO ALGORÍTMICO	26
2.6 CONOCIMIENTO OBTENIDO EN EL ANÁLISIS DE DESEMPEÑO.....	30
CAPÍTULO 3	32
3.1 ALGORITMOS METAHEURÍSTICOS PARA LA SOLUCIÓN DE BPP.....	32
3.2 OBTENCIÓN DE EXPLICACIONES DEL DESEMPEÑO ALGORÍTMICO	36
3.2.1 ANÁLISIS COMPARATIVO DE TRABAJOS RELACIONADOS	39

CAPÍTULO 4	41
4.1 METODOLOGIA DE ANÁLISIS EXPERIMENTAL PARA EL PROCESO DE OPTIMIZACIÓN ALGORÍTMICO.....	42
4.2 ETAPA 1. CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN	43
4.2.1 CARACTERIZACIÓN DEL PROBLEMA DE INVESTIGACIÓN	44
4.2.2 CARACTERIZACIÓN DEL COMPORTAMIENTO ALGORÍTMICO	52
4.2.3 CARACTERIZACIÓN DEL DESEMPEÑO FINAL	61
4.3 ETAPA 2. REFINACIÓN DE CARACTERÍSTICAS.....	64
4.3.1 ESTUDIO DE LA MATRIZ DE CORRELACIÓN	66
4.3.2 ANÁLISIS DE APORTACIÓN Y CONSISTENCIA	67
4.3.3 VALIDACIÓN DEL CONJUNTO FINAL DE MÉTRICAS	72
4.4 ETAPA 3. ESTUDIO FORMAL DE RELACIONES.....	76
CAPÍTULO 5	87
5.1 ESTRUCTURA Y DESEMPEÑO ORIGINAL DEL ALGORITMO.....	87
5.2 REDISEÑO DE LA ESTRUCTURA DEL ALGORITMO	89
5.2.1 GENERACIÓN DE INDIVIDUOS Y CRITERIO DE PARO	89
5.2.2 SUSTITUCIÓN Y CONSERVACIÓN DE INDIVIDUOS	90
5.2.3 REACOMODO DE OBJETOS LIBRES	91
5.2.4 CRUZAMIENTO	92
5.2.5 MUTACIÓN	93
5.3 RESULTADOS FINALES	94
CAPÍTULO 6	95
6.1 CONCLUSIONES	95
6.2 TRABAJO FUTURO	96
REFERENCIAS BIBLIOGRÁFICAS.....	97

Capítulo 1

INTRODUCCIÓN

En muchas organizaciones existen un gran número de problemas de optimización complejos, en los cuales se busca una asignación eficiente de recursos con la finalidad de satisfacer los objetivos planteados por la entidad. Para contribuir en ello, es necesario resolver estos problemas 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, lo cual implica que no se conocen algoritmos eficientes para resolverlos de manera exacta en el peor caso.

Para la solución de problemas NP-duros, se ha propuesto una gran variedad de algoritmos aproximados, los cuales han mostrado un desempeño satisfactorio en la solución de problemas de optimización. A pesar de los esfuerzos de una gran cantidad de

investigadores en este ámbito, 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 parece ser la única opción. Un obstáculo que se presenta al analizar el comportamiento de este tipo de algoritmos es identificar cuáles estrategias hacen que un algoritmo muestre un mejor desempeño y bajo que condiciones lo obtienen.

En este trabajo se define una metodología para el estudio integral del proceso de optimización algorítmico. La metodología propuesta combina herramientas de análisis exploratorio de datos, así como de análisis causal, con la finalidad de identificar relaciones inherentes entre los factores que afectan al desempeño algorítmico. El conocimiento obtenido mediante el análisis propuesto es aplicado en el rediseño de un algoritmo de solución del problema de empaqueo de objetos en contenedores.

1.1 ANTECEDENTES

La complejidad de los problemas reales ha hecho necesario el uso de estrategias heurísticas inteligentes. Los algoritmos diseñados a partir de estrategias heurísticas han mostrado un buen desempeño, sin embargo, no hay un algoritmo que supere a otros algoritmos para todas las instancias de un problema [Wolpert97]. Diversos trabajos han intentado explicar el comportamiento de los algoritmos ante diferentes instancias de un problema, partiendo de análisis experimentales y analizando estadísticamente los datos obtenidos.

Los algoritmos de aprendizaje automático para clasificación y predicción son muy utilizados en la minería de datos para extracción de conclusiones sobre diferentes casos de estudio, sin embargo parecen ser dependientes de los datos a partir de los cuales las conclusiones son generadas. Esto disminuye la aplicabilidad de dichas estrategias al análisis de algoritmos ya que, por si solas, no permiten generalizar las conclusiones obtenidas, ni entender el por qué de modelos generados [Pérez07, Menzies09]. Los algoritmos de descubrimiento de estructuras causales han mostrado explicaciones

confiables que permiten generalizar el comportamiento observado y beneficiarse del conocimiento obtenido [Lemeire07, Landero08].

En recientes investigaciones del CENIDET (Centro Nacional de Investigación y Desarrollo Tecnológico) e ITCM (Instituto Tecnológico de Cd. Madero) se desarrollaron tesis de maestría y doctorado que de manera conjunta formularon una metodología para la construcción de modelos causales. El presente trabajo pretende integrar lo mejor de estas dos investigaciones en una metodología más completa que pueda ser aplicada para mejorar el desempeño de algoritmos del estado del arte que dan solución al problema de empacado de objetos en contenedores.

1.2 DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN

En este trabajo se abordan dos problemas complejos, el Problema de Descubrimiento de Modelos de Explicación y el Problema de Distribución de Objetos en Contenedores. Ambos problemas han sido parte de un gran número de investigaciones debido a su importancia y utilidad en la vida real.

El *Problema de Descubrimiento de Modelos de Explicación* consiste en encontrar la estructura (modelo) que explique o pronostique las relaciones observadas en un conjunto de datos asociados a un problema. La complejidad principal de este problema radica en la indeterminación de las estructuras que pueden generarse, es decir, usualmente existe una gran cantidad de estructuras que implican el mismo conjunto de relaciones. Así mismo, las muestras de datos no siempre exhiben las mismas asociaciones e independencias incluidas en la población y los errores de medición o de selección de muestras complican la obtención de una estructura correcta [Carnegie09].

El *Problema de Distribución de Objetos en Contenedores* (BPP), intenta encontrar el menor número de contenedores en los cuales sea posible acomodar un conjunto de objetos. Éste es un problema clásico de optimización combinatoria NP-duro [Basse98],

es por ello que se ha realizado un gran esfuerzo en el desarrollo de algoritmos eficientes para su solución [Martello90, Falkenauer96, Scholl97, Kos02, Ducatelle01, Alvim04, Nieto07].

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 metaheurístico A para resolver un problema P ,
2. un conjunto de instancias o casos de prueba $I = \{i_1, i_2, \dots, i_n\}$ del problema P ,
3. un conjunto de métricas que caracterizan factores del problema y del algoritmo que impactan en el desempeño $F = \{f_1, f_2, \dots, f_m\}$.

Se busca:

1. un conjunto $R = \{r_1, r_2, \dots, r_p\}$, para $1 \leq p \leq m$, tal que cada r_i establece una relación causal entre los elementos de G_i , donde $G_i \subseteq F$,
2. aplicar el conocimiento adquirido, a través de R , en el rediseño del algoritmo A .

En otras palabras, se busca identificar relaciones entre los factores críticos para el desempeño algorítmico que permitan explicar por qué el algoritmo sigue cierto comportamiento al resolver un conjunto de casos de prueba y aplicar el conocimiento adquirido en la mejora de desempeño del algoritmo.

1.3 HIPÓTESIS

Dado un conjunto de factores que caracterizan el problema, el comportamiento de un algoritmo sobre ese problema y el desempeño obtenido por el mismo. ¿Es posible encontrar, mediante un método formal, relaciones entre ellos que expliquen por qué un

algoritmo sigue cierto comportamiento al resolver un conjunto particular de casos y aplicar dicho conocimiento en la mejora del desempeño?

1.4 JUSTIFICACIÓN

El nivel de competitividad entre las organizaciones es cada vez mayor, la rapidez y la eficiencia en el manejo de la información y la solución de problemas pueden hacer la diferencia, al disminuir costos y contribuir en la obtención de cuantiosos beneficios económicos. Para cubrir esta necesidad, y dar solución a problemas reales de alta complejidad, un gran número de algoritmos metaheurísticos han sido diseñados. Sin embargo, a la fecha ninguno ha mostrado ser superior a los demás en todos los casos. Los resultados mostrados por los algoritmos podrían ser mejorados a través del estudio de los factores que afectan su desempeño 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.

Actualmente no es fácil encontrar trabajos que proporcionen explicaciones formales del comportamiento de los algoritmos [Korb04]. La mayoría de los trabajos son descriptivos e intentan predecir o explicar de manera intuitiva e informal, basados en análisis experimentales de competencia (horse race papers), que tipo de algoritmo muestra un mejor desempeño ante ciertas instancias de un problema.

La caracterización de los factores que afectan el desempeño algorítmico además de ayudar a explicar el comportamiento de los algoritmos, permite obtener un conocimiento a detalle de cada una de las fases del proceso de solución y entender su conducta. El conocimiento obtenido de las relaciones causales entre factores de desempeño puede ser utilizado para crear guías o políticas sobre como tratar una situación (clasificación de instancias, pronósticos de desempeño), desarrollar teorías (explicación del desempeño) y mejorar el desempeño de algoritmos (rediseño de algoritmos).

1.5 OBJETIVOS

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

1.5.1 OBJETIVO GENERAL

Analizar y rediseñar estrategias de solución del problema de distribución de objetos en contenedores a través de la aplicación de una metodología basada en análisis exploratorio de características y modelado causal, con la finalidad de mejorar el desempeño del algoritmo genético híbrido HGGA-BP. Las mejoras estarán soportadas por explicaciones de los factores que afectan el comportamiento del algoritmo estudiado.

1.5.2 OBJETIVOS ESPECÍFICOS

Para lograr alcanzar el objetivo principal de este trabajo es necesario llevar a cabo una serie de etapas que, en conjunto, contribuirán en la obtención de los resultados requeridos. De esta forma se han definido los siguientes objetivos específicos:

- Disponer de un conjunto de algoritmos metaheurísticos del estado del arte, que muestren un nivel de competencia internacional y puedan ser evaluados en condiciones similares.
- Identificar estrategias heurísticas innovadoras que pudieran ser incorporadas en algoritmos metaheurísticos para incrementar su desempeño.
- Preparar conjuntos de instancias de prueba relevantes, representativas de problemas reales y/o reconocidas por la comunidad científica.
- Proponer e implementar una metodología para la generación de modelos causales de desempeño que permita:
 - Identificar y seleccionar factores relacionados con el desempeño de un algoritmo metaheurístico.

- Encontrar relaciones entre los factores que influyen en el desempeño de un algoritmo metaheurístico.
 - Explicar el desempeño algorítmico.
 - Aplicar el conocimiento obtenido en el rediseño y mejora de algoritmos.
- Desarrollar un modelo causal que explique el desempeño de un algoritmo metaheurístico en la solución del problema clásico de empaçado de objetos en contenedores (Bin Packing) y aplicarlo para mejorar su desempeño.

1.6 ALCANCES Y LIMITACIONES

1. La caracterización del proceso de optimización se aplicará únicamente al problema de distribución de objetos en contenedores (BPP).
2. El algoritmo de solución de BPP que se estudiará y rediseñará, para mostrar la validez de la metodología propuesta, es el genético híbrido HGGA-BP [Nieto07].
3. Para algunas de las instancias de prueba que se utilizarán no se conoce la solución óptima. Para estas instancias, el análisis comparativo se basará en un límite inferior del valor óptimo.
4. 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 esta organizado de la siguiente manera: el Capítulo 2 presenta una revisión de los fundamentos teóricos necesarios para el análisis experimental de algoritmos metaheurísticos. Se describe el problema de empaçado de objetos en

contenedores y las estrategias metaheurísticas aplicadas para resolver este problema. Se presentan también los fundamentos teóricos del análisis del comportamiento de algoritmos y de la caracterización del proceso de optimización. Finalmente se presenta una visión general de la obtención de explicaciones mediante estrategias de análisis exploratorio de datos y modelado causal, describiendo para esto los conceptos teóricos necesarios para su aplicación.

El capítulo 3 inicia presentando los algoritmos más destacados para la solución del problema de empaqueo de objetos en contenedores, destacando las estrategias que serán objeto de estudio. Posteriormente, se muestra una reseña de los trabajos relacionados con la caracterización del desempeño de algoritmos, así como de los enfoques propuestos para dicho estudio. Finalmente se muestra un análisis comparativo de los trabajos que intentan obtener explicaciones de desempeño de algoritmos metaheurísticos y de las aportaciones que se incluirán en la metodología propuesta en esta tesis.

En el Capítulo 4 se presenta de manera detallada la metodología propuesta y su aplicación en el estudio del algoritmo genético híbrido HGGA-BP. Las principales etapas de la metodología son: caracterización del problema (BPP), caracterización del comportamiento y desempeño algorítmico; estudio formal de relaciones entre el problema, el comportamiento algorítmico y el desempeño final alcanzado; y obtención de explicaciones del comportamiento observado.

En el Capítulo 5 se presenta el rediseño del algoritmo HGGA-BP, y se incluyen los resultados experimentales de la aplicación de la metodología propuesta para el análisis de desempeño del metaheurístico híbrido. Obteniendo como producto final un algoritmo rediseñado que muestra una contundente mejora en el desempeño.

En el Capítulo 6 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 el problema de empaqueo de objetos en contenedores y su solución, y se describen las principales técnicas empleadas en análisis del desempeño de algoritmos metaheurísticos que resuelven problemas complejos.

La mayoría de los trabajos relacionados con el análisis del desempeño de algoritmos metaheurísticos, están enfocados en análisis comparativos de resultados experimentales. No obstante, trabajos recientes en el área han señalado la necesidad de explicar el por qué de los resultados observados. 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. Por lo antes expuesto, las secciones finales del capítulo tratan el análisis de desempeño orientado a la generación de conocimiento.

2.1 PROBLEMA DE EMPACADO DE OBJETOS EN CONTENEDORES

El problema de empacado 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 [Basse98, Álvarez06].

Dados un conjunto $N = \{1, \dots, n\}$ de objetos a distribuir en contenedores del mismo tamaño, 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 m para una partición del conjunto N

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

tal que:

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

2.2 ALGORITMOS APROXIMADOS PARA LA SOLUCIÓN DE BPP

Los problemas denominados *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 (ver Figura 2.1). En estas condiciones se hace uso de algoritmos que dan soluciones aproximadas en un tiempo razonable, pero no garantizan obtener la mejor solución.

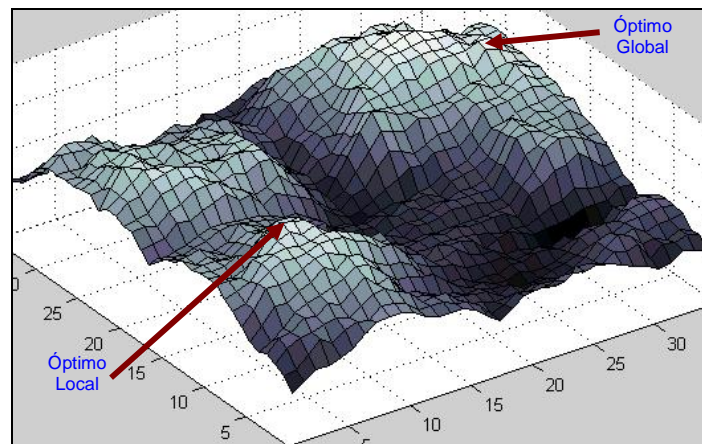


Figura 2.1 Espacio de soluciones de un problema de maximización.

Los *algoritmos aproximados* 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]. Existen procedimientos que combinan estrategias heurísticas de diferente naturaleza, dichas técnicas son conocidas como *algoritmos híbridos*.

La complejidad del problema de distribución de objetos en contenedores 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, debido a que el número de posibles particiones es mayor que $(n/2)^{n/2}$ [Basse98]. Por lo anterior los algoritmos utilizados en la solución de este problema son algoritmos aproximados que incluyen estrategias heurísticas.

2.2.1 ALGORITMOS DETERMINISTAS

Se dice que los algoritmos aproximados son *deterministas* cuando sólo pueden encontrar una única solución para cada instancia de un problema (no importa cuantas veces se ejecute el algoritmo, la solución encontrada será siempre la misma). Diferentes algoritmos de este tipo han mostrado resultados satisfactorios para la solución de BPP. Estos procedimientos incluyen heurísticas sencillas que se diferencian por la manera en que los objetos son tratados antes ser acomodarlos y por la forma en que se elige el contenedor que almacenará cada objeto. Las estrategias más destacadas en esta categoría son descritas a continuación:

- *Primer Ajuste* (First Fit, FF): Cada objeto considerado es colocado en el primer contenedor que tenga suficiente capacidad disponible. Si ningún contenedor parcialmente lleno puede almacenarlo, el objeto se coloca 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): 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): Contrario a la estrategia anterior (BF), el objeto en consideración es almacenado en el contenedor menos lleno con capacidad residual suficiente. La variación que toma los objetos en orden decreciente se denomina *Peor Ajuste Decreciente* (Worst Fit Decreasing, WFD).

- *Best 3-Fit Decreasing* (B3FD) [Alvim04]: 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 exactamente 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, para las que se ha demostrado que las soluciones obtenidas por estos algoritmos están alejadas de las soluciones óptimas [Martello90b, Schwerin97, Wäscher96]. Debido a lo anterior se han diseñado procedimientos más elaborados, que integran técnicas variadas, en la búsqueda de buenas soluciones para BPP. Resultados destacados han sido obtenidos mediante el uso de algoritmos metaheurísticos.

2.2.2 ALGORITMOS METAHEURÍSTICOS

Un *algoritmo metaheurístico* es un algoritmo aproximado de propósito general, que combina diferentes heurísticas definidas de acuerdo al problema. Estos algoritmos se basan en búsquedas por entornos y utilizan estrategias aleatorias inteligentes para evitar caer en óptimos locales (ver Figura 2.1) y mejorar las soluciones obtenidas. Las principales características de este tipo de algoritmos son [Michalewicz04, Dréo06, Gonzalez07]:

- Las estrategias aleatorias que incluyen los hacen comportarse de manera no determinista; es decir no encuentran la misma solución al ejecutarse varias veces con la misma entrada.
- Parten de soluciones iniciales (generadas mediante métodos aleatorios, funciones constructivas voraces o estrategias heurísticas deterministas) y mediante ciertas

alteraciones (definidas según un operador de búsqueda), exploran soluciones vecinas de su entorno.

- Para la terminación del proceso se sigue algún criterio de parada, tal como la convergencia del algoritmo a una solución aceptable o un número de iteraciones establecidas. El resultado final es la mejor solución visitada.
- Utilizan distintas estrategias para forzar la exploración de la mayoría de las regiones del espacio de búsqueda (diversificación) e intensificar la búsqueda en regiones prometedoras (intensificación).
- Algunas metaheurísticas memorizan acciones realizadas en el pasado para evitar regresar a soluciones visitadas con anterioridad.
- El uso de reglas de transición aleatorias al explorar soluciones les permite ampliar las posibilidades de elección para evitar caer en óptimos locales, principalmente cuando existen factores no cuantificables que no pudieron ser añadidos en el modelo, pero que también deben ser considerados.

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], ante esta situación surge un nuevo problema: la selección de algoritmos. Los trabajos relacionados con la temática han tratado de analizar el comportamiento de los algoritmos metaheurísticos con la finalidad de identificar algoritmos de alto desempeño.

Entre las metaheurísticas que han mostrado mejores resultados al ser utilizadas como base de algoritmos de solución de BPP se encuentran: la búsqueda tabú, los algoritmos genéticos y la optimización basada en colonia de hormigas.

La *Búsqueda Tabú* (Tabú Search, TS), desarrollada por Glover [Glover86], es un procedimiento de búsqueda local que memoriza la historia de la búsqueda (en una lista tabú) y utiliza dicha información al implementar estrategias de exploración del espacio de soluciones, evitando retrocesos a soluciones ya visitadas. En este tipo de algoritmos, el proceso comienza con una solución inicial y continúa buscando iterativamente un óptimo global, transitando entre soluciones vecinas que no sean tabú (prohibidas). Cada solución explorada es agregada temporalmente a la lista tabú, el tiempo de permanencia de las prohibiciones en dicha lista es actualizado en cada iteración, liberando las soluciones que han cumplido su tiempo [Scholl97, Alvim04].

Los *Algoritmos genéticos* (Genetic Algorithms, GA), presentados por Holland [Holland75], son algoritmos evolutivos (poblacionales) basados en el principio de evolución natural de los seres vivos. Establecen una analogía entre el conjunto de soluciones de un problema y el conjunto de cromosomas de una especie. La calidad de los cromosomas (soluciones) es evaluada mediante la función objetivo del problema. A través de cierto número de generaciones (iteraciones), la población es sometida a un proceso de evolución que involucra: selección (dando preferencia a los cromosomas más aptos), cruzamiento y mutación, dando como resultado soluciones de mayor calidad [Falkenauer96, Kos02, Nieto07].

La *Optimización Basada en Colonias de Hormigas* (Ant Colony Optimization, ACO), introducida por Dorigo [Dorigo92], se inspira en el comportamiento observado en hormigas reales para encontrar caminos cortos entre fuentes de comida y el hormiguero siguiendo rastros de feromona dejados en el camino por otras hormigas. Se basa en un conjunto de agentes computacionales (hormigas artificiales) que construyen, de manera probabilística, soluciones al problema. En cada iteración una hormiga añade un componente a la solución tomando en cuenta la información heurística del problema y los rastros de feromona disponibles. La reserva de feromonas evapora un poco después de cada iteración y se refuerza con buenas soluciones, reflejando la experiencia de otras hormigas [Ducatelle01].

2.3 ANÁLISIS EXPERIMENTAL DE ALGORITMOS METAHEURÍSTICOS

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 o 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 aproximados dependen de los métodos elegidos para su caracterización, que pueden ser teóricos ó 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.

El estudio teórico de la efectividad en los algoritmos metaheurísticos es muy poco común, pues la aleatoriedad en dichos algoritmos y la complejidad de los problemas de optimización combinatoria que solucionan dificultan un correcto análisis matemático. Además, la aplicabilidad de los resultados teóricos es muy limitada, pues éstos son obtenidos partiendo de condiciones idealizadas que no ocurren en situaciones prácticas [McGeoch02, Hoos07, Moret02]. 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 metaheurísticos de manera experimental.

A pesar de la importancia del análisis de desempeño experimental, éste no ha sido suficientemente explotado en el estudio de algoritmos metaheurísticos. Una de las razones que dificultan este estudio es que, generalmente, los algoritmos son considerados como cajas negras cuyo funcionamiento interno es desconocido. 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 que lo definen, estructura del espacio de solución, entre otros.
- b. *Factores del algoritmo*. Incluyen las estrategias heurísticas seleccionadas (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*. Estos factores se refieren al ambiente físico en el que serán ejecutados los algoritmos como los son el software (sistema operativo, compilador) y hardware (velocidad del procesador, 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.

Contribuyendo con lo anterior, trabajos recientes proponen herramientas que pueden ser tomadas en cuenta para facilitar el análisis de los factores que influyen en el desempeño algorítmico y con ello la explicación del desempeño [Hooker94, Barr95, Cohen95, McGeoch02, Lemeire07]. En general, las *herramientas para análisis de datos* (factores)

propuestas en esos trabajos 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 del conjunto de datos y de su estructura subyacente. Incluye métodos estadísticos, gráficos y de análisis multivariado que ayudan a obtener un modelo que describe el conjunto de relaciones de los factores bajo estudio [Tukey77, Hartwig79, Liu96].
- b. *Análisis de datos confirmatorio* (Statistical Hypothesis Testing). Comienza con suposiciones (modelos) acerca de relaciones funcionales entre variables (factores) del conjunto de datos. Incluye estimaciones de parámetros del modelo y pruebas de hipótesis estadísticas para complementar y validar el modelo propuesto [Hartwig79, Cohen95, Liu96].

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 y algoritmo). Por ejemplo, los métodos gráficos de análisis de datos son los más apropiados para analizar tendencias en grandes conjuntos de datos multivariados. Los métodos de análisis de datos confirmatorio pueden ser problemáticos, dependiendo de cuanto conocimiento se tiene de la función (modelo). 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].

Un correcto análisis de desempeño 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].

2.4 CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN

La caracterización del problema y del proceso de solución es una parte esencial en el análisis de desempeño de algoritmos, y permite identificar cuáles son los factores que influyen en el comportamiento algorítmico. Un análisis de desempeño de calidad requiere la definición de métricas adecuadas que cuantifiquen las características indicadoras del desempeño.

La metodología propuesta en este trabajo (ver Sección 4.1) consta de tres etapas principales, las cuales permiten explicar el proceso de optimización: formulación de funciones de caracterización, refinación de funciones y obtención de explicaciones del desempeño algorítmico.

2.4.1 PROCESO DE OPTIMIZACIÓN

Para lograr comprender el desempeño de un algoritmo sobre un problema, debe realizarse un estudio integral de todo el proceso de optimización. 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 2.2.

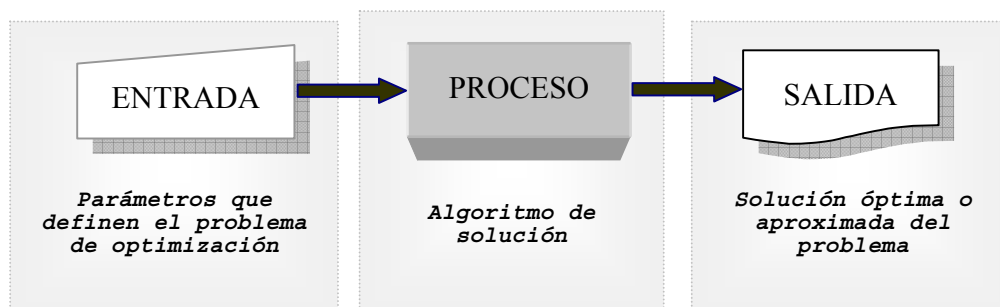


Figura 2.2 Proceso de optimización de un problema.

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.

2.4.2 FORMULACIÓN DE FUNCIONES DE CARACTERIZACIÓN

El objetivo principal de esta etapa es identificar aspectos, en cada una de las fases del proceso de optimización, que sean factibles de medición y que tentativamente proporcionen información útil para describir el desempeño de un algoritmo. En otras palabras, este análisis tiene que ver tanto con el problema que se pretende resolver como con la implementación del algoritmo que se utilizará para solucionarlo y los resultados mostrados por éste.

Los algoritmos metaheurísticos, al ser multipropósito, tienen muchas variantes que se ajustan al problema que resuelven. Por esta razón, la medición de funciones de caracterización esta ligada directamente a la estrategia utilizada. Además, el comportamiento no determinista de éste tipo de algoritmos hace más interesante el análisis debido a que su conducta puede ser diferente con cada posible entrada de un mismo problema. En la Figura 2.3 se muestran algunos elementos, propuestos en la literatura [Barr95, McGeoch02, Cruz04, Álvarez06, Pérez07,], para identificar factores críticos en el proceso de optimización.

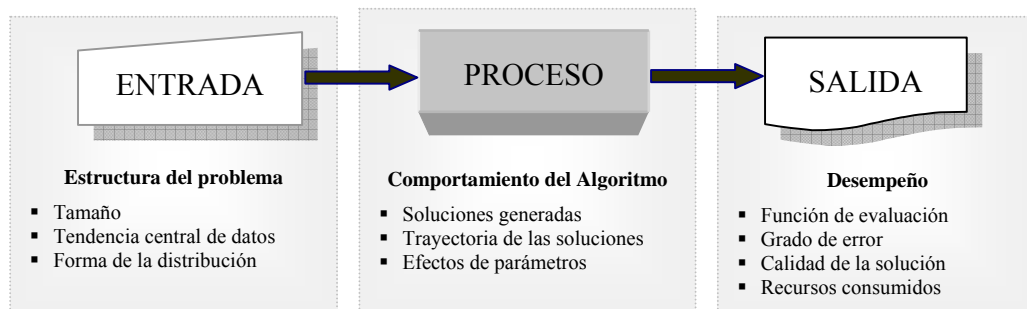


Figura 2.3 Factores que influyen en el proceso de optimización.

En la entrada, los casos proporcionan información acerca de la estructura de las instancias del problema; en el proceso los elementos principales son aquellos que proporcionan ideas acerca del comportamiento del algoritmo en ejecución; y en la salida, se buscan indicadores importantes del desempeño. Después de identificar los elementos que intervienen en cada una de las fases del proceso de optimización, es necesario analizar qué aspectos es posible medir en cada categoría. Para esto es necesario estudiar tanto el problema como la implementación utilizada para resolverlo y seleccionar variables que revelen tendencias importantes.

Tabla 2.1 Indicadores propuestos en el trabajo de Cruz [Cruz04].

Expresión	Descripción	Identificador de expresión
$p = \frac{n}{nmax}$	p es el índice del tamaño del caso, donde n = número de objetos, $nmax$ = tamaño máximo solucionado .	(2.1)
$t = \frac{\sum_{i=1}^n w_i / n}{c}$	t es el índice de capacidad ocupada por un objeto promedio, donde w_i = peso del objeto i , c = capacidad del contenedor.	(2.2)
$d = \sqrt{\frac{\sum_{i=1}^n \left[t - \left(\frac{w_i}{c} \right) \right]^2}{n}}$	El índice de dispersión d expresa el grado de la dispersión del cociente del peso de los objetos entre el tamaño del contenedor.	(2.3)
$f = \frac{\sum_{i=1}^n factor(c, w_i)}{n}$	El índice de factores f expresa la proporción de objetos cuyo peso w_i es factor de la capacidad del contenedor, donde $factor(c, w_i) = \begin{cases} 1 & \text{si } (c \bmod w_i) = 0 \\ 0 & \text{en caso contrario} \end{cases}$	(2.4)
$b = \begin{cases} 1 & \text{si } c \geq \sum_{i=1}^n w_i \\ \frac{c}{\sum_{i=1}^n w_i} & \text{en caso contrario} \end{cases}$	El uso de contenedor b expresa la proporción del peso total que se puede asignar en un contenedor de capacidad c .	(2.5)

En particular, para el problema de empaqueo de objetos en contenedores (BPP), se han propuesto funciones de caracterización que describen algunas de las fases del proceso de

optimización. El mayor avance se observa en la caracterización de la entrada, es decir de los casos de prueba o instancias de BPP, donde se pueden destacar los trabajos de Cruz [Cruz04] y Álvarez [Álvarez06]. En la Tabla 2.1 (pagina anterior) se describen las funciones desarrolladas en el trabajo de Cruz [Cruz04] para determinar la estructura de un caso de entrada en el proceso de optimización de BPP.

En el trabajo de Álvarez [Álvarez06] se proponen funciones de caracterización de propósito general, basadas en estadística descriptiva, que pueden ser aplicadas a una amplia gama de problemas, además de BPP. La Tabla 2.2 muestra las funciones propuestas por el autor para ser aplicadas en la caracterización de BPP. Los indicadores mencionados pueden ser agrupados en cuatro categorías de medidas estadísticas que, aplicadas a BPP, describen: *centralización*, indican valores con respecto a los que los pesos de los objetos parecen agruparse; *dispersión*, muestran el grado de concentración de los pesos con respecto a las medias centralizadas; *posición*, dividen un conjunto de pesos en grupos con la misma cantidad de individuos; *forma*, buscan cuantificar algunos aspectos formales de una distribución de frecuencias de los pesos.

Tabla 2.2 Indicadores propuestos en el trabajo de Álvarez [Álvarez06].

Medidas	Estadístico
Centralización	Media aritmética
	Media geométrica
	Media armónica
	Mediana
	Moda
Dispersión	Rango
	Desviación media
	Varianza
	Desviación estándar
	Cuasivarianza
	Cuasidesviación estándar
	Coefficiente de variación
Error estándar	
Posición	Cuartiles
	Deciles
	Percentiles
Forma	Coefficiente de asimetría de Pearson
	Coefficiente de asimetría de Pearson mediana
	Coefficiente de asimetría de Pearson moda
	Coefficiente de asimetría de Bowley
	Curtosis

Los indicadores anteriores (propuestos por Cruz y Álvarez) son retomados en los trabajos de Pérez y Landero [Pérez07, Landero08]. Los autores proponen, además, medir aspectos en cada etapa del proceso de optimización para analizar el desempeño de estrategias metaheurísticas aplicadas en la solución de BPP. Cada autor desarrolla métricas de desempeño para caracterizar el comportamiento de diferentes variantes del algoritmo de interés. La Figura 2.4 muestra los factores que se recomienda medir para cada fase.

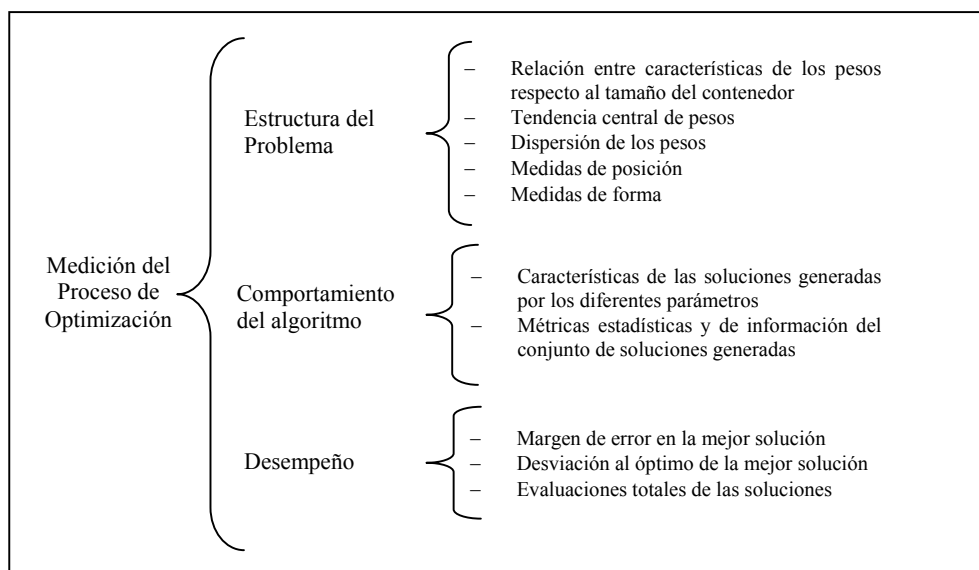


Figura 2.4 Caracterización propuesta por Pérez y Landero [Pérez07, Landero08].

La caracterización de proceso de optimización es un área emergente y prometedora que permite vislumbrar el funcionamiento de los algoritmos metaheurísticos ante cada tipo de entrada específica de un problema.

2.4.3 REFINACIÓN DE FUNCIONES DE CARACTERIZACIÓN

El objetivo de esta fase es conservar el número mínimo de métricas que permitan caracterizar tanto el problema como el desempeño del algoritmo sin sacrificar la calidad del conocimiento que se pueda obtener al evaluar dichas medidas. De esta forma, debe

realizarse una selección de funciones eliminando aquellas que no sean relevantes, así como las redundantes y las difícilmente calculables.

El análisis de datos exploratorio incluye herramientas que permiten obtener datos de calidad, como son: identificación y tratamiento de datos *outliers* (no coherentes) y datos erróneos, modificación de datos (transformación de variables, estandarización de datos), resumen de datos (estadísticas descriptivas, tablas, gráficas), técnicas de análisis multivariado (agrupamiento, componentes principales), métodos de inferencia (regresión múltiple, análisis causal), entre otros. Dichas herramientas permiten analizar cada una de las funciones de caracterización mostrando su utilidad y validez. Con lo anterior puede obtenerse un conjunto de métricas de calidad para la caracterización del proceso de optimización de BPP.

Otro aspecto importante a verificar en las funciones de caracterización es la redundancia, pues podría ocurrir que algunas funciones de caracterización midan un mismo factor. El uso de funciones con un alto grado de asociación lineal entre ellas, puede causar errores en estudios estadísticos futuros. En la literatura pueden observarse dos enfoques referentes a este tipo de refinación de características. El primero selecciona funciones de caracterización representativas, eliminando aquellas no relevantes. El segundo enfoque tiene como objetivo la obtención, de un número menor, de nuevas funciones que resuman las iniciales:

- *Selección de funciones.* Son métodos utilizados en el área de aprendizaje automático, consisten en la eliminación de funciones de caracterización con baja o ninguna relevancia, de tal forma que se reduzca el tiempo de aprendizaje del conjunto de datos, mientras se mantiene un valor de precisión aceptable del conocimiento que se puede obtener de ellos. Algunos métodos de selección son: wrappers, filtros, selección hacia adelante, eliminación hacia atrás y selección por pasos [Guyon03, Yu04, Witten05].

- *Creación de nuevas funciones.* Existen diferentes métodos multivariados que buscan relaciones entre características y crean, a partir de éstas, nuevas funciones de caracterización independientes que resumen toda la información que podría extraerse de las originales. El resumir la información en un menor número de variables facilita diversos tipos de análisis, por ejemplo análisis gráficos. Algunos métodos multivariados que crean nuevas funciones son el análisis de componentes principales, el análisis por factores, el análisis de correlación canónica, el análisis discriminante canónico y el análisis de variables canónicas [Johnson00].

2.4.4 EXPLICACIÓN DEL DESEMPEÑO ALGORÍTMICO

La caracterización de algoritmos y problemas busca establecer una relación entre las funciones de caracterización que los describen y el desempeño que tendrá el algoritmo al resolver un problema. El objetivo es entender el comportamiento de un algoritmo metaheurístico al aplicarse en la solución de un problema particular. En lo sucesivo, a estas relaciones que permiten explicar el desempeño algorítmico se les llamará: *relaciones de desempeño*.

La finalidad de esta etapa es extraer las relaciones de desempeño que existen en el proceso de optimización. Para lograr dicho objetivo, una vez definidas y refinadas las funciones de caracterización, es necesario analizar los valores mostrados por dichas medidas en diferentes procesos de optimización (comportamiento del algoritmo ante diversas instancias del problema utilizando diferentes configuraciones). Como se mencionó anteriormente (Sección 2.3), la obtención de conocimiento de grandes conjuntos de datos debe conjuntar técnicas exploratorias para visualización, descripción y modelado de relaciones de desempeño, así como procedimientos confirmatorios para probar los modelos, hipótesis y predicciones generados mediante la fase exploratoria.

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 cuál 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, de sin dar detalles de la forma en la que dichos resultados fueron obtenidos. La tendencia parece ser 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].

En este trabajo de investigación se propone contribuir a la sistematización del proceso de descubrimiento creativo de explicaciones de desempeño, proponiendo una metodología (presentada en el Capítulo 4), que combina herramientas exploratorias, para descubrir relaciones entre los factores que influyen en el proceso de optimización algorítmico.

2.5 RELACIONES DE DESEMPEÑO ALGORÍTMICO

Las relaciones entre conjuntos de variables tienen tres características importantes [Hartwig79]: intensidad, dirección y forma. Dichas características deben ser tomadas en cuenta al momento de elegir los métodos que serán utilizados para la identificación de relaciones de desempeño.

La *intensidad* se refiere al grado en el que los datos de una variable corresponden con los datos de otras, a menudo esto es determinado por el grado en el que los valores observados de algunas variables pueden predecirse partir de los valores correspondientes de otras variables. La *dirección* tiene que ver con lo que ocurre con una variable cuando los valores de otras son grandes o pequeños, las relaciones pueden ser positivas, negativas o tener más de una dirección y esto tiene que ver con la forma de la relación. La *forma* de la relación se refiere a la manera en la cual las variables se relacionan unas con otras. Cuando se habla de relaciones entre pares de variables, la forma puede ser una línea recta, una curva o una curva compuesta; y cuando son relaciones múltiples pueden encontrarse formas multidimensionales, por ejemplo hiperplanos.

La búsqueda de estructuras (modelos de explicación) que capturen las relaciones de desempeño, explicando el comportamiento algorítmico, es un proceso de descubrimiento creativo, en el que medidas de asociación, comparaciones tabulares, técnicas de visualización de datos, así como herramientas de modelado, deben complementarse de manera inteligente e intuitiva, siguiendo las pistas que van obteniéndose en la exploración de datos y el conocimiento y dominio que se tiene del problema objeto de estudio. En esta sección se describen herramientas que son de utilidad para la identificación de relaciones de desempeño y para la definición de modelos que expliquen, en conjunto, la estructura subyacente del proceso de optimización.

Un primer acercamiento, en la búsqueda de relaciones de desempeño, es el cálculo de la matriz de correlación, la cual permite tener cierto conocimiento sobre la intensidad de las relaciones existentes entre cada par de variables. El *Coefficiente de correlación de Pearson* es una medida de asociación lineal entre dos variables. Esta medida varía entre -1 y 1, donde un valor cercano a cero indica que no existe una relación lineal entre las dos variables (aunque esto no indica necesariamente que no existe otro tipo de relación). La matriz de correlación incluirá las correlaciones existentes entre todos los factores que afectan el desempeño algorítmico. De aquí pueden obtenerse indicios de cuales factores pueden tener un mayor impacto en el desempeño algorítmico.

Otra herramienta útil tanto para descubrir relaciones de desempeño como para observar las características de dichas relaciones son las *gráficas de dispersión*. Este tipo de gráficas permiten obtener una representación visual de la relación existente entre dos variables al situarlas, una contra otra, en una gráfica. Por lo común, revelan si, en realidad, dos variables están relacionadas entre sí, además de cuándo y cómo podrían estar relacionadas, es decir, ayudan a visualizar la intensidad, la forma y la dirección de la relación existente entre pares de variables.

También pueden utilizarse gráficas de dispersión que ayudan a visualizar las relaciones existentes entre más de dos variables, por ejemplo gráficas de burbujas, gráficas tridimensionales y la combinación de ambas, para graficar cuatro variables. Algunos paquetes estadísticos permiten crear gráficas de dispersión en tres o cuatro dimensiones y, a continuación, girar la nube de puntos (datos) alrededor de los vértices de los ejes. Estas gráficas interactivas revelan aspectos importantes del conjunto de datos y facilitan el estudio de relaciones de desempeño desde diferentes perspectivas.

Las presentaciones visuales de los datos son más informativas que su impresión en grandes conjuntos o que ciertas medidas estadísticas. Las gráficas de distribución que combinan variables, los histogramas y otras gráficas multidimensionales, además de posibilitar la identificación de relaciones de desempeño, facilitan la identificación de conjuntos de datos con características similares.

Existen métodos multivariados que ayudan reducir la dimensionalidad de conjuntos de datos, creando nuevas variables con propiedades deseables para futuros análisis. En muchas ocasiones la transformación de los datos, favorece los análisis gráficos y permite identificar conjuntos de variables que se encuentran fuertemente relacionadas entre sí. Un método que puede destacarse es el *análisis de componentes principales*, que es un procedimiento matemático que transforma un conjunto de variables respuesta correlacionadas en un nuevo conjunto de variables no correlacionadas conocidas como componentes principales (Principal Component Analysis, PCA).

El PCA además de ser una excelente herramienta para cribar datos de variables múltiples y facilitar análisis gráficos, suele ser bastante útil para complementar los análisis de agrupación. El objetivo de un *análisis de agrupación* (Cluster Analysis, CA) es realizar la división en subgrupos de las unidades experimentales, de modo que unidades experimentales similares pertenezcan al mismo subgrupo. Las gráficas de los componentes principales ayudan a validar los subgrupos encontrados. En algunos casos, el utilizar las calificaciones de los componentes principales como entrada de programas de agrupación incrementa la eficacia de éstos [Johnson00]. Este tipo de análisis ayuda a identificar instancias del problema que tienen características comunes, para estudiar, por ejemplo, las relaciones del comportamiento algorítmico dentro de cada uno de los grupos.

Los métodos presentados hasta ahora, permiten descubrir relaciones de desempeño entre conjuntos de variables. Sin embargo, pudiera ocurrir que en éste punto del estudio, las relaciones sean sólo piezas de un rompecabezas. Se tiene conocimiento de ellas y cómo deberían conectarse, incluso se puede diferenciar entre las piezas más importantes y las complementarias, y armar algunas. Pero no es tan fácil tener una explicación de cómo todas las piezas deben conectarse y de lo que se obtendrá al completar el rompecabezas. Una manera de explicar como trabajan todas las piezas del rompecabezas es la construcción de modelos causales.

Un *modelo causal*, o estructura causal, es una representación generalizada del conocimiento, obtenido al encontrar dependencias que impliquen relaciones de causa y efecto entre variables [Lemeire04]. El uso de *herramientas de inferencia causal* permite analizar las funciones características del proceso de optimización para determinar la estructura causal (modelo) que explique, en su conjunto, las relaciones de desempeño existentes en el proceso de optimización.

El término *causalidad* indica la manera en que el mundo responde a una intervención, es decir describe la relación entre causas y eventos. Para que un suceso *A* sea la causa de

un suceso B se tienen que cumplir tres condiciones: 1) que A suceda antes que B , 2) que siempre que suceda A suceda B , y 3) que A y B estén próximos en el espacio y en el tiempo. Al observar en repetidas ocasiones el cumplimiento de estas condiciones se llega a generalizar que puesto que hasta ahora siempre que ocurrió A se ha dado B , en el futuro ocurrirá lo mismo. Así se establece una ley. Las relaciones causales son transitivas, irreflexivas y antisimétricas. Esto quiere decir que: 1) si A es causa de B y B es causa de C , entonces A es además causa de C , 2) un evento A no puede causarse a sí mismo, y 3) si A es causa de B entonces B no es causa de A .

Existen diferentes herramientas computacionales disponibles para la creación de modelos causales, como Hugin Tool [Madsen05], Netica [Norsys09] y TETRAD [Carnegie09]. Además de recursos desarrollados para la enseñanza de modelos causales como el laboratorio Causality Lab de la Universidad de Carnegie-Mellon [Carnegie09]. La herramienta seleccionada para realizar el aprendizaje de estructuras causales en este trabajo es TETRAD.

TETRAD [Carnegie09] es un software gratuito desarrollado con el objetivo de crear, simular datos, estimar, probar, predecir y buscar modelos estadísticamente causales. Ofrece métodos para descubrimiento causal en una interfaz amigable y sencilla de usar. Dicha herramienta cuenta con varias implementaciones de algoritmos de inferencia causal para generar grafos que representan modelos causales. Los algoritmos incluidos se basan, principalmente, en pruebas de independencia condicional realizadas a partir de la matriz de varianzas-covarianzas y pueden ser aplicados tanto a datos continuos como discretos.

2.6 CONOCIMIENTO OBTENIDO EN EL ANÁLISIS DE DESEMPEÑO

La caracterización del proceso de optimización además de ayudar a explicar el desempeño de algoritmos, permite obtener un conocimiento a detalle de cada una de las fases de este proceso. El conocimiento de la estructura de dicho proceso, los valores que

la definen y las relaciones entre estos valores, permiten, a su vez, entender la conducta del algoritmo de solución. 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 difíciles y fáciles para todos los 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 porque de su buen o mal desempeño.
- *Pronósticos de desempeño.* Dado que no se conoce un algoritmo metaheurístico 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 permitiría ahorrarse tiempo en la solución de problemas, obteniendo buenos resultados.
- *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.

Capítulo 3

ESTADO DEL ARTE

En este capítulo se presenta una breve revisión de los algoritmos metaheurísticos para solución de BPP que han mostrados resultados sobresalientes. Posteriormente, se ofrece una síntesis de los principales trabajos relacionados con la caracterización y el estudio del desempeño de algoritmos. Para concluir, se muestra un análisis comparativo de los trabajos que intentan obtener explicaciones de desempeño en algoritmos metaheurísticos y de las aportaciones que se incluirán en la metodología propuesta en esta tesis.

3.1 ALGORITMOS METAHEURÍSTICOS PARA LA SOLUCIÓN DE BPP

Dada la complejidad del problema de distribución de objetos en contenedores, un gran número de investigadores (tanto matemáticos, como expertos en computación) se han dado a la tarea de dar soluciones prácticas a implementaciones reales relacionadas con este problema. En particular, el uso de algoritmos metaheurísticos híbridos ha mostrado los resultados sobresalientes. A continuación se mencionan algunos de los trabajos que hicieron contribuciones importantes en este ámbito.

Martello y Toth [Martello90b] introducen el algoritmo Branch and Bound *MTP*, método exacto que se vuelve aproximado para evitar tiempos de procesamiento inaccesibles. El procedimiento se basa en técnicas de reducción del problema que utilizan el *criterio de dominación* propuesto por los autores [Martello90a]. Dicho criterio establece que: si es posible sacar un conjunto de objetos de un contenedor e intercambiarlos por un solo objeto del mismo peso que el conjunto, la solución puede ser mejorada, debido a que será más fácil distribuir el conjunto de objetos pequeños en el resto de los contenedores, que encontrar un lugar para el objeto grande. Este método utiliza el algoritmo determinista FFD para crear soluciones y, mediante el establecimiento de límites inferiores del número de contenedores que debería tener una solución óptima, busca reducir el tamaño de las soluciones.

Falkenauer [Falkenauer96] propone el híbrido *HGGA*. El algoritmo utiliza un esquema de representación por grupos (contenedores). La población es generada por la estrategia heurística determinista FF, a partir de permutaciones aleatorias de objetos. Mediante la aplicación de operadores de cruce y muta, que incluyen un método de optimización local inspirado en criterios de dominación [Martello90a], se evoluciona creando mejores soluciones. Los experimentos son realizados con 160 instancias duras, propuestas por el autor (disponibles en: <http://people.brunel.ac.uk/~mastjbj/jeb/orlib/binpackinfo.html>). Falkenauer compara sus resultados (*HGGA*) con los obtenidos por el procedimiento *MTP* [Martello90b] y obtiene mejores soluciones y en menor tiempo.

Scholl et al. proponen el procedimiento híbrido exacto *BISON* [Scholl97], donde se combinan límites inferiores de contenedores, procedimientos de reducción, heurísticos deterministas para BPP, la metaheurística de búsqueda tabú y un procedimiento Branch and Bound. Los autores utilizan una estrategia llamada dual, mediante la cual intentan solucionar BPP iniciando con un límite (inferior) de contenedores, el cuál va siendo incrementado hasta encontrar una solución factible. Los autores hacen una contribución a la comunidad científica al generar tres conjuntos de instancias de prueba (disponibles en <http://www.wiwi.uni-jena.de/Entscheidung/binpp/>).

Ducatelle [Ducatelle01] desarrolla un algoritmo híbrido *HACO-BP*. El procedimiento implementa la metaheurística optimización basada en colonia de hormigas (ACO) incluyendo una estrategia de búsqueda local, apoyada en el criterio de dominación. La solución de cada hormiga se mejora moviendo algunos objetos alrededor, y las soluciones mejoradas se usan para actualizar la matriz de feromonas. Para problemas grandes, ACO con búsqueda local no siempre encuentra la solución señalada por Falkenauer, sin embargo, siempre supera al algoritmo puro de ACO.

Kos y Duhovnik describen un algoritmo genético híbrido de agrupación para el problema de empaqueo de objetos en contenedores de tamaños variables (variable-sized bin packing problem, VBPP) *GA-VBPP* [Kos02]. Los objetos liberados al aplicar operadores de cruce y muta son empaquetados mediante una técnica de optimización local que intenta aumentar y mejorar el llenado de los contenedores. Dicho algoritmo obtiene soluciones óptimas para instancias de las propuestas por Falkenauer de tamaño 120 [Falkenauer96] (mejorando dos soluciones), sin embargo al aumentar el tamaño de las instancias el algoritmo se vuelve notablemente más lento.

Alvim [Alvim04] desarrolla el procedimiento híbrido *HI-BP*, en el que retoma la estrategia dual utilizada por Scholl et al. [Scholl97], así como las técnicas de reducción de Martello y Toth [Martello90a]. El algoritmo consta de tres fases: construcción, redistribución y mejora. Durante la fase de construcción se generan soluciones inviables (duals) mediante modificaciones de cuatro algoritmos deterministas para BPP. En la redistribución se intenta eliminar la inviabilidad resolviendo problemas contenidos en BPP mediante el equilibrio y desequilibrio de pesos de contenedores. Las soluciones obtenidas son mejoradas mediante Búsqueda Tabú. La estrategia logra resolver de manera óptima todas las instancias de prueba utilizadas por los algoritmos presentados en los párrafos anteriores, además de considerar instancias estándar que no fueron consideradas en dichos trabajos.

Otro algoritmo genético híbrido de agrupación *HGGA-BP* es propuesto en el trabajo de Nieto [Nieto07]. Este algoritmo controla el proceso de búsqueda de soluciones, interactuando con un conjunto de estrategias heurísticas. Las soluciones son creadas y modificadas utilizando algoritmos deterministas que manejan permutaciones aleatorias de objetos. Se incorporan límites para estimar valores cercanos al óptimo que son utilizados para delimitar el espacio de búsqueda y como criterio de paro del algoritmo. Los resultados de los experimentos muestran que el algoritmo híbrido tiende a converger rápidamente hacia el óptimo con muy bajo nivel de error.

Al analizar los trabajos anteriores se llegó a las siguientes conclusiones:

1. Actualmente, parece ser que el mejor algoritmo del estado del arte es la estrategia HI-BP [Alvim04]. Al ser probada en 1587 instancias de la literatura (descritas en el trabajo del autor, así como en el capítulo 5 de esta tesis) HI-BP resuelve (de manera óptima) el 99.68% de éstas, y supera las mejores soluciones reportadas para once instancias. Solamente el valor óptimo de cinco instancias no pudo ser encontrado. No parece existir en la literatura otro método aislado que haya sido capaz de resolver tantas de las instancias consideradas o que muestre la robustez de HI-BP.
2. El algoritmo HGGA-BP [Nieto07] alcanza resultados prometedores, sin embargo hubo varios casos para los cuales dicho algoritmo no logra obtener las mejores soluciones conocidas. Al probar la estrategia con las 1587 instancias de la literatura sugeridas por Alvim, el algoritmo muestra un 88.78% de efectividad, resolviendo de manera óptima 1409 instancias y utilizando un contenedor de más en las 178 instancias restantes. Esto muestra un área de oportunidad en el análisis de desempeño formal de la estrategia. Un estudio integral del proceso de optimización podría ayudar a explicar su desempeño y mejorar los resultados mostrados por éste algoritmo.

3. Los resultados alcanzados por los algoritmos del estado del arte presentados en esta sección, así como las estrategias que incluyen, serán tomados en cuenta al realizar el análisis de desempeño del algoritmo HGGGA-BP. Las contribuciones de cada uno de los trabajos podrían ayudar a sugerir cambios en la estructura de dicho algoritmo para mejorar su comportamiento y efectividad.

3.2 OBTENCIÓN DE EXPLICACIONES DEL DESEMPEÑO ALGORÍTMICO

Existe una gran cantidad de artículos que tratan de relacionar las características de algunas clases de instancias de problemas específicos con el desempeño de varios algoritmos. Un ejemplo es el trabajo sobre problemas de optimización combinatoria NP-hard realizado por investigadores como Angel y Zissimopoulos [Angel98], Merz y Freisleben [Merz00], Hoos y Stützle [Hoos04], Drezner y Taillard [Drezner05], Konak y Kulturel [Konak05] y Chevalier [Chevalier06]. 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.

Un ejemplo de este tipo análisis es el trabajo realizado por Chevalier [Chevalier06] en el que se estudian, de manera experimental, diferentes configuraciones de los parámetros de un algoritmo Genético para resolver el problema de diagnóstico de fallas múltiples, tratando de obtener un balance correcto entre la exploración y la explotación del espacio de búsqueda para obtener mejores individuos. El desempeño del algoritmo es medido utilizando una métrica que considera la habilidad para producir respuestas correctas y rápidas; y su estudio es realizado a través de gráficas y tablas generales para las distintas configuraciones. Finalmente, el autor obtiene explicaciones intuitivas de cuáles podrían ser las causas de un pobre ó alto desempeño.

De manera similar, Hoos y Stützle [Hoos04] analizaron el comportamiento y desempeño de un algoritmo de Búsqueda Local Iterativa que resuelve instancias del problema MAX-SAT. Las trayectorias de búsqueda trazadas por el algoritmo para resolver

instancias del problema son caracterizadas por diferentes métricas de correlación. Las instancias del problema son clasificadas considerando la varianza de la distribución de pesos en las cláusulas (*vdpc*). Mediante un análisis tabular y estadístico de los resultados de la ejecución del algoritmo sobre diferentes clases de instancias del problema, los autores dan explicaciones acerca de la dependencia del desempeño del algoritmo sobre *vdpc* y como algunas estrategias de búsqueda pueden ser replanteadas para manejar diferentes estructuras del espacio de búsqueda.

En la actualidad existen muy pocos trabajos que proporcionen explicaciones formales sobre el comportamiento de algoritmos [Korb04]. En particular, el problema de explicar formalmente porque un algoritmo es el mejor al solucionar un conjunto de instancias permanece abierto a la investigación. La relación entre los indicadores que caracterizan los problemas y algoritmos no ha sido analizada con anterioridad para explicar porque algunos patrones relacionales implican la superioridad de un algoritmo en la solución de un problema. Un estudio de este tipo es importante, ya que puede proporcionar bases sólidas para el análisis y diseño de algoritmos. Los trabajos que se presentan en los siguientes párrafos representan los primeros esfuerzos por aplicar el modelado causal a la explicación del desempeño de algoritmos.

En el trabajo de Lemeire y Dirx [Lemeire05], se construye un modelo causal para analizar el desempeño de un algoritmo paralelo de multiplicación de matrices, con la finalidad de detectar las principales causas que originan varias anomalías observadas en el proceso de comunicación de distintas corridas del algoritmo. Inicialmente los autores identifican variables que registran información importante de cada operación de comunicación que realiza el algoritmo. Para realizar el análisis del desempeño utilizan un algoritmo de aprendizaje de estructuras causales, del software de inferencia causal TETRAD, mediante el cual construyen un modelo causal que muestra las relaciones entre las variables críticas identificadas, y permite explicar la existencia de ciertas anomalías en el envío de la información.

Pérez [Pérez07] presenta una metodología para el análisis experimental de algoritmos metaheurísticos con el propósito de identificar aspectos que influyen en su desempeño y establecer relaciones entre ellos para explicar su comportamiento al resolver un problema real. La metodología plantea un estudio sistemático de los elementos que intervienen en el proceso algorítmico asociado a una metaheurística. El estudio involucra cinco etapas: diseño experimental, identificación de variables, creación de indicadores del proceso algorítmico, construcción de modelos causales y análisis estadístico de éstos para proporcionar explicaciones del comportamiento observado.

Pérez aplica la metodología al análisis del desempeño de ocho variantes del algoritmo de Búsqueda Tabú en la solución del problema de distribución de objetos en contenedores. Se encontraron relaciones que revelaron que este algoritmo se desempeña mejor cuando se incorporan estrategias que le permiten una mayor exploración del espacio de búsqueda, como iniciar con una solución totalmente aleatoria, y utilizar varios operadores de búsqueda local.

Landero [Landero08] aborda el problema de la explicación del desempeño de algoritmos con una metodología similar a la propuesta en el trabajo de Pérez, sin embargo, en lugar de partir de un análisis experimental para identificar variables e indicadores del proceso algorítmico, se obtienen índices que caracterizan las instancias del problema y la estructura interna del algoritmo. El modelo causal encontrado es validado al utilizarlo para predecir el mejor algoritmo para solucionar una instancia de un problema. Las explicaciones formales obtenidas del análisis causal son utilizadas para incluir pequeñas reglas en algoritmos elementales y mejorar su desempeño.

Landero analiza el desempeño de dos variantes del algoritmo de Búsqueda Tabú y de dos variantes del algoritmo de Aceptación por Umbral para el problema de empaqueo de objetos en contenedores (BPP). El proceso algorítmico de solución es caracterizado con índices del comportamiento interno del algoritmo e índices de la trayectoria de búsqueda. La calidad de la solución encontrada y el tiempo en que se obtuvo son

utilizadas para crear un índice de la región de dominación de un algoritmo con respecto a otros. Posteriormente, mediante un análisis gráfico y estadístico se identifican un conjunto mínimo de índices significativos necesarios para realizar el modelado causal.

3.2.1 ANÁLISIS COMPARATIVO DE TRABAJOS RELACIONADOS

Algunos trabajos relacionados son presentados en la Tabla 3.1. La segunda columna indica, para cada trabajo, si se realizó un análisis experimental inicial (*AEI*) con el objetivo de obtener variables e indicadores del proceso algorítmico. La tercera y cuarta columnas indican si el análisis del algoritmo se realiza de manera conjunta con la información de instancias del problema; este análisis puede realizarse sobre la descripción de la instancia del problema (*IDP*, Índices de Descripción del Problema) ó de una muestra del espacio de soluciones de la instancia del problema (*IEP*, Índices del Espacio del Problema).

Las columnas de la quinta a la séptima indican si la información del algoritmo es contemplada en el análisis; dicha información puede ser obtenida del comportamiento del algoritmo con sus principales parámetros de control (*ICA*, Índices de Comportamiento del Algoritmo), trayectoria de búsqueda (*ITB*, Índices de Trayectoria de Búsqueda) y estructura interna del algoritmo (*IEI*, Índices de Estructura Interna). La octava columna indica si los trabajos relacionados han realizado explicaciones formales del desempeño de algoritmos. La columna final muestra si los trabajos realizan cambios en la estructura interna de los algoritmos para mejorar su desempeño.

El análisis de trabajos relacionados revela que algunos de ellos [Hoos04, Konak05, Chevalier06] proveen explicaciones a través de un análisis tabular de resultados experimentales del comportamiento de un algoritmo sobre un conjunto de instancias de un problema. Otros trabajos [Lemeire05, Pérez07, Pérez08] llevan a cabo un análisis causal usando algoritmos de aprendizaje de estructura y de parámetros, y proveen un modelo formal del comportamiento de un algoritmo.

Tabla 3.1 Trabajos relacionados con la caracterización del desempeño de algoritmos.

Trabajo	Identificación Inicial de Variables <i>AEI</i>	Información Analizada					Explicaciones Formales	Rediseño de Estructura
		Índices de las Instancias del problema		Índices del Algoritmo				
		<i>IDP</i>	<i>IEP</i>	<i>ICA</i>	<i>ITB</i>	<i>IEI</i>		
[Hoos04]		✓			✓			
[Konak05]						✓		
[Lemeire05]				✓			✓	
[Chevalier06]				✓				
[Pérez07]	✓		✓	✓			✓	
[Landro08]		✓	✓	✓	✓	✓	✓	
Este trabajo	✓	✓	✓	✓	✓	✓	✓	

El trabajo de Pérez [Pérez07] explica formalmente porqué un algoritmo se desempeña mejor al resolver un subconjunto de instancias, aunque, su explicación es limitada, pues no se incluyen todos los índices del problema y del algoritmo, sin embargo, el mérito de este trabajo radica en las fases iniciales pues en ellas se realiza un análisis experimental profundo para obtener variables que son tratadas mediante métodos multivariados y transformadas en indicadores del desempeño algorítmico.

La metodología de solución propuesta por Landero [Landro08] permite encontrar sistemáticamente relaciones entre índices influyentes de la dominación de un algoritmo y la estructura interna del mismo, para proveer explicaciones a través del análisis causal, además de validar y aplicar el modelo causal obtenido. Sin embargo pudiera hacer falta identificar algunos indicadores de desempeño mediante un análisis experimental inicial.

En esta tesis se propone una metodología de solución que reúne lo mejor de los trabajos anteriores para realizar un análisis formal de desempeño del proceso de optimización de BPP con el algoritmo HGGA-BP. El conocimiento obtenido a través de relaciones de desempeño y modelos generados es aplicado en el rediseño de la estructura interna del algoritmo. La metodología es validada al obtener tajantes mejoras en el desempeño del algoritmo rediseñado.

Capítulo 4

METODOLOGÍA PARA EL ANÁLISIS EXPERIMENTAL DE ALGORITMOS

En el Capítulo 2 (Sección 2.3) se presentaron los principales avances en el análisis experimental de algoritmos metaheurísticos. Posteriormente (Sección 2.4), se definieron las etapas sustanciales de la metodología experimental, propuesta en este trabajo, para la caracterización y análisis de desempeño del proceso de optimización algorítmico. Asimismo (Sección 2.4.4), se hizo hincapié en la necesidad de utilizar métodos de análisis exploratorio para la obtención de relaciones de desempeño y de modelos de explicación del comportamiento algorítmico.

En este capítulo se presenta, de manera detallada, la metodología propuesta y su aplicación en el estudio del algoritmo genético híbrido para el problema de empaqueo de objetos en contenedores, HGGA-BP. Las explicaciones obtenidas en dicho estudio permiten comprender el comportamiento del algoritmo e identificar posibles cambios en su estructura interna para mejorar su desempeño. La metodología expuesta se sugiere como una guía para el análisis experimental de algoritmos metaheurísticos.

4.1 METODOLOGIA DE ANÁLISIS EXPERIMENTAL PARA EL PROCESO DE OPTIMIZACIÓN ALGORÍTMICO

En la Figura 4.1 se presenta la metodología propuesta para el análisis experimental de algoritmos metaheurísticos en problemas de optimización.

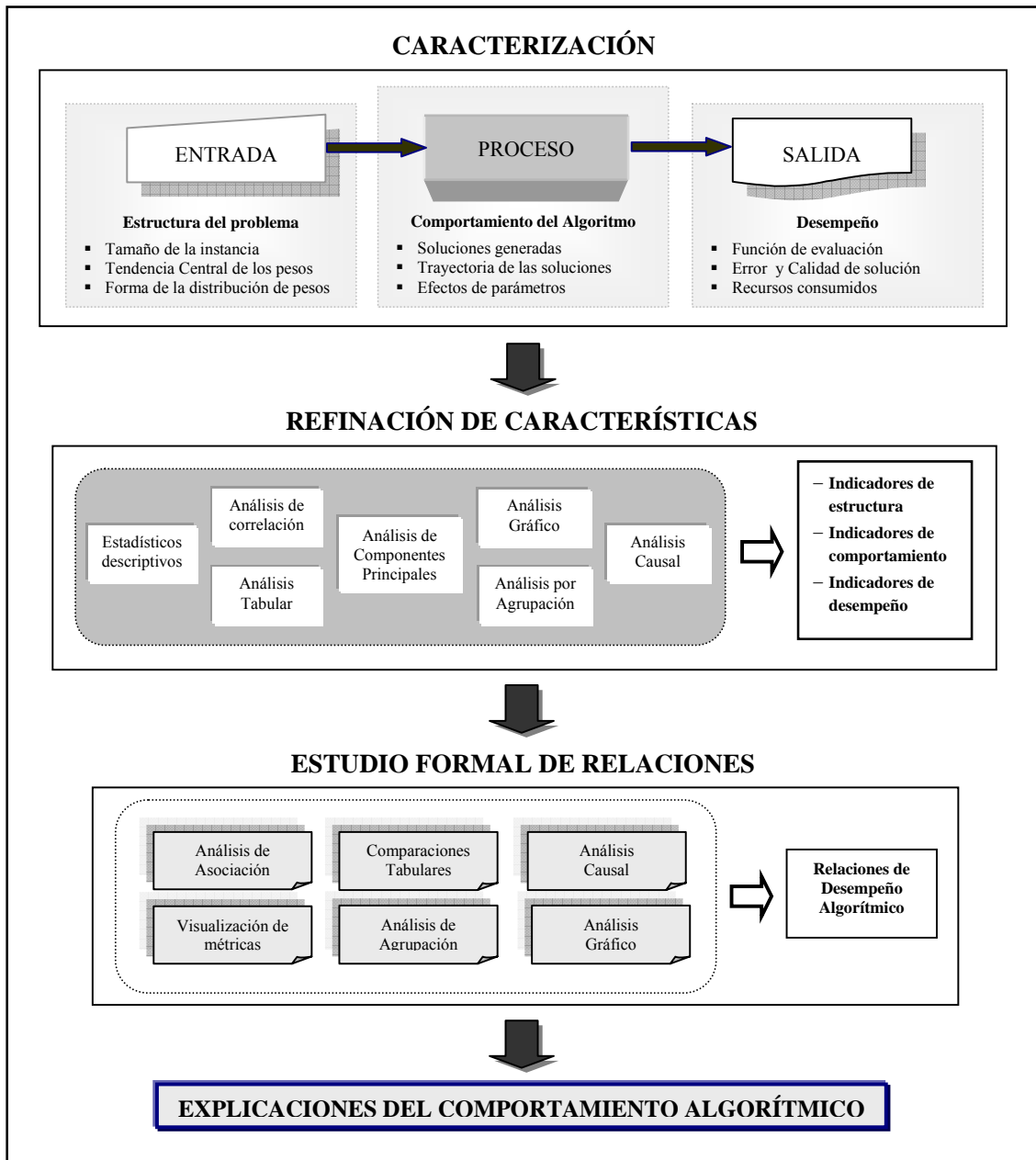


Figura 4.1 Esquema de la metodología propuesta.

El objetivo principal de la etapa de *caracterización* es identificar factores importantes, factibles de medición, en cada una de las fases del proceso de optimización. Estos factores son caracterizados a través de funciones de caracterización (métricas) que proporcionan información útil para describir el desempeño algorítmico. En la segunda etapa, *refinación de características*, las métricas definidas en la etapa de caracterización son analizadas mediante técnicas exploratorias con la finalidad de descartar variables incorrectas, redundantes o irrelevantes. En caso de ser necesario nuevas variables son definidas mediante el uso de técnicas de análisis multivariado.

En la tercera etapa, *estudio formal de relaciones*, se realiza un análisis exploratorio de las características del proceso de optimización con el propósito de obtener relaciones de desempeño que expliquen el comportamiento del algoritmo metaheurístico estudiado. Para este fin se utilizan métodos estadísticos de análisis multivariado, análisis tabulares y gráficos, técnicas de visualización de datos y análisis causal. El conocimiento obtenido, producto de la metodología de análisis propuesta, permite comprender el comportamiento del algoritmo metaheurístico estudiado, explicando de qué manera su desempeño es afectado por las características que lo causan, visualizando posibles mejoras en su estructura.

En las siguientes secciones se muestra la aplicación de la metodología propuesta en la caracterización y análisis de desempeño de un algoritmo genético híbrido del estado del arte que soluciona el problema de empaqueo de objetos en contenedores (BPP). El algoritmo objeto de análisis, HGGA-BP, fue presentado en el capítulo anterior (Sección 3.1) y la definición de BPP se incluye en el capítulo 2 (Sección 2.1).

4.2 ETAPA 1. CARACTERIZACIÓN DEL PROCESO DE OPTIMIZACIÓN

En esta etapa se lleva a cabo 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 métricas que caracterizan estos factores.

4.2.1 CARACTERIZACIÓN DEL PROBLEMA DE INVESTIGACIÓN

La estructura de una instancia de BPP es una medida importante para predecir el comportamiento que tendrá un algoritmo metaheurístico 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 sobre un algoritmo de solución. El reto es la formulación de indicadores que cuantifiquen estos factores.

Descripción del Conjunto de Casos de Prueba

Con el objetivo de validar la metodología propuesta, evaluar el desempeño del algoritmo de interés (HGGA-BP) y comparar sus cualidades con las de otros algoritmos del estado del arte se llevó a cabo la recopilación de 1,668 casos estándar reconocidos por la comunidad científica. Dichas instancias de prueba fueron tomadas de sitios de Internet reconocidos [Beasley90, Klein09, ESICUP, CaPaD].

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 (GRUPO-I) 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} . El valor de la solución óptima para cada una de estas instancias es conocido [Alvim03].

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 (GRUPO-II) 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. Los valores óptimos de las 1210 instancias que integran este grupo son conocidos [Klein09, Alvim03].

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, 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 entre 20000 y 35000. Los pesos de los objetos se encuentran ampliamente dispersos, pues existe una gran cantidad de valores diferentes para los objetos. El número de elementos por contenedor cae entre tres y cinco.

El tercer grupo (GRUPO-III) 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-hard (difíciles de resolver por la heurística FFD) [Schwerin97, Schwerin99]. La tercera clase también ha sido considerada difícil [Wäscher96]. Los valores óptimos de las 217 instancias son conocidos [Schoenfeld02, Alvim03].

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.

Las instancias Gau 1 fueron el único conjunto (de los analizados por Alvim) en el que la heurística híbrida HI-BP, no pudo obtener todas las soluciones óptimas. El algoritmo sólo resuelve 12 de las 17 instancias [Alvim04].

El grupo cuatro (GRUPO-IV) contiene instancias difíciles que fueron considerados en problemas de corte de una dimensión (one-dimensional Cutting Stock Problem). Los resultados óptimos sólo son conocidos para las instancias de la clase Hard28 [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. El número de objetos n varía entre 160 y 200. La capacidad del contenedor c es 1000.
2. **53NIRUP** [CaPaD]: Conjunto de 53 instancias propuestas por Schoenfield [Schoenfield02] como selección para una prueba exhaustiva. Para todas las instancias la capacidad del contenedor es $c = 1000$, y el número de objetos varía $n = 20, 40, 60, \dots, 200$. Los pesos se distribuyen en variedad de rangos.

Los primeros tres grupos fueron utilizados por Alvim [Alvim04] para el análisis experimental de su heurística híbrida (HI-BP), los resultados obtenidos por el autor fueron descritos en el capítulo 3 (Sección 3.1).

Índices de Caracterización

La información descriptiva de cada instancia del problema se caracteriza por medio de índices que utilizan la información de los parámetros del problema de dicha instancia. En el capítulo 2 (Sección 2.4.2) se presentó un conjunto de índices de caracterización de BPP propuestos en la literatura [Cruz04, Álvarez06], que han sido utilizados por otros investigadores [Pérez07, Landero08] mostrando resultados satisfactorios. Dichos índices son retomados en este trabajo.

El conjunto de índices fue estudiado, analizando si permitían discriminar entre las diferentes instancias de BPP. 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. La Figura 4.2 muestra la gráfica de una

instancia de la clase `u_1000` (GRUPO-I, 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 como porcentaje de la capacidad del contenedor ($0 < w_i / C \leq 1$). El eje vertical cuantifica el número de objetos en cada porcentaje de pesos. 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.

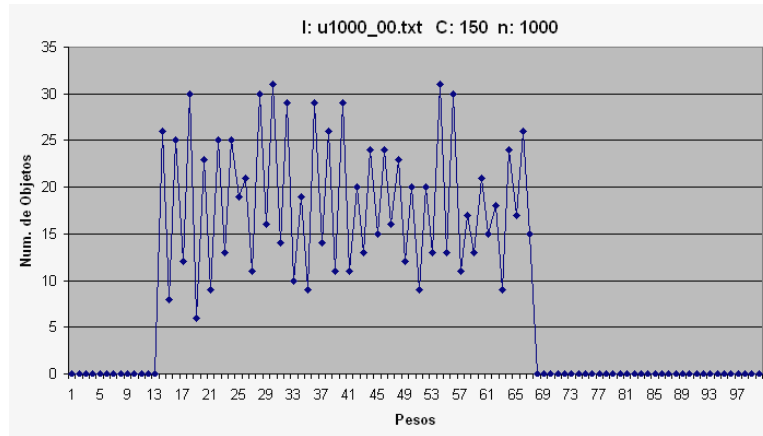


Figura 4.2 Gráfica de distribución de pesos para la instancia `u1000_00`.

La Figura 4.3 muestra gráficas representativas de cada uno de los conjuntos de instancias que conforman los cuatro grupos objeto de estudio. 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 conjuntos de instancias y ayuda a descubrir factores característicos. Los índices propuestos por Cruz y Álvarez logran caracterizar la mayor parte de la estructura de una instancia de BPP. Sin embargo parece ser que hay ciertos aspectos, que no fueron tomados en cuenta por los autores y que permiten puntualizar diferencias entre instancias.

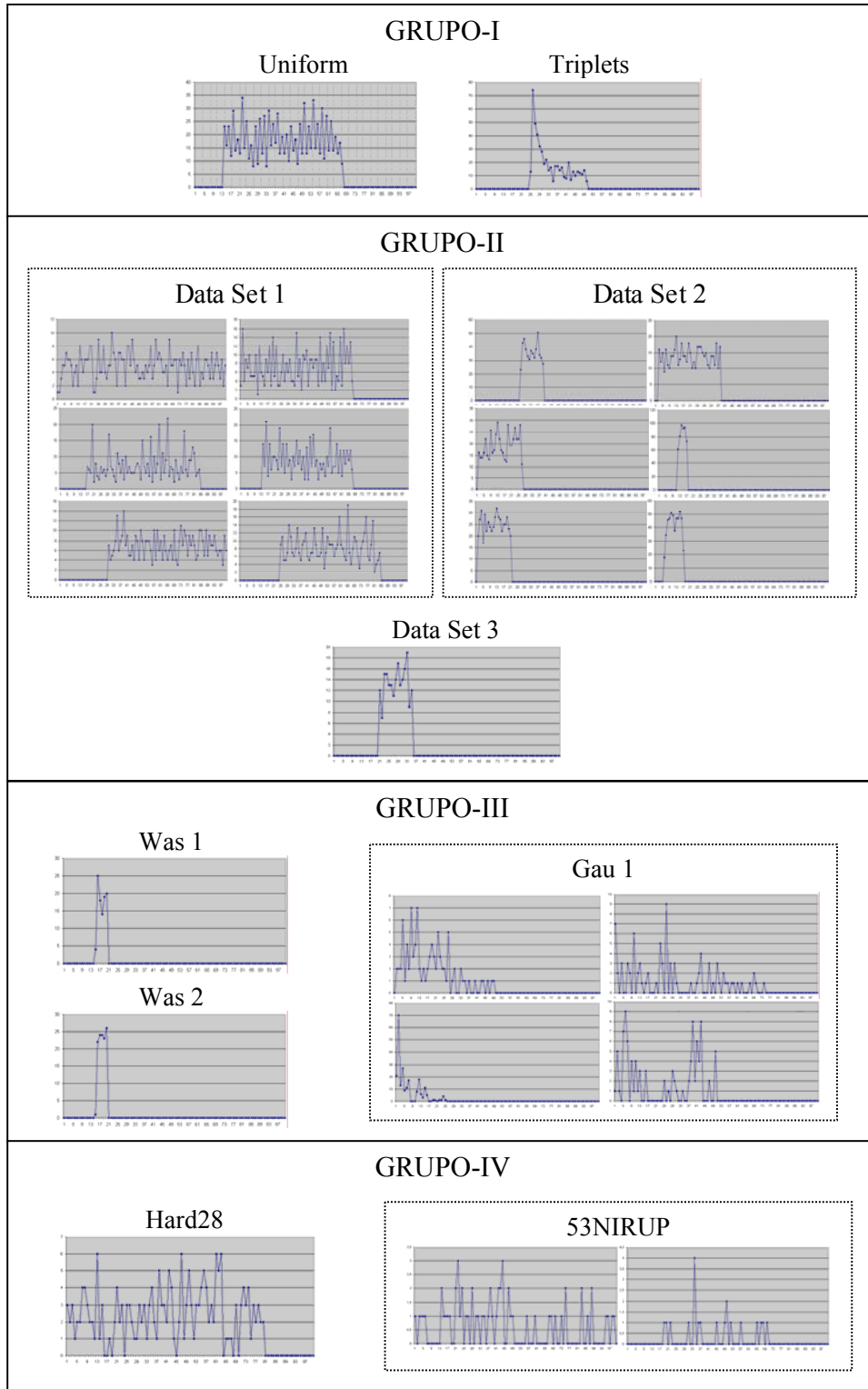


Figura 4.3 Ejemplos de gráficas de distribución de pesos para los cuatro grupos de instancias.

En este trabajo de investigación se proponen cinco nuevos índices de caracterización. Estas nuevas métricas 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 con la que se repite cada peso; y, contribuir en la caracterización de la forma de la distribución de los pesos. Dichos índices son definidos a continuación.

Menor representa el peso del objeto más pequeño, en relación al tamaño del contenedor y permite ubicar el principio de la distribución de pesos. La métrica es definida por la Expresión 4.1, donde W es el arreglo de pesos de objetos, c es la capacidad del contenedor y $\min(W)$ es una función que devuelve el peso menor.

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

Mayor cuantifica el peso del objeto más grande, en relación al tamaño del contenedor y ubica el final de la distribución de pesos. La medida es calculada con la Expresión 4.2, donde W es el arreglo de pesos de objetos, c es la capacidad del contenedor y $\max(W)$ es una función que devuelve el peso mayor.

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

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 Expresión 4.3 define esta métrica, 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 ($1 \leq \hat{n} \leq n$).

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

MaxRepe representa la frecuencia máxima con la que un peso se repite en el conjunto de objetos. La métrica es definida por la Expresión 4.4, 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.

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

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 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 pesos de los objetos

$e = \frac{n}{4}$, número esperado de objetos en una cuarta parte del rango

$r_0 = 0$, parte inicial del rango

$r_1 = \min(W) + \frac{R}{4}$, primera división del rango

$r_2 = \min(W) + \frac{R}{2}$, segunda división del rango

$r_3 = \min(W) + \frac{3R}{4}$, tercera división del rango

$r_4 = \max(W)$, parte final del rango

$B_j \subset W \left| \bigcup_{j=1}^4 B_j = W \right.$, una cuarta parte del rango de distribución de pesos

El conjunto de objetos contenido en cada subrango es:

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

La Expresión 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.

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

4.2.2 CARACTERIZACIÓN DEL COMPORTAMIENTO ALGORÍTMICO

Los algoritmos metaheurísticos híbridos incluyen estrategias que tratan de ajustar el proceso de solución al problema de optimización que resuelven. Para estos algoritmos, la medición de funciones de caracterización debe estar ligada directamente con las técnicas heurísticas utilizadas. En esta sección se describe el algoritmo genético estudiado (HGGA-BP) y los índices propuestos para caracterizar el comportamiento del algoritmo en ejecución.

Algoritmo Genético para el Problema de Empacado de Objetos: HGGA-BP

La estrategia metaheurística objeto de estudio de este trabajo de investigación es un algoritmo genético híbrido de agrupación para el problema de empaqueo de objetos en contenedores (Hybrid Genetic Grouping Algorithm for Bin Packing, HGGA-BP). Este algoritmo incorpora estrategias heurísticas y criterios que, al aplicarse en la creación y evolución de individuos, obtienen buenas soluciones en tiempos cortos [Nieto07]. El Algoritmo 4.1 describe el procedimiento general.

El proceso comienza generando una población inicial para lo cual se crea un conjunto de individuos en forma no determinista (Líneas 4-14 del Alg. 4.1). 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 con base a la aptitud, se selecciona un conjunto de individuos para aplicarles el operador evolutivo 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. El resultado final del algoritmo es el individuo más apto de todo el proceso evolutivo.

ALGORITMO GENÉTICO HGGA-BP

```

1 Inicio
2   Inicializar parámetros
3   para cada generación
4     para cada individuo actual de la población
5       nuevo = Generar individuo
6       Calcular aptitud del individuo nuevo
7       si generación > 1 y aleatorio[0-1) <= 0.95
8         si aptitud(nuevo) > aptitud(actual)
9           actual = nuevo
10        fin si
11      si no
12        actual = nuevo
13      fin si
14    fin para
15    Seleccionar individuos a cruzar de forma proporcional
16    Aplicar Cruzamiento por Agrupación y generar nuevos individuos
17    Sustituir los peores individuos de la población por los nuevos individuos
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.

Los principales *parámetros del algoritmo* y los valores para éstos, que permitieron al autor [Nieto07] obtener los mejores resultados son definidos a continuación.

- *POBLACIÓN*. Indica el número de individuos (soluciones) que integrarán la población y evolucionarán, de generación en generación, para crear mejores soluciones. El tamaño de población utilizado es de 50 individuos.

- *CORRIDAS*. Establece el número de veces que el programa será ejecutado, para verificar su robustez y comprobar su estabilidad. La literatura recomienda fijarlo en 30 corridas.
- *GENERACIONES*. Determina el número máximo de generaciones que el algoritmo iterará generando individuos y aplicando operadores de cruce y muta. El autor [Nieto07] fija el parámetro en 100 generaciones indicando que este número es suficiente para obtener una buena solución.
- *CRUZAMIENTO*. Indica el porcentaje de individuos de la población que se cruzarán entre ellos en cada generación, para producir nuevos individuos. Cruzar el 30% de los individuos ha mostrado buenos resultados.
- *MUTACIÓN*. Establece el porcentaje de individuos de la población que son mutados en cada generación, para producir pequeñas modificaciones en su estructura y aptitud. En la configuración actual 50% de los individuos mutan en cada generación.

Para la *generación de individuos* se utilizan estrategias heurísticas deterministas para BPP (descritas en la Sección 2.2.1), soluciones distintas son obtenidas al manejar diferentes métodos y órdenes aleatorios del conjunto de objetos a acomodar. Las estrategias de construcción de individuos inician con el cálculo de un límite inferior de contenedores (el número mínimo de contenedores que se utilizaría en el mejor de los casos). Una vez calculado dicho límite, los objetos más pesados son almacenados cada uno en un contenedor (dentro del límite) y el resto de los objetos son acomodados con las técnica para empaqueo de objetos BF y DB3 (Sección 2.2.1). El límite inicial de contenedores, L_1 , se obtiene mediante la Expresión 4.6. Donde w_i es el peso del objeto i y c es la capacidad del contendor.

$$L_1 = \left\lceil \sum_{j=1}^n w_j / c \right\rceil \quad (4.6)$$

El *cálculo de la aptitud* de cada individuo es realizado utilizando una función de costo alternativa, introducida por Falkenauer, que evalúa el promedio de llenado de los contenedores que conforman una solución [Falkenauer92]. Dado que, según esta función, las soluciones que poseen un mayor promedio de llenado son mejores, el objetivo del algoritmo HGGA-BP es maximizar el valor de las aptitudes de los individuos que conforman la población. La expresión 4.7 define el cálculo de la aptitud, donde N 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.

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

Además del número máximo de generaciones (parámetro inicial GENERACIONES), el algoritmo utiliza el valor del límite L_1 para establecer otro *criterio de paro*. Cada vez que una solución es generada o modificada, se realiza la comparación del número de contenedores utilizado con el valor de L_1 , en caso de que estos valores sean iguales el algoritmo termina y devuelve dicha solución (óptimo teórico).

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 deben ser tomadas en cuenta al llevar a cabo la caracterización 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*

Las estrategias anteriores son detalladas en el capítulo 6, donde se realiza el rediseño y mejora del metaheurístico híbrido. Tanto los parámetros del algoritmo, como cada una de las estrategias que lo conforman contribuyen con el desempeño que se obtendrá al solucionar una instancia de BPP. La formulación de métricas adecuadas que permitan cuantificar estos factores, contribuirá con la obtención de explicaciones detalladas sobre el comportamiento algorítmico.

Índices de Caracterización

La caracterización del comportamiento del algoritmo objeto de estudio (metaheurístico HGGA-BP) posibilita: 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 que conforma el algoritmo; y evaluar la calidad de las soluciones que conforman la población. En este trabajo se ha definido un conjunto de índices que permiten medir cada uno de estos factores. Las métricas propuestas son presentadas en los siguientes párrafos.

Para caracterizar el proceso de generación de individuos se definieron tres métricas las cuales permiten evaluar el número y la calidad de los individuos nuevos, aceptados en cada generación, como parte de la población.

Nuevos representa el número promedio de individuos, de reciente creación, aceptados en cada generación. La Expresión 4.8 define esta métrica, donde ind_i es el número total de individuos aceptados en la generación i y gen indica el número de generaciones que el algoritmo iteró.

$$Nuevos = \frac{\sum_{i=1}^{gen} ind_i}{gen} \quad (4.8)$$

Los individuos *Nuevos* sólo pudieron ser aceptados: 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).

Buenos cuantifica el número promedio de individuos nuevos, aceptados por generación, que superaron la aptitud del individuo que sustituyeron. La Expresión 4.9 detalla este índice, donde b_ind_i indica el número de individuos buenos que fueron aceptados en la generación i y gen representa el número de generaciones que formaron el proceso de evolución del algoritmo.

$$Buenos = \frac{\sum_{i=1}^{gen} b_ind_i}{gen} \quad (4.9)$$

Malos mide el número promedio de individuos nuevos, aceptados por generación, cuya aptitud era menor que la del individuo que sustituyeron. La Expresión 4.10 define la función de caracterización, m_ind_i indica el número de individuos malos aceptados en la generación i y gen 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).

$$Malos = \frac{\sum_{i=1}^{gen} m_ind_i}{gen} \quad (4.10)$$

La calidad de las soluciones que conforman la población generada por el algoritmo es evaluada mediante cuatro métricas. Estas medidas permiten registrar las aptitudes de los individuos que conforman la población en cada generación. Esto posibilita el análisis del proceso evolutivo, observando en que puntos se alcanzan las mejores soluciones.

$Mejor_Aptitud_g$ es la aptitud del mejor individuo de la población en la generación g . Sea P un conjunto que incluye las aptitudes de los individuos que forma parte de la población y $\max(P)$ una función que devuelve la aptitud máxima, la Expresión 4.11 captura la aptitud del mejor individuo de la generación.

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

$Peor_Aptitud_g$ registra la aptitud del peor individuo de la población en la generación g . La Expresión 4.12 define esta medida, donde P contiene las aptitudes de la población y $\min(P)$ devuelve la aptitud mínima.

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

$Prom_Aptitud_g$ cuantifica el promedio de la aptitud de la población en la generación g . La Expresión 4.13 permite hacer el cálculo, donde p_i es la aptitud del individuo i y T indica el tamaño de la población.

$$Prom_Aptitud_g = \frac{\sum_{i=1}^T p_i}{T} \quad (4.13)$$

$Mejor_Aptitud$ caracteriza la aptitud promedio de las mejores soluciones obtenidas en el cada generación. La Expresión 4.14 define la medida, donde $Mejor_Aptitud_g$ es la aptitud del mejor individuo de la generación g y gen es el número de generaciones.

$$Mejor_Aptitud = \frac{\sum_{g=1}^{gen} Mejor_Aptitud_g}{gen} \quad (4.14)$$

Otro factor importante a estudiar, para comprender el comportamiento algorítmico, es la superficie de aptitudes de los individuos que participan en el proceso evolutivo. 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]. Este factor, es caracterizado mediante un conjunto de métricas 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 un problema (instancia de BPP) para el 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 estanque en ciertas zonas de la superficie de aptitudes, generando soluciones muy parecidas, lo que indicaría alguna inconsistencia en el mismo.

$Desv_Aptitud_g$ mide la variabilidad en la aptitud de los individuos que conforman la población de la generación g . La métrica es definida en la Expresión 4.15, que calcula la desviación estándar de las aptitudes, donde p_i es la aptitud del individuo i , \bar{p} es el promedio de las aptitudes de la población en la generación g y T indica el tamaño de la población.

$$Desv_Aptitud_g = \sqrt{\frac{\sum_{i=1}^T (\bar{p} - p_i)^2}{T}} \quad (4.15)$$

$Desv_Mejor$ representa la desviación en las aptitudes de los mejores individuos de cada generación. La Expresión 4.16 define la métrica, donde ma_i es la mejor aptitud de

la generación i , \overline{ma} es el promedio de las aptitudes de los mejores individuos del proceso evolutivo y gen indica el total de generaciones.

$$Desv_Mejor = \sqrt{\frac{\sum_{i=1}^{gen} (\overline{ma} - ma_i)^2}{gen}} \quad (4.16)$$

La *Función de Auto-correlación* (ρ_1), propuesta por Weinberg [Weinberg90], se retoma en este trabajo, para medir las asociaciones existentes entre las mejores soluciones. Para su cálculo, sea ma_i la mejor aptitud de la generación i , \overline{ma} el promedio de las aptitudes de los mejores individuos del proceso evolutivo y gen el total de generaciones. La métrica es definida por la Expresión 4.17.

$$\rho_1 = \frac{\sum_{i=1}^{gen-1} (ma_i - \overline{ma})(ma_{i+1} - \overline{ma})}{\sum_{i=1}^{gen} (ma_i - \overline{ma})^2} \quad (4.17)$$

Un valor de la función de auto-correlación cercano a uno indica que las diferencias en la aptitud entre cualquier par de soluciones vecinas es en promedio muy similar, implicando una superficie de aptitudes un tanto plana. En caso contrario, un valor cercano a cero indica que los valores de aptitud son casi independientes, y por tanto la superficie debe presentar valles y picos. Una vez más esto puede estar relacionado con las características del problema de optimización.

Finalmente, las estrategias que tienen mayor impacto en la obtención de las mejores soluciones del algoritmo son registradas a través de la métrica *Mejor_Heurística*. Este índice, obtiene la estrategia del algoritmo que permitió alcanzar el mayor número de mejores soluciones en C ejecuciones del algoritmo. La Expresión 4.18 define la medida,

donde r_i representa el número de veces que la heurística i generó la solución final del algoritmo y $\max_2(R)$ devuelve el índice de la heurística que presenta el mayor número de repeticiones.

$$\text{Mejor_Heurística} = \max_2(R) \quad (4.18)$$

4.2.3 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. Estas métricas, registran las aptitudes de las soluciones finales alcanzadas en diferentes corridas del programa, así como el número de contenedores utilizado en dichas soluciones. Un valor de aptitud igual a uno, significa que los contenedores que conforman la solución se encuentran completamente llenos, y por lo tanto, se tiene una solución óptima.

$\text{Max_Mejor}F_{BPP}$ registra la aptitud de la mejor solución obtenida, para una instancia de BPP, en C corridas del algoritmo. Sea M un conjunto que incluye las aptitudes de las C soluciones obtenidas en C ejecuciones del algoritmo, la Expresión 4.19 calcula la mejor aptitud, donde $\max(M)$ es una función que devuelve el mayor elemento del conjunto M .

$$\text{Max_Mejor}F_{BPP} = \max(M) \quad (4.19)$$

$\text{Min_Mejor}F_{BPP}$ representa la aptitud de la peor solución obtenida, para una misma instancia, en C ejecuciones del algoritmo. La Expresión 4.20 define esta métrica, donde

M el conjunto de aptitudes de las soluciones obtenidas en C corridas del algoritmo y la función $\min(M)$ devuelve la menor aptitud.

$$Min_MejorF_{BPP} = \min(M) \quad (4.20)$$

$Prom_MejorF_{BPP}$ calcula el promedio de las aptitudes de las soluciones obtenidas, para el mismo caso de prueba, en C corridas del algoritmo. La Expresión 4.21 precisa esta métrica, donde ma_i es la aptitud de la solución alcanzada en la corrida i y C es el total de ejecuciones.

$$Prom_MejorF_{BPP} = \frac{\sum_{i=1}^C ma_i}{C} \quad (4.21)$$

El *Radio teórico* es otra medida de calidad, y representa la razón de la mejor solución encontrada por el algoritmo Z_{enc} y la solución óptima Z_{opt} (ambos en cuanto al número de contenedores utilizado). La Expresión 4.22 define este índice.

$$Radio_teorico = \frac{Z_{enc}}{Z_{opt}} \quad (4.22)$$

Para caracterizar la estabilidad del algoritmo se mide la desviación estándar de los resultados obtenidos. $Desv_Z_{enc}$ cuantifica la variabilidad las soluciones encontradas en diferentes ejecuciones del algoritmo, para acomodar la misma instancia de BPP. Esta medida es obtenida mediante la Expresión 4.23, 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 C corridas. Una valor de desviación alto puede indicar que una instancia de BPP difícil de resolver o que las estrategias aleatorias incluidas en el algoritmo lo hacen un tanto inestable.

$$Desv_Z_{enc} = \sqrt{\frac{\sum_{i=1}^C (\bar{z} - z_i)^2}{C}} \quad (4.23)$$

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

La Expresión 4.24 define la métrica *Tiempo* que cuantifica el promedio del tiempo, en segundos, que el algoritmo tarda en encontrar una solución, para una instancia de BPP. El tiempo utilizado en la corrida *i* es cuantificado por t_i .

$$Tiempo = \frac{\sum_{i=1}^C t_i}{C} \quad (4.23)$$

Generación representa la generación promedio en que el algoritmo encuentra la mejor solución de una instancia. La Expresión 4.24 define este índice, donde gen_i es la generación en la que se encontró la mejor solución de la corrida *i*.

$$Generación = \frac{\sum_{i=1}^C gen_i}{C} \quad (4.24)$$

Por último, para caracterizar la estructura de la solución final obtenida, se propone la métrica *Objetos_Contenedor*. Esta medida calcula el número promedio de objetos que fueron almacenados en cada contenedor de la solución alcanzada por el algoritmo. La medida puede ser calculada a partir de la Expresión 4.25, donde Z_{enc} representa el

número de contenedores de la solución final obtenida y n es el número de objetos de la instancia de BPP.

$$\text{Objetos_Contenedor} = \frac{Z_{enc}}{n} \quad (4.25)$$

4.3 ETAPA 2. REFINACIÓN DE CARACTERÍSTICAS

En la etapa anterior se propuso el uso de un conjunto de índices para caracterización de BPP y del comportamiento del algoritmo HGGA-BP. 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 proceso de optimización. El objetivo de la refinación es encontrar un conjunto mínimo de índices de calidad, que describan, correctamente, las características del proceso de optimización. Los estudios realizados en esta etapa pretenden: eliminar medidas redundantes; descartar índices incorrectos y poco relevantes; y validar las métricas propuestas.

En el caso de la caracterización del problema, los índices propuestos (ver Sección 4.2.1) permiten conocer la estructura de una instancia de BPP, midiendo los siguientes aspectos (definidos en las Secciones 2.4.1 y 4.2.1):

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 el rango de la distribución del conjunto de pesos.
6. Diferencias entre valores que ocupan posiciones 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. Cada instancia debe ser caracterizada en las mismas unidades, para poder realizar futuros análisis en búsqueda de relaciones de desempeño. Debido a lo anterior, en el cálculo de las métricas, los pesos de los objetos fueron tomados 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>Coficiente 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>Max_Repe</i> [Este Trabajo]

A simple vista, parece ser que algunos 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 ayuda a verificar lo anterior.

En los siguientes párrafos se describen las herramientas que permitieron seleccionar el conjunto final de métricas de caracterización del proceso de optimización. El cálculo de la matriz de correlación, y de otros estadísticos descriptivos, permitió la identificación de asociaciones existentes entre conjuntos de índices. Así mismo, el estudio de gráficas, de dispersión y distribución, ayudó a descubrir la utilidad y consistencia de cada métrica. Finalmente, herramientas de análisis multivariado y análisis causal, permitieron validar el conjunto final de funciones de caracterización.

4.3.1 ESTUDIO DE LA MATRIZ 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.

Tabla 4.2 Matriz de correlación de variables de tendencia central.

	<i>Med Aritmetica</i>	<i>Med Geometrica</i>	<i>Med Armonica</i>	<i>Mediana</i>	<i>t</i>
<i>MedAritmetica</i>	1	0.9996	0.9978	0.9999	1
<i>MedGeometrica</i>	0.9996	1	0.9993	0.9996	0.9996
<i>MedArmonica</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

Si la matriz de correlación tiene colinealidad es posible que los análisis estadísticos basados en ésta no tengan sentido o 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. La métrica seleccionada fue *t*, las otras medidas fueron descartadas.

El análisis de correlación mostró también la presencia de colinealidad en las medidas que caracterizan la dispersión y la forma. Una vez más, algunas métricas fueron descartadas para eliminar la redundancia y la presencia de problemas en análisis futuros.

La Tabla 4.3 muestra el nuevo conjunto de índices, una vez eliminadas las métricas redundantes. El conjunto se redujo de 31 a 21 funciones de caracterización.

Tabla 4.3 Índices de caracterización de BPP sin colinealidad.

Medidas	Estadístico
Tamaño	p [Cruz04]
Tendencia Central	<i>Moda</i> [Álvarez06]
	t [Cruz04]
Dispersión	<i>Rango</i> [Álvarez06]
	<i>Coficiente_variación</i> [Álvarez06]
	<i>Error_estándar</i> [Álvarez06]
	d [Cruz04]
Forma	<i>Asimetría_Pearson</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>Max_Repe</i> [Este Trabajo]

4.3.2 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 seleccionado. El objetivo de este estudio, es identificar y eliminar aquellas métricas que no aporten información significativa para la caracterización de la estructura de las instancias de BPP. Índices inconsistentes o incorrectos sólo agregan información errónea en la descripción de una instancia y no permiten discriminar entre instancias de diferente naturaleza.

Después de su análisis, la métrica *moda* fue descartada, debido a que no permitía caracterizar ningún aspecto importante del problema de interés. 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).

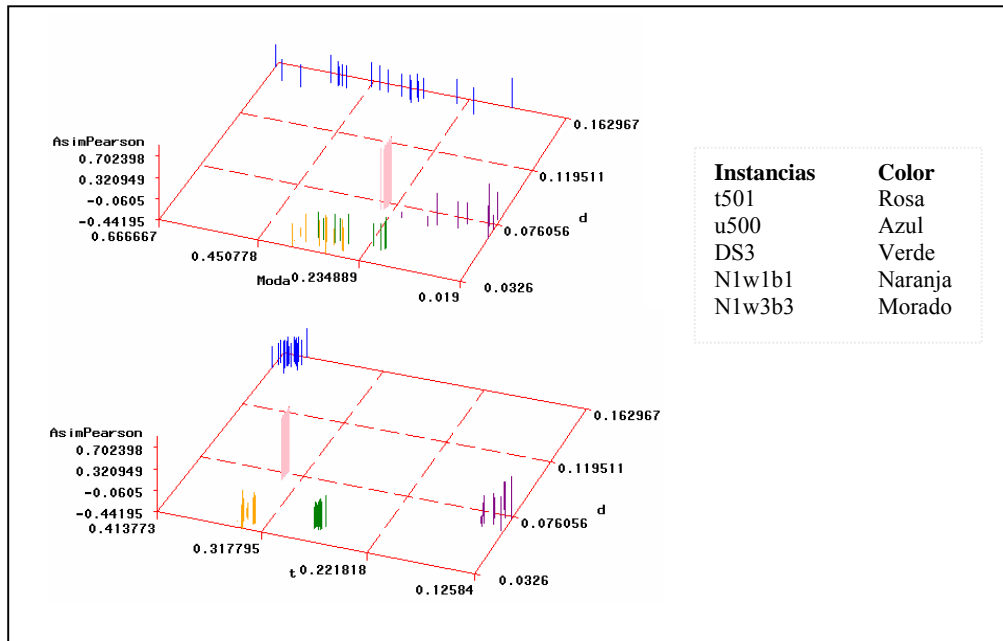


Figura 4.4 Gráficas de dispersión para *moda*, *t*, *d* y *Asimetría_Pearson* en cinco conjuntos de instancias.

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 de diferente estructura (descritas en la Sección 4.2.1). 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 la métrica no permite discriminar entre diferentes instancias. En la segunda gráfica se aprecia claramente la importancia de la información proporcionada por los índices *t*, *d* y *Asimetría_Pearson*.

El análisis exploratorio sugirió que las medidas de posición (*cuartiles*, *deciles* y *percentiles*), propuestas por Álvarez [Álvarez] para caracterizar la uniformidad de la

distribución del conjunto de pesos, tampoco aportaban información significativa para la caracterización de la estructura de una instancia de BPP. El estudio combinado, de las gráficas de distribución de las instancias (ver Sección 4.2.1) y de los valores mostrados por estas métricas, mostró que los valores de estos índices no correspondían con las características reales de las instancias y que se contradecían unos con otros.

Las Figuras 4.5 y 4.6 muestran, respectivamente, las gráficas de distribución de dos de las instancias menos uniformes (de los cuatro grupos de instancias objeto de estudio). Estas instancias pertenecen al conjunto Gau 1, GRUPO-III. Los índices de posición, no destacaron estas instancias de entre otras que presentan mayor uniformidad. En la Figura 4.7 se observa la inconsistencia de las medidas, las dos instancias menos uniformes fueron marcadas, respectivamente, con un diamante y una estrella.

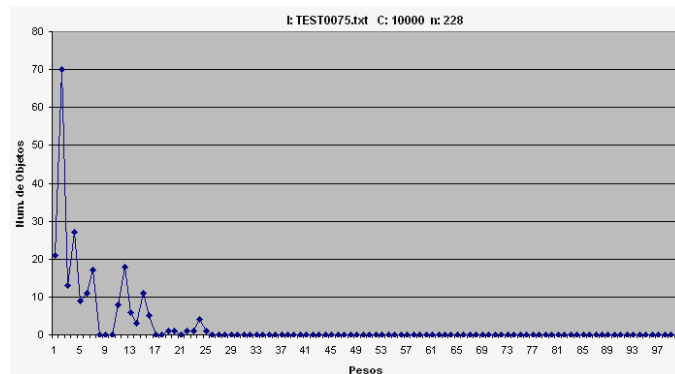


Figura 4.5 Gráfica de distribución de pesos para la instancia TEST0075.

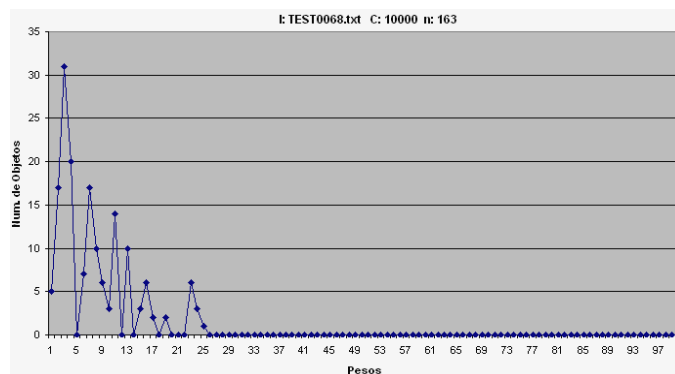


Figura 4.6 Gráfica de distribución de pesos para la instancia TEST0068.

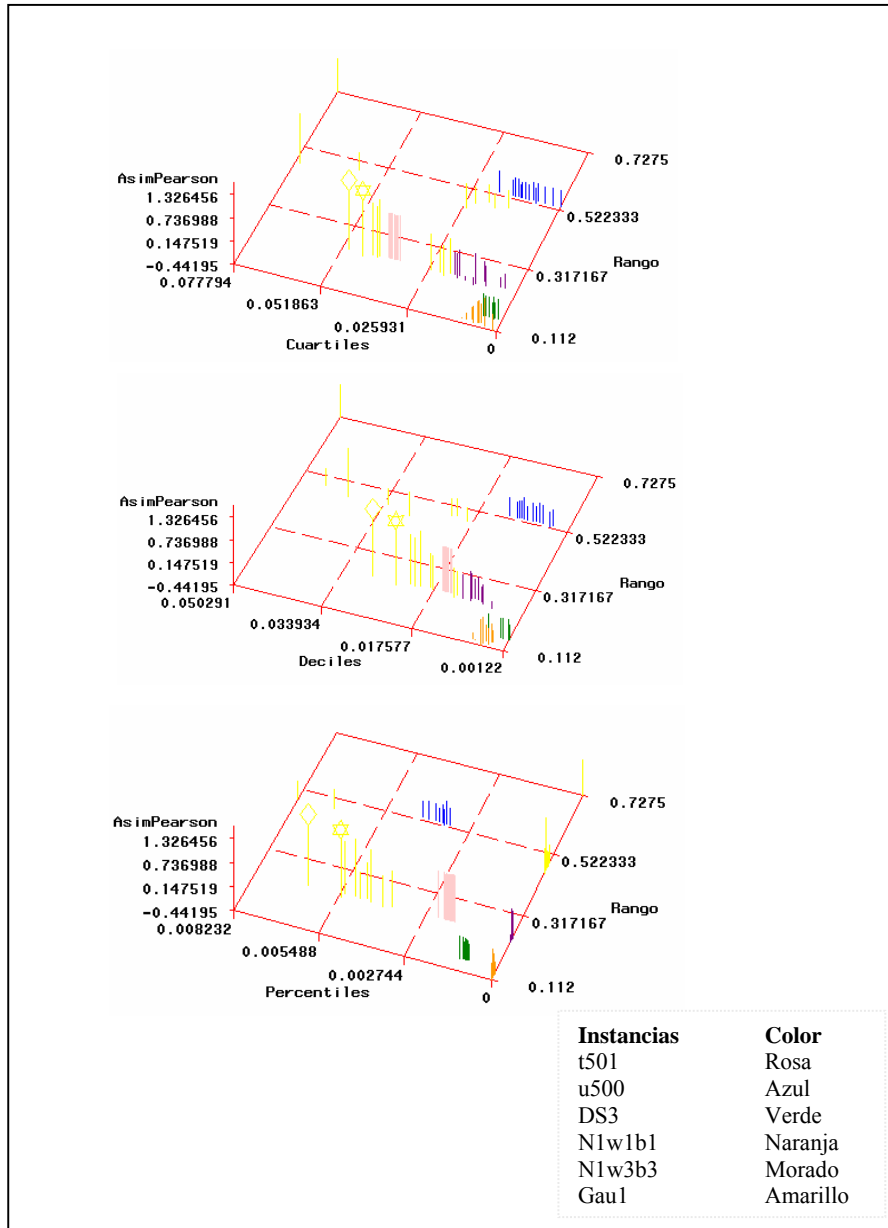


Figura 4.7 Gráficas de dispersión para índices de posición, *Rango* y *Asimetría_Pearson*.

La Figura 4.7 permite apreciar la presencia de ciertas contradicciones en los valores de las métricas. El orden del grado de “uniformidad” asignado a las instancias es diferente en las tres medidas (un valor mayor en las medidas corresponde con un menor grado de uniformidad.). La información proporcionada por estos índices, sólo introduciría “ruido” en la búsqueda de explicaciones de desempeño. La Figura 4.8 muestra una gráfica de

dispersión similar, pero ahora se hace uso de la función de caracterización *uniformidad*, propuesta en este trabajo. La figura demuestra el poder discriminatorio y la consistencia de la métrica propuesta.

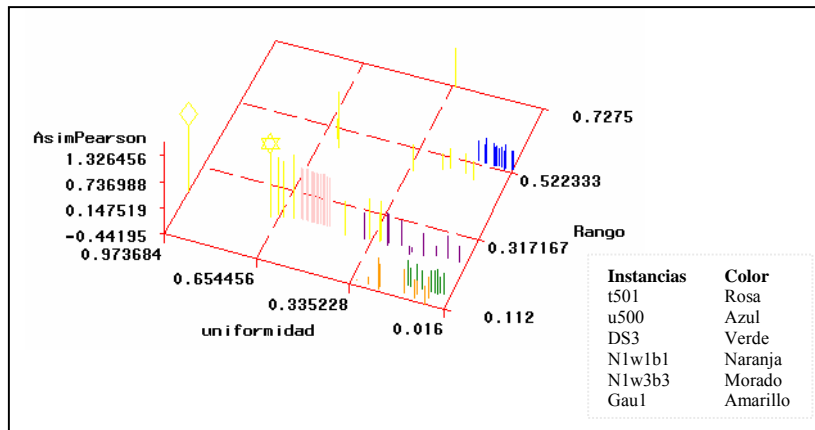


Figura 4.8 Gráfica de dispersión para las medidas de *uniformidad*, *Rango* y *Asimetría_Pearson*.

Las conclusiones obtenidas en los estudios anteriores llevaron a descartar las métricas *Asimetría_Pearson_moda* y *Asimetría_Bowley*. Pues se basan, respectivamente, en las métricas *moda* y *cuartiles*, cuya inconsistencia (para la caracterización de BPP) fue demostrada en párrafos anteriores.

Finalmente, la métrica *Error_estándar*, propuesta por Álvarez [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 el autor cometió un gran descuido al proponer este índice de caracterización.

El análisis de las métricas de caracterización del comportamiento algorítmico mostró que no existían índices incorrectos, redundantes o inconsistentes, pues cada uno captura aspectos importantes del algoritmo HGGa-BP. Conclusiones similares fueron obtenidas al analizar las funciones de caracterización para el desempeño final.

4.3.3 VALIDACIÓN DEL CONJUNTO FINAL DE MÉTRICAS

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 métricas 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	<i>Rango</i> [Álvarez06]
	<i>Coficiente_variación</i> [Álvarez06]
	d [Cruz04]
Forma	<i>Asimetría_Pearson</i> [Álvarez06]
	<i>Curtosis</i> [Álvarez06]
	<i>Uniformidad</i> [Este Trabajo]
Ubicación	<i>Menor</i> [Este Trabajo]
	<i>Mayor</i> [Este Trabajo]
Relaciones Objetos/Contenedor	f [Cruz04]
	b [Cruz04]
Multiplicidad	<i>Multiplicidad</i> [Este Trabajo]
	<i>Max_Repe</i> [Este Trabajo]

Un Análisis de Componentes Principales ayudó a confirmar la importancia y aportación del conjunto final de métricas. Este estudio fue realizado sobre las medidas obtenidas al aplicar las métricas 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 variabilidad en los pesos de los objetos y agrupa las medidas de tendencia central, dispersión y ubicación. El segundo (*Forma*), mide la forma de 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.

Los componentes principales permitieron graficar las características de los seis grupos de instancias, en la gráfica obtenida (Figura 4.9) es posible distinguir, claramente, entre los seis tipos de instancias analizadas.

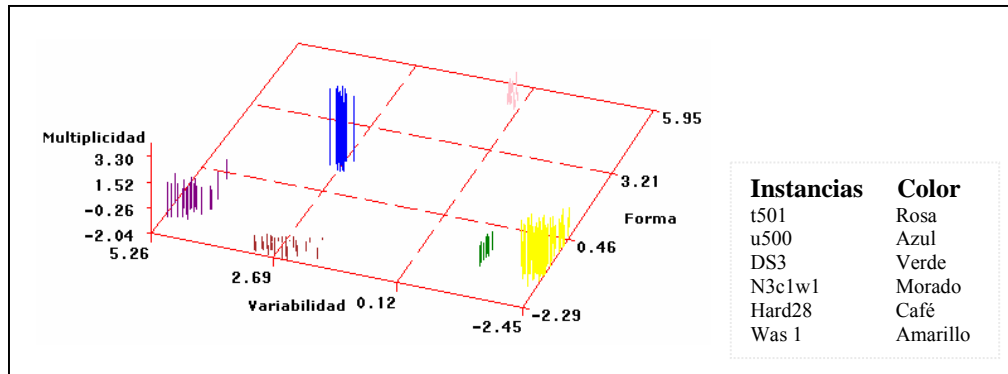


Figura 4.9 Gráfica de dispersión para los tres primeros componentes principales del conjunto de métricas.

El análisis de las gráficas de distribución de frecuencia de pesos (definidas en la Sección 4.2.1) comprobó la exactitud de la discriminación, 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. En esta figura 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.

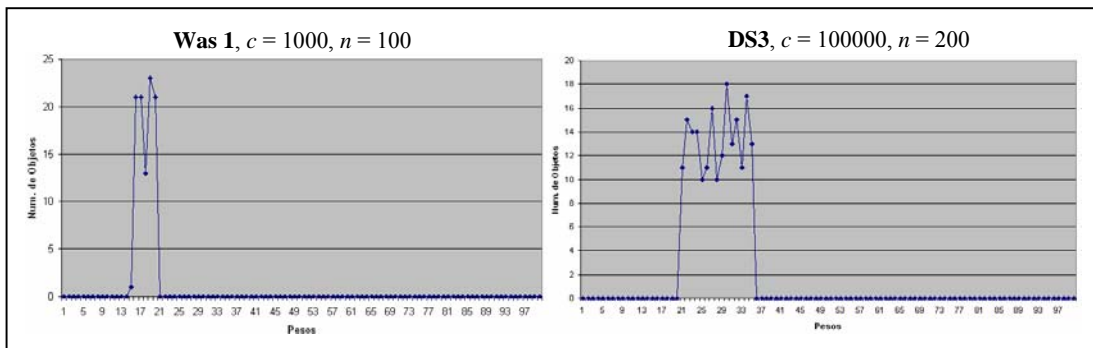


Figura 4.10 Gráficas de distribución de frecuencias de pesos para instancias Was 1 y DS3.

Por último, un análisis causal de las calificaciones de los componentes principales y de tres métricas de caracterización de desempeño (descritas en la Sección 4.2.3), 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 de desempeño final, *Prom_MejorF_{BPP}*, *Radio_teorico* y *Generación*, en los seis conjuntos de instancias (ver Figura 4.9).

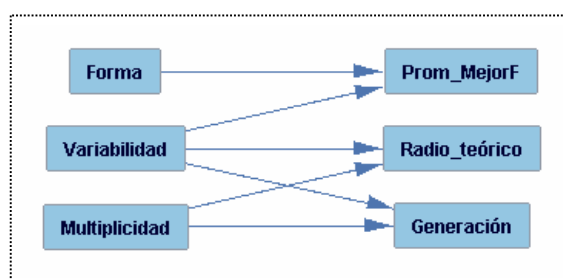


Figura 4.11 Relaciones causales entre las características del problema y desempeño de HGGA-BP.

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 por el algoritmo (*Radio_teorico*) y la generación promedio en que la mejor solución es encontrada (*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 (*Promedio_MejorF*) parece estar relacionada con la forma y la variabilidad de la distribución de los pesos. Los resultados obtenidos mostraron la validez de las métricas de caracterización de BPP seleccionadas.

En el caso de los índices de caracterización del problema, una vez más, el análisis de componentes principales (PCA), comprobó la exactitud discriminatoria de las métricas. 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.

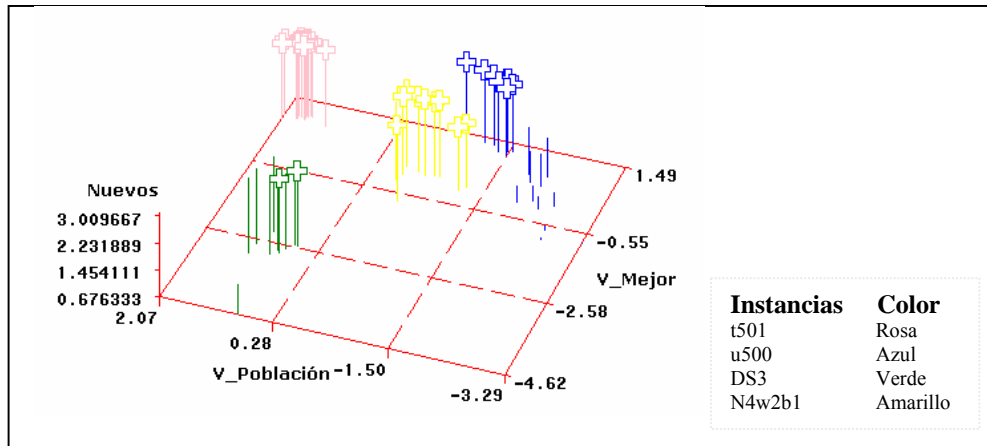


Figura 4.12 Gráfica de dispersión de los tres primeros componentes principales del conjunto de métricas.

Para este grupo de instancias (t501, u500, DS3 y N4w2b1), el PCA agrupó las métricas 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 métricas 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 las métricas del proceso de generación de algoritmos.

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 las métricas propuestas caracterizan factores del comportamiento algorítmico impactan el desempeño final. La Figura 4.13 muestra el

modelo causal obtenido en el estudio. En la figura se observa que el componente *Nuevos* (destacado en color amarillo), parece no tener relación con el desempeño final, sin embargo, este hecho puede deberse, más que a un error en el cálculo de la métrica, a una inconsistencia en el algoritmo. La conjetura anterior será demostrada al analizar las relaciones que describen el proceso de optimización.

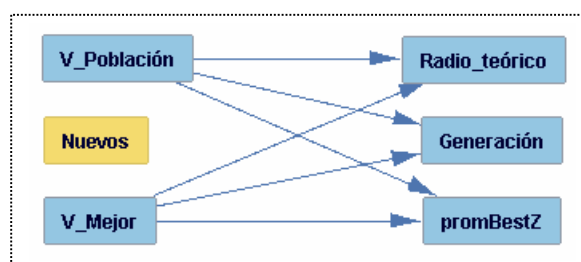


Figura 4.13 Relaciones causales entre el comportamiento y desempeño de HGGA-BP.

4.4 ETAPA 3. ESTUDIO FORMAL 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 la estructura del proceso de optimización. El objetivo principal de este estudio es adquirir conocimiento que permita explicar el desempeño del algoritmo. Las relaciones entre los índices son estudiadas mediante métodos multivariados, gráficos, estadísticos y de inferencia causal.

Para poder explicar el desempeño del algoritmo metaheurístico HGGA-BP es necesario analizar las relaciones existentes entre los índices de caracterización que definen la estructura de las instancias de BPP, las funciones de caracterización que describen el comportamiento mostrado por el algoritmo y las métricas que cuantifican el desempeño final alcanzado. Las relaciones son obtenidas estudiando la matriz de correlación, así como gráficas de dispersión y de visualización de conjuntos de variables. Inicialmente los análisis fueron realizados sobre el conjunto total de instancias (1,668 casos definidos en la Sección 4.2.1), 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 métricas 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. 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.

Métricas		Correlación
ρ_1	<i>t</i>	0.53043
ρ_1	<i>D</i>	0.46751
ρ_1	<i>Rango</i>	0.46663
ρ_1	<i>Mayor</i>	0.52167

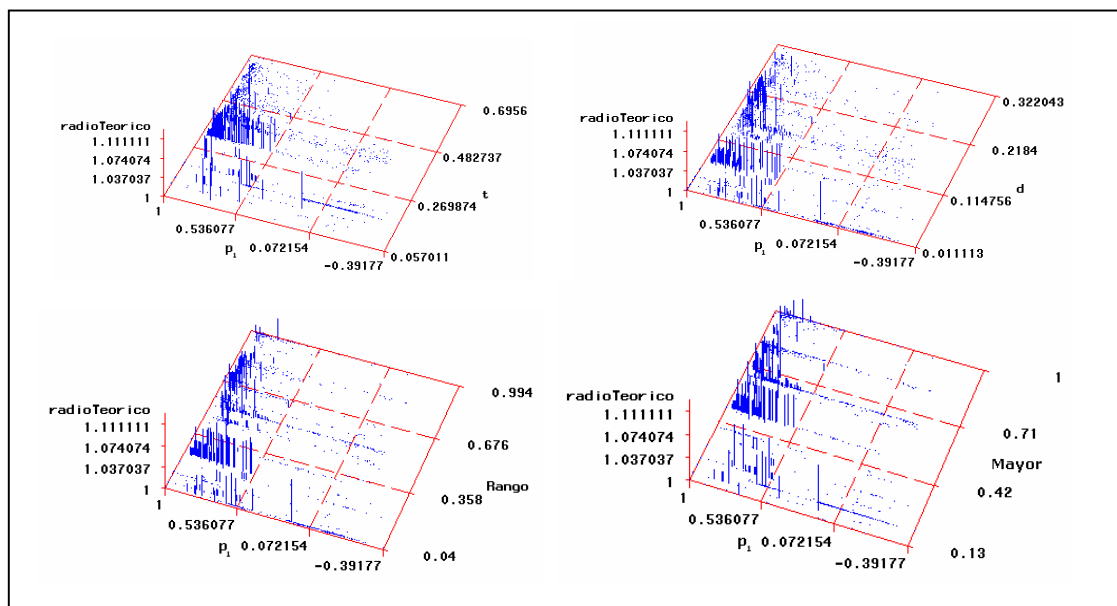


Figura 4.14 Gráficas de dispersión de índices de BPP y de comportamiento de HGGA-BP.

Las relaciones obtenidas muestran que, las asociaciones que puedan existir entre las mejores soluciones generadas por el algoritmo son causadas por la tendencia central y la variabilidad del conjunto de 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. En este tipo de instancias, en la mayoría de los casos, el óptimo es encontrado cuando las mejores soluciones generadas por el algoritmo no están muy asociadas entre si.
- 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 si, 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.

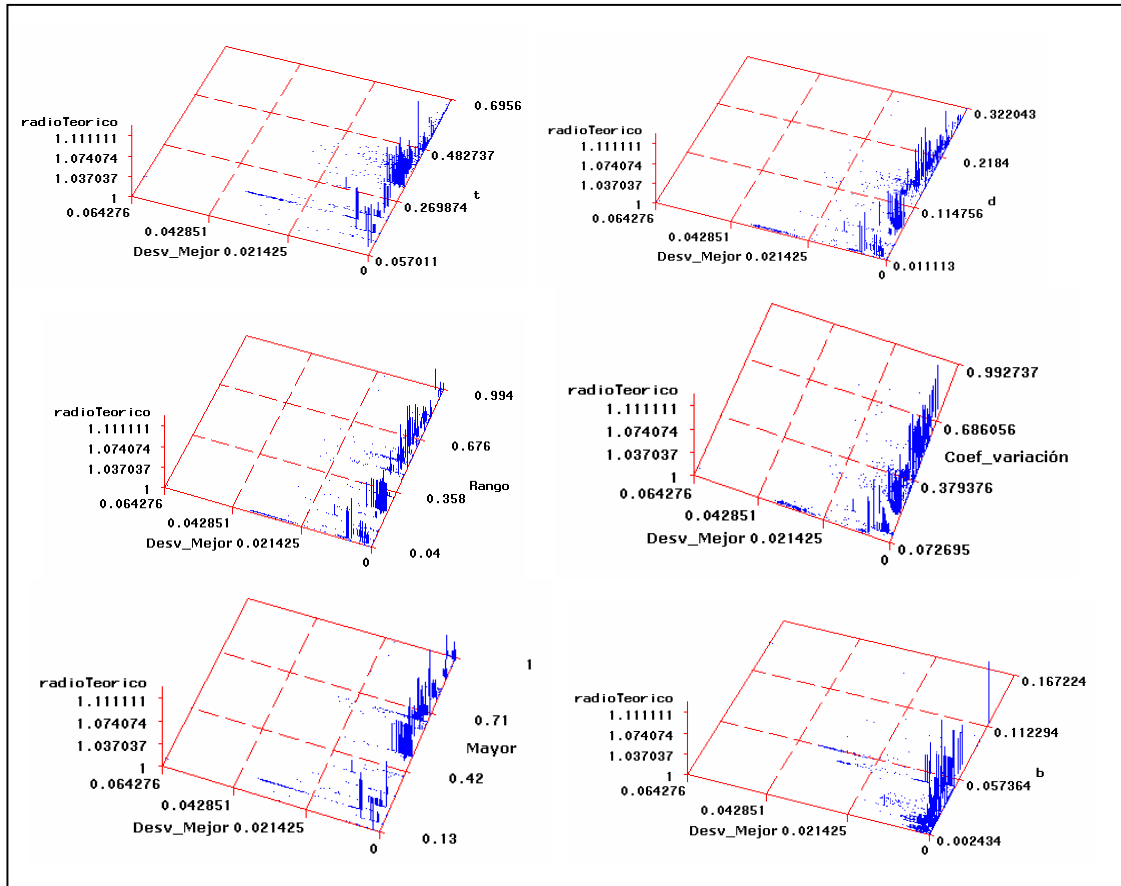


Figura 4.15 Gráficas de dispersión de índices de BPP y de comportamiento de HGGA-BP.

Por su parte, las desviaciones existentes 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 rango de variación de las mejores soluciones son: *t*, *d*, *Rango*, *Coeficiente_variación*, *Mayor* y *b*. En la Tabla 4.6 se muestra el grado de relación lineal existente entre cada par de variables.

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

	Métricas	Correlación
<i>Desv_Mejor</i>	<i>t</i>	-0.60287
<i>Desv_Mejor</i>	<i>d</i>	-0.60631
<i>Desv_Mejor</i>	<i>Rango</i>	-0.60579
<i>Desv_Mejor</i>	<i>Coeficiente_variación</i>	-0.54876
<i>Desv_Mejor</i>	<i>Mayor</i>	-0.63186
<i>Desv_Mejor</i>	<i>b</i>	0.57900

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 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 implica que los objetos son grandes y/o que son muchos), la instancia es más fácil, pues el espacio de soluciones es más pequeño y 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 atorarse 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.

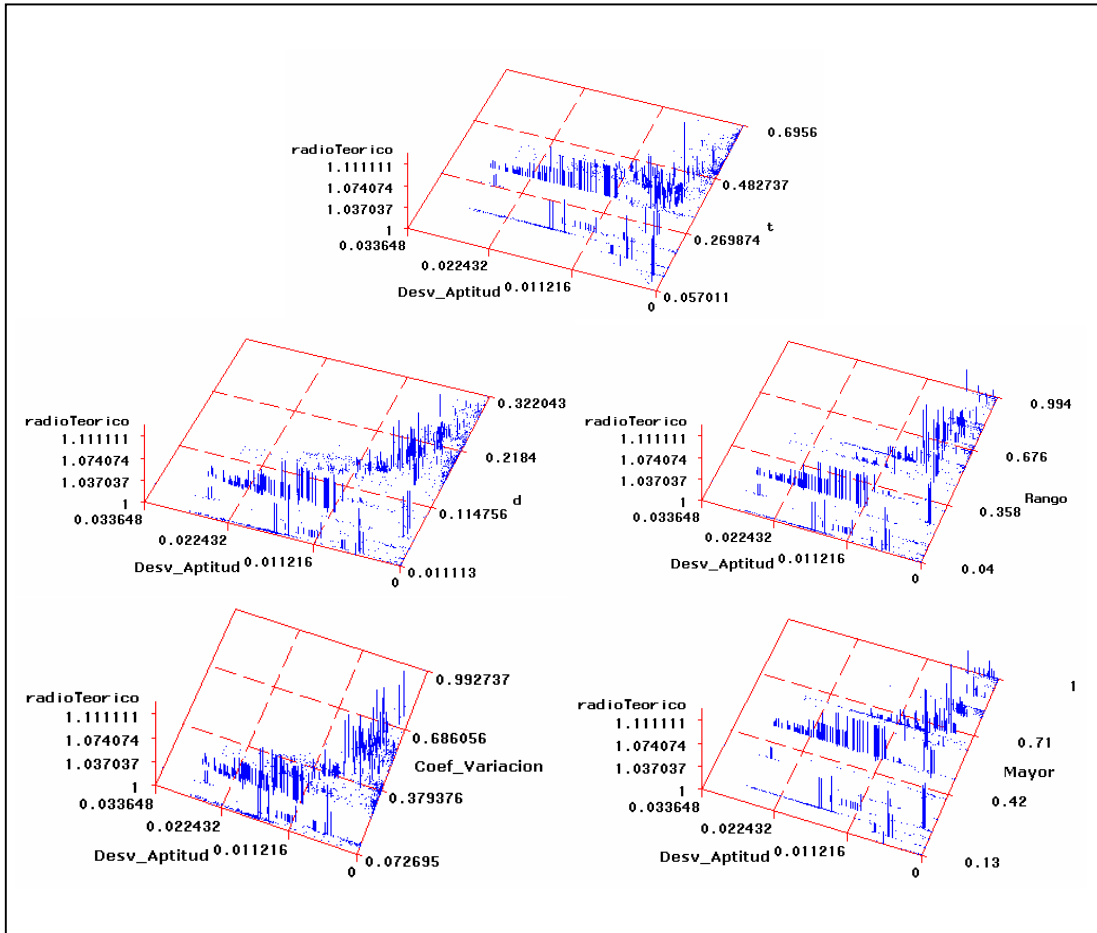


Figura 4.16 Gráficas de dispersión de índices de BPP y de comportamiento de HGGA-BP.

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 Tabla 4.6 y la Figura 4.16 detallan la intensidad y la forma de cada relación.

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.

Métricas		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>Coficiente_variación</i>	-0.53074
<i>Desv_Aptitud</i>	<i>Mayor</i>	-0.56648

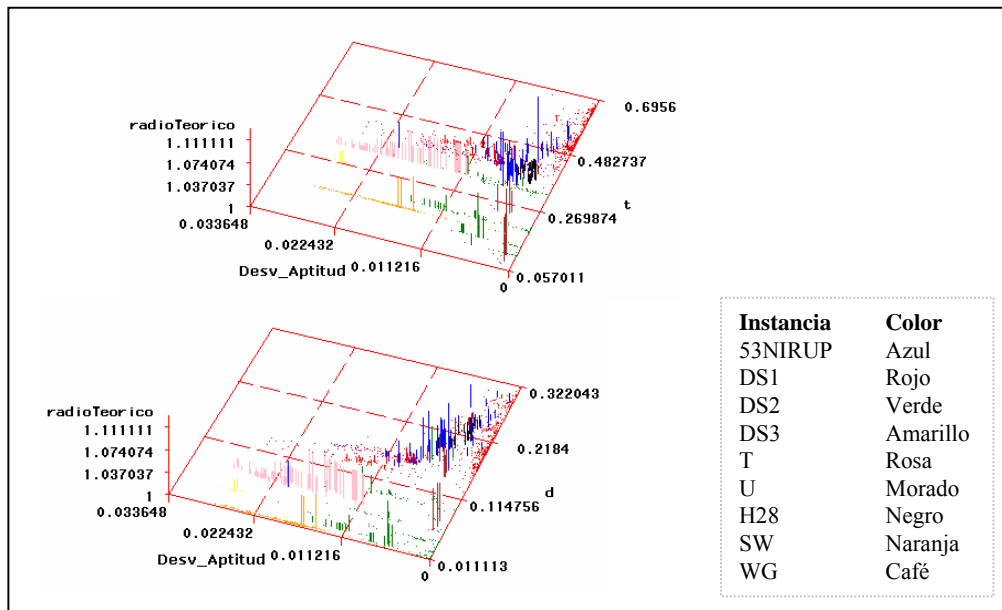


Figura 4.17 Gráficas de dispersión de índices de BPP y de comportamiento de HGGA-BP.

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 y las 53NIRUP. Los demás conjuntos incluyen también algunas instancias que no pudieron ser resueltas. La complejidad de estas instancias parece ser causada por la forma de la distribución de las frecuencias de los pesos de los objetos, así como por la multiplicidad y tamaño del problema. 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* causa: 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. 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 otra característica del problema. Para este tipo de instancias, 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.

La Figura 4.18, muestra gráficas de dispersión que ilustran el comportamiento mostrado por el algoritmo con cada tipo de instancias. 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.

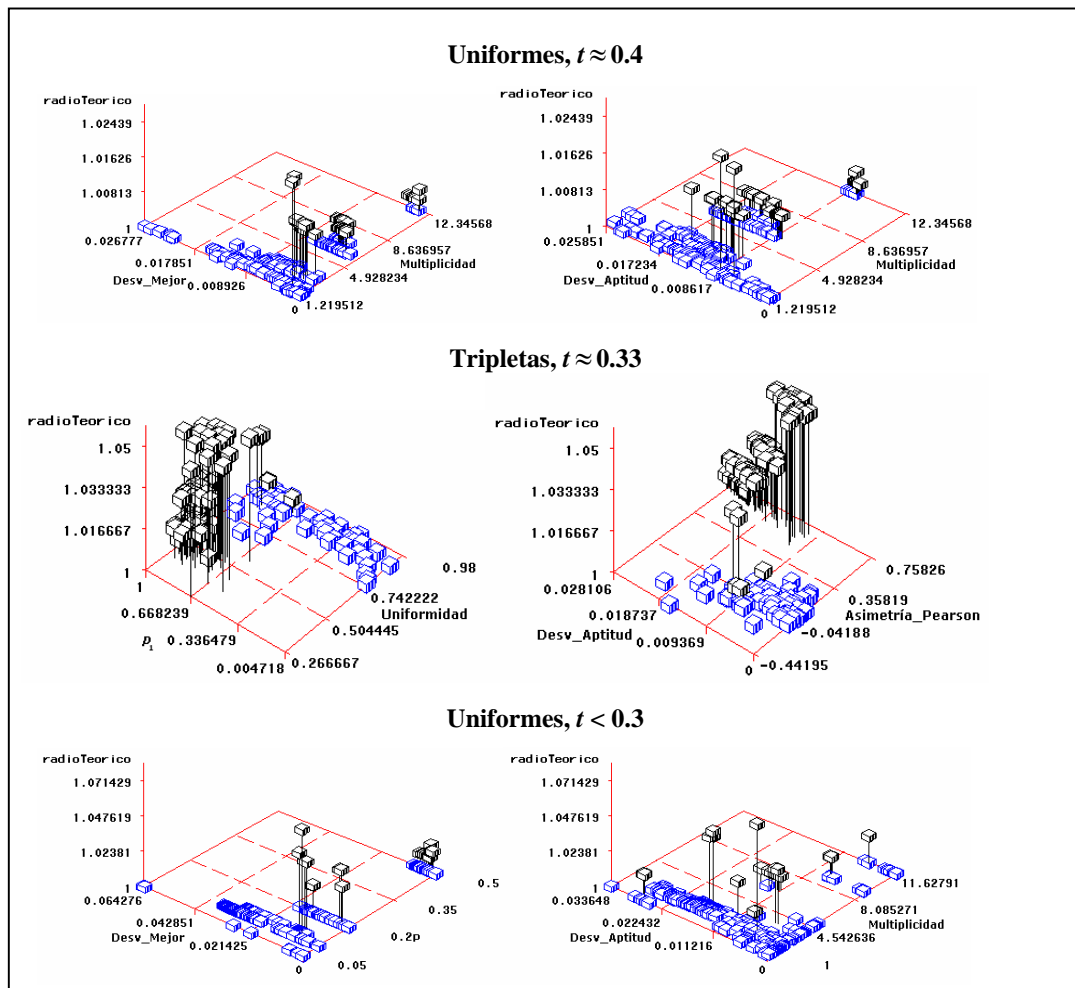


Figura 4.18 Gráficas de dispersión de índices de BPP y de comportamiento de HGGA-BP.

En la Sección 4.3.3 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 se pudo comprobar al analizar los valores de las métricas *Nuevos*, *Buenos* y *Malos* en la solución de instancias de diferentes características. La Figura 4.19 muestra una gráfica donde se observa el comportamiento del algoritmo en cada generación. Este comportamiento se presenta, en general, para todas las instancias de prueba.

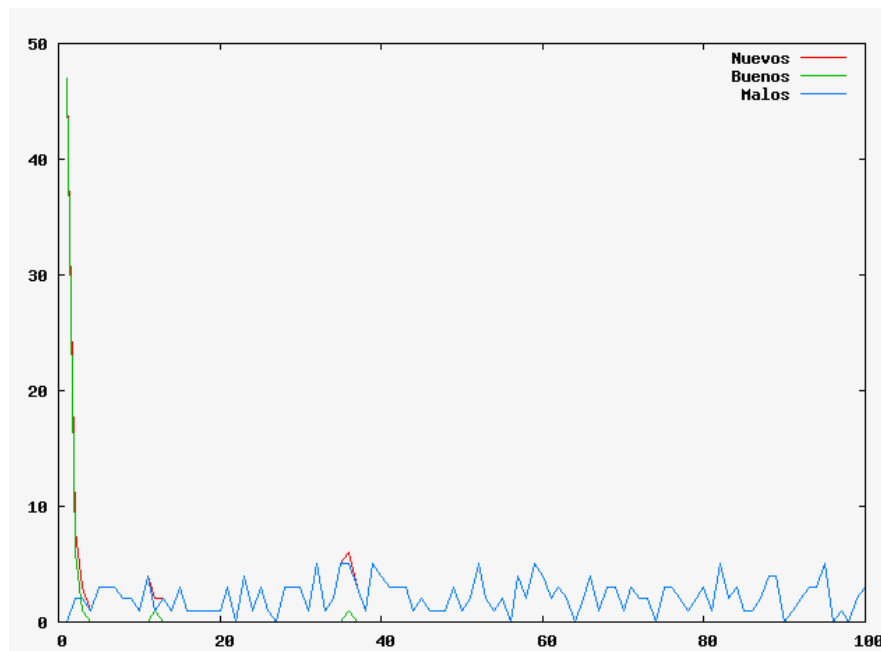


Figura 4.19 Individuos nuevos, agregados a la población en cada generación por HGGA-BP.

En la Figura 4.19 se observa que, la primera generación (donde se crea la población inicial) es la única en que se agrega diversificación 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 los métodos de generación de individuos no parecen crear individuos con aptitudes variadas, que lleven al algoritmo a explorar diversas regiones del espacio de búsqueda. Los pocos individuos que se aceptan en cada iteración cayeron en el 5% de probabilidad de ser aceptados sin tomar en cuenta su aptitud (ver Algoritmo 4.1).

El estudio de las relaciones entre los índices de caracterización demostró que las características que impactan en mayor 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 la multiplicidad del conjunto de pesos los que definen la complejidad. El análisis del comportamiento algorítmico evidenció que las estrategias incluidas en el algoritmo 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 parecen 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 4.30 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.

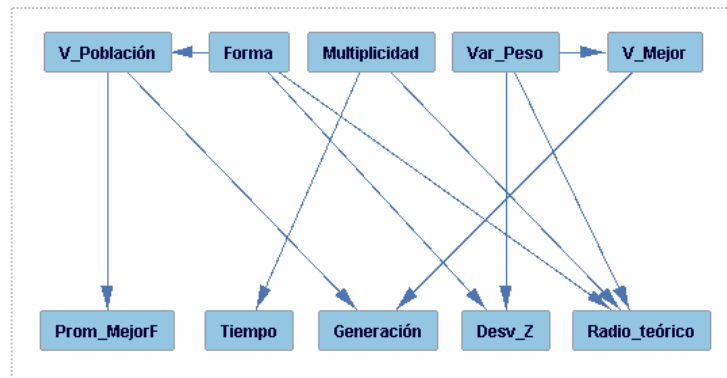


Figura 4.20 Diagrama causal del proceso de optimización de BPP con el algoritmo HGGA-BP.

Capítulo 5

REDISEÑO DEL ALGORITMO HGGA-BP

En este capítulo, el conocimiento obtenido del análisis experimental del proceso de optimización del algoritmo HGGA-BP es utilizado para incrementar su efectividad. Las principales estrategias del metaheurístico son examinadas, identificando áreas de mejora. La estructura del algoritmo es rediseñada, obteniendo como producto final una nueva versión del algoritmo que muestra una contundente mejora en el desempeño, superando la versión anterior. Los resultados obtenidos muestran la validez y aplicabilidad de la metodología para el análisis experimental de algoritmos propuesta en este trabajo.

5.1 ESTRUCTURA Y DESEMPEÑO ORIGINAL DEL ALGORITMO

El algoritmo original fue desarrollado en lenguaje C y todos los cambios fueron hechos en este mismo lenguaje. Las experimentaciones fueron realizadas en una computadora personal con procesador Pentium D a 3 Ghz y 2GB de RAM. En cada experimento se realizaron 30 corridas del algoritmo, los valores reportados como soluciones finales son los promedios de las 30 soluciones generadas para cada instancia.

Las estrategias y parámetros iniciales, utilizados en el algoritmo HGGA-BP [Nieto07], así como el desempeño mostrado se presentan a continuación. Los mejores resultados obtenidos por el autor son mostrados en la tabla 5.1.

Tabla 5.1 Resultados experimentales para el algoritmo HGGA-BP.

Instancias	No. Instancias	Instancias Óptimas	Tiempo	Generación Promedio
Data Set 1	720	690	4.75	45
Data Set 2	480	452	1.64	12
Data Set 3	10	8	9.33	88
T	80	0	7.55	69
U	80	53	14.56	48
Gau 1	17	9	1.15	53
Was 1	100	99	0.22	7
Was 2	100	99	1.14	29
Hard28	28	0	4.60	92
53NIRUP	53	5	7.05	83
Total	1668	1415	5.19	53

Dichos resultados son obtenidos con la configuración siguiente (las heurísticas para BPP y la descripción general del algoritmo fueron presentados en las Secciones 2.3.1 y 4.2.2):

- El tamaño de la población es de 50 individuos.
- Antes de iniciar el proceso evolutivo se generan tres soluciones deterministas con las heurísticas *FFD*, *LB+BFD* y *LB+B3FD*. En caso de encontrar la solución con número de contenedores igual al óptimo teórico (L_1) el algoritmo termina.
- Si es necesario entrar al proceso evolutivo el resto de la población se compone de: un individuo generado con *LB+B3F_Aleatorio* y 46 individuos generados con *LB+BF_Aleatorio*.
- Conservar individuos de la generación previa si son mejores con respecto a los nuevos individuos, tomando una probabilidad del 95% de aceptar dicha condición.
- Cruzar 30% de los individuos de la población.
- Mutar el 50% de los individuos de la población.
- Como criterio de paro: una solución con L_1 contenedores ó 100 generaciones.

En el capítulo anterior se mostró que esta configuración y estructura no parecen ser las más adecuadas para solucionar problemas de BPP con objetos medianos y pequeños.

5.2 REDISEÑO DE LA ESTRUCTURA DEL ALGORITMO

En el capítulo anterior, las relaciones de desempeño obtenidas del análisis experimental del algoritmo HGGA-BP evidenciaron que las estrategias incluidas, para creación de soluciones y exploración del espacio de búsqueda, presentaban ciertas inconsistencias (ver Sección 4.4). En esta sección se muestran las conclusiones obtenidas del análisis de las principales estrategias que definen la estructura y comportamiento del algoritmo y se presenta el rediseño de los procedimientos que tienen mayor impacto en el proceso de optimización.

5.2.1 GENERACIÓN DE INDIVIDUOS Y CRITERIO DE PARO

Los métodos construcción de individuos de HGGA-BP (detallados en la Sección 4.2.2) consisten en calcular el tamaño inicial de la solución ($LB = L_1$), acomodar los LB objetos más pesados en diferentes contenedores y almacenar el resto de los objetos con las heurísticas BF y DB3 (descritas en la Sección 2.2.1).

Al analizar el comportamiento del algoritmo, así como la aportación de los métodos para generación de individuos se observó que los dos métodos utilizados para generar individuos ($LB+BF_Aleatorio$ y $LB+B3F_Aleatorio$) limitaban el espacio de soluciones a cierta zona, al establecer desde el principio que algunos objetos ($LB = L_1$) debían estar 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 era posible encontrar la solución óptima.

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 sus objetos eran en realidad “pequeños” en relación a la capacidad del contenedor y algunos objetos podrían estar mezclados entre sí en la solución óptima.

Para identificar las mejores estrategias de generación de individuos se hizo un estudio de técnicas de acomodo de objetos propuestas en la literatura (Sección 2.3.1). El análisis de las principales heurísticas para BPP culminó con la generación de una nueva métrica que ayuda a cuantificar que tan complicada es una instancia del problema para una estrategia heurística. Dicha métrica es calculada tomando el porcentaje del número de veces que se obtuvo la solución óptima al resolver permutaciones aleatorias del conjunto de pesos de la instancia de interés. Los cambios realizados en el algoritmo fueron los siguientes:

1. El valor del límite inicial de contenedores para los métodos de generación de individuos aleatorios LB , anteriormente $LB = L_1$, fue cambiado por el valor de un nuevo límite, $LB = LM$, que es igual al número de objetos de tamaño mayor que la mitad de la capacidad del contenedor. LM representa los objetos realmente grandes y que no podrían combinarse entre sí.
2. La población inicial incluirá dos tipos de individuos, deterministas y aleatorios. La parte determinista incluye: 1 individuo creado con FFD , 1 individuo creado con BFD , 1 individuo creado con WFD , 1 individuo creado con $B3FD$, 1 individuo creado con $LB+FFD$, 1 individuo creado con $LB+BFD$ y 1 individuo creado $LB+WFD$. La población aleatoria se compone de: 1 individuo creado con $LB+B3F_Aleatorio$, $2/5$ creados con $LB+FF_Aleatorio$, $2/5$ creados con $LB+BF_Aleatorio$ y $1/5$ creado con $LB+WF_Aleatorio$.
3. El tamaño de la población se incrementó, con el objetivo de incluir una mayor variedad de grupo de objetos, ahora la población consta de 150 individuos.
4. el criterio de paro ahora utiliza el límite L_2 que es más exacto que el límite L_1 .

5.2.2 SUSTITUCIÓN Y CONSERVACIÓN DE INDIVIDUOS

En el algoritmo original, en cada generación 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 un el valor de un número aleatorio $U[0,1]$ sea mayor que 0.95.

El análisis del comportamiento algorítmico reveló que la generación de 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 el resto de los individuos generados una pérdida de tiempo.

Por otra parte, el estudio de las relaciones entre las aptitudes de los individuos de la población en cada generación mostró que después de las primeras 5 generaciones comenzaba a haber muchas repeticiones de individuos en la población, lo cuál conducía al algoritmo a estancarse en óptimos locales.

Los cambios propuestos fueron:

1. La población sólo será creada en la primera generación, los únicos cambios que pueden ocurrir en ella en generaciones subsecuentes serán debido a los operadores de cruzamiento y mutación.
2. Al final de cada generación, los individuos repetidos serán sustituidos por nuevos individuos, generados uno con el método *LB+B3F_Aleatorio* y el resto con la estrategia *LB+FF_Aleatorio*.

5.2.3 REACOMODO DE OBJETOS LIBRES

En el proceso de solución de HGGA-BP, las estrategias de cruzamiento y mutación generan y modifican individuos, durante este proceso algunos contenedores son vaciados y existen objetos libres que necesitan ser reinsertados en la solución. En el cruzamiento

por agrupación los objetos libres son reacomodados con la estrategia determinista *BFD*. En la mutación por agrupación también se utiliza este método, sin embargo, antes de llamarlo, se intenta mejorar el llenado de los contenedores que ya pertenecen a la solución. Lo anterior se realiza recorriendo los contenedores, y para cada contenedor, se intenta sustituir, cada uno de los objetos contenidos en él, por un objeto libre de mayor tamaño, esto cuidando de no sobrepasar la capacidad del contenedor. Para los objetos libres que no pudieron ser reacomodados se utiliza la estrategia *FFD*.

En este trabajo se propone agregar un nuevo procedimiento de acomodo. La estrategia propuesta ha sido llamada *Reacomodo_por_Pares* y consiste principalmente en:

1. Generar una permutación aleatoria del orden de los contenedores
2. Recorren por pares todos los objetos de cada contenedor intentando:
 1. Sustituir el par de objetos del contenedor por un objeto libre de peso igual o mayor que no sobrepase la capacidad del contenedor.
 2. Sustituir el par de objetos del contenedor por un par de objetos diferentes que sumen lo mismo o que llenen más el contenedor.

5.2.4 CRUZAMIENTO

El algoritmo GGA-BP incluye el cruzamiento por agrupación propuesto por Falkenauer [Falkenauer92]. Cruzar individuos implica que contenedores sean heredados de padres a hijos. Aquellos contenedores que sobreviven el proceso evolutivo se caracterizan por ser dominantes sobre otros. En esta técnica, se crean, de manera aleatoria, dos puntos de corte en cada individuo a cruzar, que dividen cada solución en tres segmentos. El nuevo individuo (hijo) está formado el 1er segmento de contenedores del primer padre, el 2do segmento de contenedores del segundo padre y el resto de contenedores del primer padre, eliminándose contenedores con objetos duplicados y reacomodando los objetos libres con *FFD*.

El análisis del comportamiento algorítmico mostró que aumentando el porcentaje de individuos a cruzar el algoritmo converge a mejores resultados, sin significar este aumento un incremento en el tiempo de ejecución, debido a que en muchos de los casos, los resultados óptimos son encontrados en generaciones tempranas. Debido a lo anterior el porcentaje de individuos de la población a cruzar fue incrementado a 50%.

Así mismo se incluyó un nuevo método de cruzamiento para explotación de buenas soluciones. En esta nueva estrategia, los individuos a cruzar son seleccionados de manera aleatoria tomando en cuenta sólo soluciones de calidad (la mejor mitad de la población) y los objetos libres son reacomodados utilizando el esquema propuesto en este trabajo *Reacomodo_por_Pares*. El nuevo cruzamiento es aplicado para generar 10 nuevos individuos en cada generación.

5.2.5 MUTACIÓN

El operador de mutación por agrupación del algoritmo HGGA-BP original, elimina un porcentaje de los contenedores menos llenos y utiliza una estrategia ambiciosa de reacomodo que intercambia objetos libres por objetos de cada contenedor. Antes de entrar a este método, los individuos son ordenados según aptitud. Los individuos que mutan son los que poseen la menor aptitud (peores soluciones). Para cada individuo a mutar, el número de contenedores a vaciar (para volver a acomodar sus objetos) es igual al 10% del total de los contenedores de la solución.

Al analizar las soluciones creadas por el algoritmo se observó que el porcentaje de contenedores a vaciar (originalmente 10% del número de contenedores de la solución) debería depender del número de contenedores libres. Esto se concluyó al observar que para algunas soluciones, vaciar sólo el 10% de los contenedores de la solución era muy poco, ya que muchos contenedores estaban incompletos, mientras que para otros, era una gran cantidad, pues siempre se vaciaban contenedores que ya estaban completos.

El estudio formal de las relaciones entre los factores que afectan el proceso de optimización, mostró que mutar soluciones malas no es suficiente para obtener buenas soluciones, por lo que se decidió que individuos buenos también debían ser mutados, para dichos individuos se creó un nuevo método. La mutación para individuos con buenas aptitudes consiste principalmente en: elegir el porcentaje de contenedores a vaciar de acuerdo con el tamaño de la solución y el número de contenedores y reacomodar los objetos libres con estrategia *Reacomodo_por_Pares*.

5.3 RESULTADOS FINALES

La aplicación del conocimiento obtenido en el análisis de las características del proceso de optimización de BPP mediante el algoritmo GGA-BP, permitió obtener una contundente mejora en el desempeño del algoritmo. Los resultados finales son mostrados en la Tabla 5.2, donde se incluyen el número de óptimos, tiempo promedio (en segundos) y generaciones que el algoritmo iteró. Se presenta, además, el número de óptimos obtenidos por el algoritmo HI-BP [Alvim04], que es en la actualidad el mejor algoritmo del estado del arte.

Tabla 5.2 Resultados mejorados para el algoritmo GGA-BP.

Instancias	No. Instancias	HI-BP	HGGA-BP Original			HGGA-BP Mejorado		
			Óptimos	Tiempo	Gen	Óptimos	Tiempo	Gen
Data Set 1	720	720	690	4.75	45	718	4.72	19
Data Set 2	480	480	452	1.64	12	480	0.83	6
Data Set 3	10	10	8	9.33	88	9	8.90	77
T	80	80	0	7.55	69	72	5.11	56
U	80	80	53	14.56	48	79	3.08	13
GAU1	17	12	9	1.15	53	15	2.24	42
WAE1	100	100	99	0.22	7	100	0.06	3
WAE2	100	100	99	1.14	29	100	1.61	27
Hard28	28	5	0	4.60	92	6	6.55	91
NIRUP	53	3	5	7.05	83	3	4.55	82
Total	1668	1590	1415	5.19	53	1582	3.76	42
Efectividad		0.953	0.848			0.948		

Capítulo 6

CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se presentan conclusiones y resultados finales obtenidos en este trabajo y se dan sugerencias para el desarrollo de trabajos futuros.

6.1 CONCLUSIONES

Entre las principales aportaciones de este trabajo se encuentran:

1. El desarrollo de una **metodología experimental para el análisis de desempeño de algoritmos metaheurísticos** que permite comprender el comportamiento del algoritmo y mejorar su desempeño.
2. Se analizó y propuso un conjunto de índices de caracterización para el problema de empacado de objetos en contenedores.
3. Se mejoró el desempeño del algoritmo HGGA-BP, mejorando su efectividad de un 84.83% a un 94.84% para el conjunto de 1668 instancias de prueba.
4. La nueva versión del algoritmo HGGA-BP permite solucionar instancias que el mejor algoritmo del estado del arte, HI-BP, no logra solucionar.

Los resultados obtenidos por la nueva versión del algoritmo muestran la validez y aplicabilidad de la metodología propuesta.

6.2 TRABAJO FUTURO

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

- a. 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.
- b. Ampliar la metodología propuesta agregando técnicas estadísticas de análisis confirmatorio.
- c. Aplicar la metodología a otra clase de problemas, por ejemplo estudiar las relaciones entre propiedades de grafos y problemas de grafos.

REFERENCIAS BIBLIOGRÁFICAS

- [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 Aplicação 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. <http://people.brunel.ac.uk/~mastjbjeb/orlib/binpackinfo.html>
- [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.

- [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.
- [Brglez07] Brglez F., Osborne J. A. *Performance testing of combinatorial solvers with isomorph class instances*. Workshop On Experimental Computer Science, Article No. 13, 2007.
- [CaPaD] Cutting and Packing at Dresden University.
<http://www.math.tu-dresden.de/~capad/cpd-ti.html#pmp>
- [Carnegie09] Carnegie Mellon University. *Causal and Statistical Reasoning*. Open Learning Initiative, 2009. <http://oli.web.cmu.edu/openlearning/>
- [Chevalier06] Chevalier R. *Balancing the effects of parameter settings on a genetic algorithm for multiple fault diagnosis*. Artificial Intelligence, The university of Georgia, 2006.
- [Cohen95] Cohen P. R. *Empirical Methods for Artificial Intelligence*. The MIT Press, Cambridge, Massachusetts, 1995.
- [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.
- [Díaz96] Díaz A., González J., Laguna M., Mascato P., Tseng T., Glover F., Ghaziri M. *Optimización Heurística y Redes Neuronales*. Editorial Parainfo, España, 1996.
- [Dorigo92] Dorigo M. *Optimization, Learning and Natural Algorithms*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Itali, 1992.
- [Dréo06] Dréo J., Pétrowski A., Siarry P., Taillard E. *Metaheuristics for Hard Optimization Methods and Case Studies*. Springer-Verlang Berling Heidelberg, 2006.
- [Drezner05] Drezner Z., Hann P., Taillard E. *Recent Advances for the Quadratic Assignment Problem with Special Emphasis on Instances That Are Difficult for Meta-heuristic Methods*. Annals of Operations Research, 139: 65-94, 2005.

- [Ducatelle01] Ducatelle F. *Ant Colony Optimisation for Bin Packing and Cutting Stock Problems*. School of Artificial Intelligence, Division of Informatics University of Edinburgh, 2001.
- [ESICUP] Euro Especial Interest Group on Cutting and Packing. *One Dimensional Cutting and Packing Data Sets*. Disponible en: http://paginas.fe.up.pt/~esicup/tiki-list_file_gallery.php?galleryId=1
- [Falkenauer92] 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.
- [Fayyad96] Fayyad U., Piatetsky G., Smyth P., Uthurusamy R. *Advances in Knowledge discovery and data mining*. Ed. AAAI Press/The MIT Press, California, 1996.
- [Garey79] Garey M. R., Jonson D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979. A classic introduction to the field.
- [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.
- [Glover86] Glover F. *Future paths for integer programming and links to artificial intelligence*. Computers and Operations Research, 13: 533-549, 1986.
- [Gonzalez07] Gonzalez T. *Handbook of Approximation Algorithms and Metaheuristics*. Chapman & Hall/CRC, Taylor & Francis Group, 2007.
- [Hartwig79] Hartwig F., Dearing B. E. *Exploratory Data Analysis*. Sage University Paper Series on Cuantitative Research Methods, Vol. 16, Newbury Park, CA: Sage, 1979.
- [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.
- [Johnson00] Johnson D. *Método Multivariados Aplicados al Análisis de Datos*. Primera Edición. International Thomson Editores, 2000.
- [Kadie01] Kadie C., Hovel D., Horvitz E. *MSBNx: a component-centric toolkit for modelling and inference with bayesian networks*. Rep. Téc. MST-TR-2001-67, Redmon, WA: Microsoft, 2001.
- [Kalisch07] Kalisch M., Bühlmann P. *Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm*. Journal of Machine Learning Research, 8: 613-636, 2007.
- [Kaski97] Kaski S. *Data exploration using self-organizing maps*. Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82, Espoo 1997.
- [Klein09] Klein R., Scholl A. *Bin Packing*. Disponible en: <http://www.wiwi.uni-jena.de/Entscheidung/binpp/>
- [Konak05] Konak A., Kulturel S. *Simulation optimization using Tabu search: An empirical study*. Proceedings of Winter simulation Conference, M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds, 2005.
- [Korb04] Korb K., Nicholson A. *Bayesian Artificial Intelligence*. Chapman and Hall, London, UK, 2004.
- [Kos02] Kos L., Duhovnik J. *Cutting optimisation with variable-sized stock and inventory status data*. International Journal of Production Research, 40(10): 2289-2301, 2002.
- [Lander08] Lander V. *Desarrollo de un Método Formal que Muestra 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.

- [Lemeire05] Lemeire J., Dirkx E. *Causal Models for Parallel Performance Analysis*. 4th PA3CT Symposium, Edegem, Belgium, 2004.
- [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.
- [Madsen05] Madsen A., Jensen F., Kjaerulff U., Lang M. *The Hugin Tool for Probabilistic Graphical Models*. International Journal on Artificial Intelligence Tools, 14(3): 507-544, 2005.
- [Liu96] Liu X. *Intelligent Data Analysis: Issues and Challenges*. The Knowledge Engineering Review, 11: 365-371, 1996.
- [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.
- [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.
- [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.
- [Merz00] Merz P., Freisleben B. *Greedy and Local Search Heuristics for Unconstrained Binary Quadratic Programming*. Tech. Rep., Dept. of Computer Science, University of Siegen, Germany, 2000.
- [Michalewicz04] Michalewicz Z., Fogel D. B. *How to Solve It: Modern Heuristics*. 2nd edition, Springer-Verlang, 2004.

- [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.
- [Nadkarni01] Nadkarni S., Shenoy P. *A bayesian network approach to making inferences in causal maps*. European Journal of Operational Research, 128: 479-498, 2001.
- [Neapolitan04] Neapolitan R. *Learning Bayesian Networks*. Pearson Prentice Hall, 2004.
- [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.
- [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.
- [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.

- [Tukey77] Tukey J. W. *Exploratory Data Analysis*. Addison-Wesley, 1977.
- [Vila04] Vila M., Sánchez D., Escobar L. *Relaciones Causales en Reglas de Asociación*. XII Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF'2004), 2004.
- [Wäscher96] Wäscher G., Gau T. *Heuristics for the one-dimensional cutting stock problem: A computational study*. OR Spektrum, 18: 131-144, 1996.
- [Weinberg90] Weinberg E. D. *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.