

---

# Un Algoritmo GRASP con dos Parámetros de Relajación para el problema 3D-BIN Packing con Restricciones de Estabilidad

---

David Mauricio<sup>1</sup>, Ana Rojas<sup>2</sup>, Gonzalo Calderón<sup>3</sup>

Universidad Nacional Mayor de San Marcos,  
Facultad de Ingeniería de Sistemas e Informática  
Av. Germán Amézaga s/n, Ciudad Universitaria, Lima 01, Lima, Perú

<sup>1</sup>dms\_research@yahoo.com, <sup>2</sup>ana\_rojasrimarachin@yahoo.es, <sup>3</sup>gonzalo\_calderonpinto@yahoo.es

---

## RESUMEN

Dado un conjunto de paquetes rectangulares y un número ilimitado de contenedores rectangulares de dimensiones mayores a los paquetes, el problema de Three-Dimensional Bin Packing Problem (3D-BPP) consiste en empaquetar todos los paquetes en los contenedores haciendo uso del menor número de contenedores. El problema es NP-Arduo y presenta aplicaciones en los procesos de embalajes de los diversos sectores de la industria y servicio.

El presente trabajo introduce un algoritmo Construction GRASP, el cual usa dos parámetros de relajación para resolver el problema de 3D-BPP con restricciones de estabilidad, esto es, el empaquetamiento debe ser realizado de forma que el conjunto de paquetes empaquetados no presente movimiento. Se establece una condición suficiente de optimalidad y un procedimiento de mejoría. Los resultados numéricos muestran superioridad en cuanto a la solución respecto al algoritmo goloso BFD.

**Palabras clave:** 3D Bin Packing Problem, GRASP, Estabilidad.

## ABSTRACT

Given a set of rectangular boxes and an unlimited number of rectangular containers of dimensions bigger than the boxes, the Three-dimensional Bin Packing Problem (3D-BPP) consists of packing all the boxes in the containers such that this one uses the minor number of containers. This problem is NP-hard and presents applications in the packing processes of the diverse sectors of the industry and service.

The present work introduces a GRASP Construction algorithm that uses two relaxation parameters to solve the problem of 3D-BPP with restrictions of stability, this is the packing must be realized so that the set of stiff packages does not present movement. A sufficient condition is established of optimality and a procedure of improvement. The numerical results show superiority as for the solution with regard to the greedy algorithm BFD.

**Keywords:** 3D Bin Packing Problem, GRASP, Stability.

---

## 1. INTRODUCCIÓN

### (3D-BPP):

Considere un número ilimitado de contenedores rectangulares con dimensiones de largo  $L$ , ancho  $A$  y altura  $H$ , y un conjunto de  $n$  paquetes rectangulares de ancho, altura y largo, respectivamente,

$$w_1 \times h_1 \times d_1, \\ w_2 \times h_2 \times d_2, \dots, w_n \times h_n \times d_n \text{ con } w_i \leq W, h_i \leq H$$

y  $d_i \leq D \forall i$ . El three-dimensional bin packing problem, consiste en empaquetar (colocar) los  $n$  paquetes en los contenedores con lados paralelos a las mismas de forma a usar el menor número de contenedores.

De acuerdo a la tipología de Dyckhoff (1990) este problema puede ser clasificado como 3/N/1/M.

En este trabajo es considerado, adicionalmente, la estabilidad estática del empaquetamiento, esto es, el empaquetamiento debe ser realizado de forma que el conjunto de paquetes empaquetados no presente movimiento. Además, se pueden hacer rotación de los paquetes.

La principal aplicación del 3D-BPP es referida a la reducción del espacio ocioso ocasionado durante los procesos de embalajes o en la gestión de contenedores. Entre estas aplicaciones setienen: cargas de piezas metálicas por Roberts y Taylor (1972), carga de rollos de papel de Haessler (1979), gestión de cajas no uniformes en un contenedor George and Robinson (1980), Han et al (1989), Morabito y Arenales (1993).

Garey e Jonhson (1979) demuestran que el problema de cortes de una dimensión (one-dimensional bin packing problem) es un problema NP-Arduo, y, siendo este un problema particular del problema en estudio (basta considerar la altura y el ancho de los paquetes y de los contenedores como siendo un valor igual para todos), podemos concluir que 3D-BPP es también un problema de clase NP-Arduo. Esto justifica el desarrollo de algoritmos aproximados para su resolución. A continuación, se mencionan algunos de estos métodos.

Pisinger et al. (1999) presenta una adaptación del método general de búsqueda local de proble-

mas combinatorios. Una solución basada en algoritmo genético fue introducida por Corcoran y Waunwriht (1992). Bischoff y Dowland (1982) proponen un método basado en el empaquetamiento de dos dimensiones. Gehring et al (1990) consideran restricciones de pesos en el problema 3D-BPP y proponen una heurística. Martello et al (2000) describen dos heurísticas golosas del tipo BFD (Best Fit Decreasing) y proponen un algoritmo exacto usando el método de branch and bound. Lodi et al (2002) también proponen una heurística golosa. Propuestas de algoritmos golosos FFD (on-line y out-line) también se encuentran en el trabajo de tesis de Flavio Miyasawa (1993). Li y Cheng (1992) diseñan un algoritmo de empaquetamiento en línea para 3D-BPP, que es basado en la reducción del problema de tres dimensiones a una secuencia de problemas de una dimensión. Miyasawa y Wakabayashi (1997) proponen una heurística que divide la lista de ítems en sub-listas, y aplica algoritmos aproximados para cada una de estas sub-listas; el empaquetamiento final es obtenido como una concatenación de las soluciones de las sub-listas; los resultados numéricos muestran que su propuesta es superior al algoritmo propuesto por Li y Cheng (1992). Un sistema experto que considera reglas y restricciones para cargar manualmente paquetes no idénticos en un contenedor es propuesto por Gilmore et al. (1989). Un algoritmo basado en el método de simulated annealing es introducido por Faina (2000). Un estudio de de algoritmos de búsqueda local para el problema general de bin packing es desarrollado por Osogami y Okano (2003).

Los diversos algoritmos desarrollados para el problema 3D-BPP no toman en cuenta la estabilidad del empaquetamiento. En este sentido, Castro et al (2003) desarrollan una heurística golosa basada en el algoritmo de Martello et al (2000), que consideran restricciones de estabilidad.

El procedimiento de búsqueda adaptativa aleatoria y golosa (GRASP) desarrollada por Feo y Resende (1995) está comprobado que sea altamente eficiente para resolver diversos problemas de grande porte de la optimización combinatoria. Los resultados numéricos de los algoritmos GRASP de Feo y Resende (1989) para los problemas de cobertura, máximo conjunto independiente Feo et al (1994), localización de facilidades indeseadas Cruz y

Mauricio (1997), optimización de cortes de barra Ganoza y Solano (2004), entre otros, son buenos indicadores para justificar el desarrollo de algoritmos GRASP para resolver el problema 3D-BPP.

GRASP es un proceso iterativo, cada iteración GRASP consiste de dos fases: fase de construcción y fase de búsqueda local, en cada iteración se deberá registrar la mejor solución encontrada. La iteración GRASP termina cuando algún criterio de parada, tal como un número máximo de iteraciones, haya ocurrido o alguna solución encontrada es satisfactoria.

La fase de construcción GRASP es similar a un proceso goloso, con la variación que el criterio goloso de selección del elemento a ser parte de la solución es relajado considerando un componente de aleatoriedad, lo que permite en cada corrida del algoritmo construir soluciones diferentes. La diversidad de soluciones es obtenida debido a la relajación del criterio goloso. En este sentido, en Mauricio et al (2004) se introducen algoritmos GRASP para el problema 3D-BPP pero con dos relajaciones.

El presente trabajo describe un algoritmo de construcción GRASP con dos parámetros de relajación para resolver 3D-BPP. La propuesta desarrollada considera el criterio goloso del algoritmo de Martello et al. (2000) y las condiciones de estabilidad propuesta por Castro et al. (2003).

El orden del trabajo es como sigue. En la siguiente sección se describe, sucintamente, el proceso de empaquetamiento. A fin de mejor comprensión de la propuesta, es descrito el algoritmo goloso BFD en la sección 3. En la sección 4, se presenta el algoritmo de construcción GRASP, se muestra una condición suficiente de optimalidad y un procedimiento de mejora (search local procedure). Resultados numéricos del algoritmo propuesto comparados con el algoritmo BFD son mostrados en la sección 5. Finalmente, las conclusiones siguen en la sección 6.

## 2. EL PROCESO DE EMPAQUETAMIENTO

### 2.1. Notación

En lo que sigue del presente trabajo se usará la

siguiente notación:

$B_i$ :  $i$ -ésimo contenedor (paralelepípedo rectangular de dimensiones ancho, altura y largo  $W \times H \times L$ ).

$P_i$ :  $i$ -ésimo paquete (paralelepípedo de dimensiones igual o inferior a las dimensiones del contenedor) con dimensiones  $w_i \times h_i \times d_i$ .

$V(P_i)$ : volumen del paquete  $P_i$ .

Sin pérdida de generalidad los contenedores son colocados en el primer octante del eje de coordenadas tridimensional, con uno de sus vértices de esquinas en el centro de coordenadas  $(0, 0, 0)$ , como se observa en la Figura 1. A los ejes de coordenadas X, Y, Z corresponden, respectivamente, las dimensiones de largo, ancho y altura.

### 2.2. Puntos de Encaje y Punto de Esquina

Se usará el término de punto de encaje de un paquete a su punto extremo más próximo del origen de coordenadas (observe que un paquete presenta 8 puntos extremos, y recuerde que los paquetes serán empaquetados con lados paralelos a los lados del contenedor), posición donde se fijara el paquete.

Para el proceso de encaje se ha considerado el concepto de punto de esquina introducido por Martello et al (2000). Los puntos esquinas son puntos espaciales dentro de un contenedor que son candidatos a ser un punto de encaje para un paquete. En siguiente Figura 1.B se muestra un contenedor con 3 paquetes y 7 puntos de esquinas. El punto de esquina de un contenedor vacío es dado por las coordenadas  $(0, 0, 0)$ .

En lo que sigue del presente trabajo se considera la siguiente notación:

$(x_j, y_j, z_j)$ : Punto de encaje de  $P_j$ .

$J_i$ : Conjunto de índices de los paquetes empaquetados en  $B_i$ .

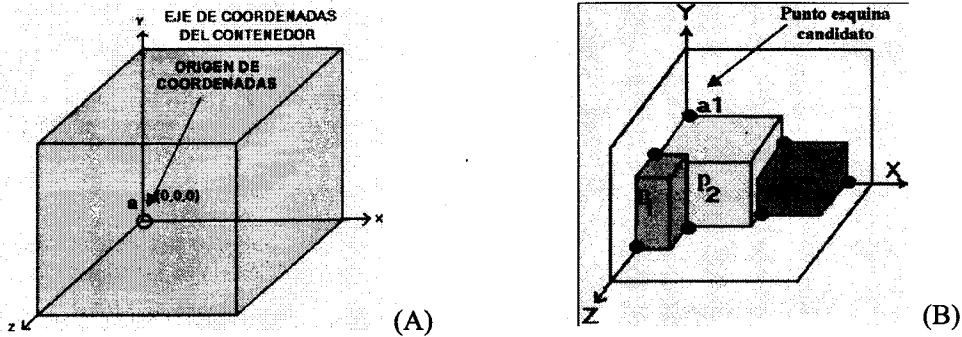


Figura 1: A) Un contenedor en el eje de coordenadas tri-dimensional.  
 B) Puntos de esquina de un contenedor con tres paquetes

$E_i$ : Conjunto de puntos de esquinas de  $B_i$ .

Las condiciones para un encaje válido de un paquete  $P_j$  en el punto de esquina  $(x, y, z)$  de un contenedor  $B_i$ , son dadas por las siguientes relaciones:

$$x + w_j \leq W, \quad y + h_j \leq H, \quad z + d_j \leq D \quad (1)$$

No existe  $q \in J_i$  tales que:

$$z + d_j > z_q, \quad x_q \leq x \leq x_q + w_q, \quad y < y_q \quad \text{o} \quad (2a)$$

$$x + w_j > x_q, \quad z_q \leq z \leq z_q + d_q, \quad y < y_q \quad (2b)$$

La condición dada en la relación (1) establece

que el encaje debe ser realizado de forma que el paquete encaje totalmente en el contenedor. Las relaciones 2a y 2b establecen las condiciones de no traslape.

### 2.3. Estabilidad

La segunda consideración para el proceso de empaquetamiento es la estabilidad estática, lo cual significa que el empaquetamiento debe ser realizado de forma que el conjunto de paquetes empaquetados no presente movimiento. Se han considerado las condiciones de estabilidad de Castro et al (2003), las cuales se resumen en: paquetes situados en la base del contenedor, el centro de gravedad del nuevo empaquetamiento se debe encontrar entre la superficie de algún otro paquete (Figura 2), y el centro de gravedad de nuevo empaquetamiento debe encontrarse en una línea cuyo extremos se encuentre sobre la superficie de otros dos bloques (Figura 3).

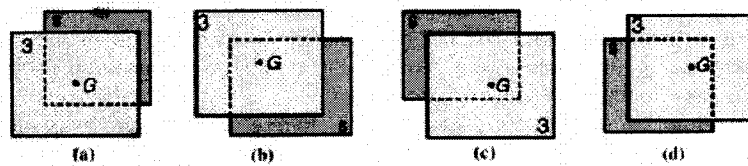


Figura 2: Centro de Gravedad (G), vista desde arriba

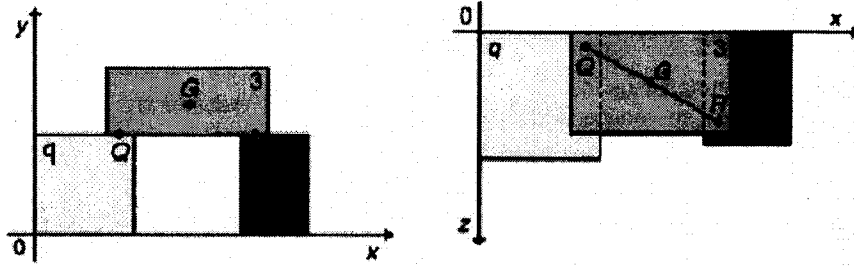


Figura 3: Estabilidad de un empaquetamiento, vistas laterales

### 2.4. Estrategia de Rotación

En algunas situaciones es más adecuado rotar el paquete antes de ser empaquetado. La rotación de un paquete puede ser concebido como el trueque de sus dimensiones. Así por ejemplo, un paquete de dimensiones  $w \times h \times d$  que es rotado  $90^\circ$  respecto al eje  $x$  (ancho), corresponde a un paquete de dimensiones  $h \times w \times d$ . Una estrategia de rotación consiste en rotar (en algunos de los ejes) siempre que sea posible encajar el mayor número de paquetes de igual dimensión en el contenedor. Esta estrategia obedece a problemas donde existen paquetes de igual dimensión.

### 2.5. Encaje en Contenedor Nuevo

Considere un paquete  $P_j$  que se desea empaquetar en un contenedor no usado  $B_i$ . El procedimiento de encaje consiste en tres pasos. Primero, se aplica la estrategia de rotación sobre el paquete con el fin de determinar la mejor posición para este. Segundo, se establece como punto de encaje del paquete  $P_j$  (rotado si fuera el caso) siendo el punto de esquina  $(0, 0, 0)$ . Tercero, se determinan los puntos de esquina del contenedor  $B_i$ , el cual se denota por  $E_i$  y se actualiza  $J_i$  como  $J_i = \{j\}$ .

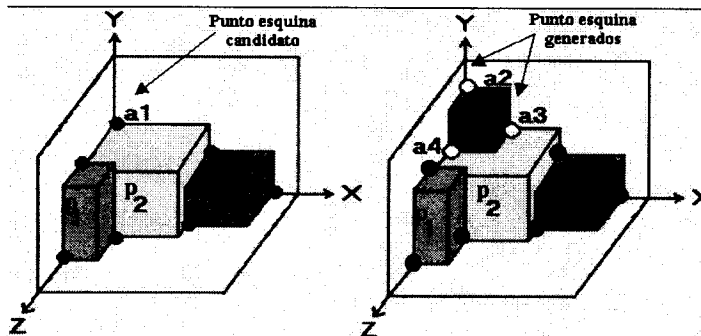


Figura 4: Encaje en un contenedor usado.

## 2.6. Encaje en Contenedor Usado

Considere un paquete  $P_j$  que se desea empaquetar en un contenedor usado  $B_i$ , y que esto es posible (esto es, se verifica la condición de estabilidad y de encaje para alguna rotación  $P_j$ ). El procedimiento de encaje consiste en tres pasos. Primero, se selecciona un punto de esquina  $(x, y, z)$  desde  $E_i$  de forma para que sea punto de encaje de  $P_j$  (rotado si fuera el caso). El punto seleccionado debe verificar la condición de encaje, estabilidad, y debe originar menor espacio no usado. Segundo, se asigna como punto de encaje de  $P_j$ , el punto de esquina seleccionado. Tercero, se debe actualizar el conjunto de paquetes asignados a  $B_i$  como  $J_i := J_i \cup \{j\}$ , se determina los nuevos puntos esquinas y se actualiza el conjunto de puntos esquinas  $E_i$ .

## 3. ALGORITMO BFD

El algoritmo BFD (Best Fit Decreasing) selecciona el contenedor que presenta menor espacio no utilizado con respecto al paquete a empaquetar. El espacio no utilizado es determinado como la diferencia de volúmenes entre el volumen del contenedor y el volumen de los paquetes colocados en dicho contenedor. Denotemos el espacio no utilizado de un contenedor  $B_i$  por  $R_i$ . A continuación, sigue el algoritmo BFD:

- El orden de prioridad para empaquetar los paquetes es dado por la ordenación no creciente de sus volúmenes (paso 2).
- La expresión  $B_k \nexists P_j$  significa que existe un punto de esquina en  $B_k$  donde se puede encajar  $P_j$ , y se cumple las condiciones de estabilidad estática.
- En el paso 4.1 se determina el contenedor que presenta menor espacio no utilizado donde se puede encajar el paquete  $P_j$  y se cumple la condición de estabilidad. Observe

### BFD Algorithm

1. InputInstance  $(n, P_1, P_2, \dots, P_n, L, A, H)$ ;
2. Sort  $(P_i)$  such that:  
 $V(P_1) \geq V(P_2) \geq \dots \geq V(P_n)$
3.  $m := 0$ ;
4. For  $j := 1, \dots, n$ 
  - 4.1  $i := \underset{1 \leq k \leq m+1}{\text{ArgMin}} \{R_k : B_k \nexists P_j\}$
  - 4.2 If  $i = m+1$
  - 4.3 Then Encaje\_Contenedor\_Nuevo  $(P_j, B_i), m := m+1,$
  - 4.4 Else Encaje\_Contenedor\_Usado  $(P_j, B_i),$
5. Return  $(m, B_i \forall i)$

Figura 5: Algoritmo BFD para 3D-BPP.

que, si  $i = m+1$  entonces el contenedor seleccionado es nuevo (paso 4.3), caso contrario el contenedor seleccionado será uno usado (paso 4.4).  $m$  indica el número de contenedores usados.

La heurística propuesta de Castro et al. (2003) es un algoritmo que en cada iteración (realiza  $n$  iteraciones, una por paquete) ejecuta una variación del algoritmo BFD. La variación consiste en fijar como primer paquete a empaquetar uno de los " $n$ " paquetes, esto se hace luego que se determina la secuencia de los paquetes a empaquetar (paso 2). Esta propuesta permite determinar " $n$ " soluciones para el problema en estudio.

## 4. ALGORITMO GRASP

### 4.1. Construcción GRASP

#### Criterio de Selección del Paquete

En lo que sigue se denotará por  $J$  el conjunto de los índices de los paquetes por empaquetar. Observe que al inicio del problema,  $J$  corresponde a todos los paquetes, esto es  $J = \{1, 2, \dots, n\}$ .

El algoritmo de construcción GRASP y el algoritmo goloso se diferencian principalmente en el proceso de construcción de la solución. El

algoritmo goloso construye una solución usando un criterio de selección goloso-miope. Para el caso del problema 3D-BPP, el algoritmo BFD seleccionan el paquete que presenta mayor volumen (paso 2 del algoritmo BFD). Entretanto, el algoritmo de construcción GRASP construye una solución usando un criterio de selección golosa – aleatoria, el cual consiste de dos pasos. Primero, se construye un conjunto de elementos candidatos a ser parte de la solución, luego se realiza un llamado de lista de candidatos restringidos ( $RCL_J$ ), esto se hace mediante una relajación del criterio goloso. Segundo, el elemento seleccionado para ser parte de la solución es obtenido desde  $RCL_J$  a través de una selección aleatoria de sus elementos.

**Criterio Goloso de Selección del Paquete**

$$V(P_j) := \text{Max}\{V(P_i) : i \in J\}$$

**Criterio GRASP de Selección del Paquete**

1.  $\bar{V} := \text{Max}\{V(P_i) : i \in J\};$
2.  $\underline{V} := \text{Min}\{V(P_i) : i \in J\};$
3.  $RCL_J := \{i \in J : \bar{V} - \alpha(\bar{V} - \underline{V}) \leq V(P_i) \leq \bar{V}\};$
4.  $j := \text{Random}(RCL_J);$

Figura 6: Criterio Goloso y criterio GRASP de selección del requerimiento.

El parámetro  $\alpha$  es llamado parámetro de relajación, cuyo valor varía entre 0 y 1. Se observa que, cuando  $\alpha = 0$  el criterio GRASP de selección de paquete GRASP se toma exactamente igual al criterio goloso; y cuando  $\alpha = 1$ , este se toma un criterio de selección totalmente aleatorio. Esto significa que la calidad de la solución dependerá del parámetro de relajación. Cuando se use valores para  $\alpha$  próximo a cero se deberá obtener soluciones próximas a la solución golosa; entretanto, para valores próximos a uno, se deberá obtener soluciones muy diversas posiblemente indeseables.

**Criterio de Selección del Contenedor**

Sea  $P_j$  el paquete a ser empaquetado, entonces los criterio de selección del contenedor del algoritmo BFD consiste en seleccionar el contenedor en donde se pueda encajar  $P_j$  que cumpla las condiciones de estabilidad y menor espacio ocioso. En la Figura 7 se formaliza este criterio. Observe que, si el valor de "i" es "m+1", entonces el contenedor seleccionado es nuevo (en el sentido que no es usado).

**Criterio BFD de Selección del contenedor**

$$i := \text{ArgMin}_{1 \leq k \leq m+1} \{R_k : B_k \supseteq P_j\};$$

**Criterio GRASP de Selección del Contenedor**

1.  $\underline{R} := \text{Min}_{1 \leq k \leq m+1} \{R_k : B_k \supseteq P_j\};$
2.  $RCL_B := \{k \in \{1, \dots, m+1\} : B_k \supseteq P_j, \underline{R} \leq R_k \leq \underline{R} + \alpha(LAH - \underline{R})\};$
3.  $i := \text{Random}(RCL_B);$

Figura 7: Criterio BFD y GRASP de selección del Contenedor.

Para la construcción del algoritmo de construcción GRASP con dos parámetros de relajación se relajará el criterio BFD de selección del contenedor a usar. Primero, se construye un conjunto de contenedores llamado de listas de contenedores candidatos restringidas ( $RCL_B$ ), esto se hace mediante una relajación del criterio BFD. Segundo, el contenedor seleccionado se obtendrá a través de una selección aleatoria desde  $RCL_B$ .

**Algoritmo de Construcción GRASP**

En la Figura 8 se presenta un algoritmo de construcción GRASP que usa dos parámetros de relajación ( $\alpha$  y  $\theta$ ). El parámetro  $\alpha$  es usado para relajar el criterio goloso de selección del paquete a empaquetar (pasos 3.1 – 3.4); entretanto, el parámetro  $\theta$  es usado para relajar el criterio de selección contenedor (pasos 3.5 – 3.7). Se observa que las listas de candidatos restringidos de los paquetes y de los contenedores

se han denotado respectivamente por  $RCL_J$  y  $RCL_B$ .

```

ConstructionGRASP Algorithm
1. InputInstance( $n, P_1, P_2, \dots, P_n, L, A, H$ );
2.  $J := \{1, 2, \dots, n\}; \quad m := 0;$ 
3. While  $J \neq \emptyset$ 
3.1    $V := \text{Max}\{V(P_t) : t \in J\};$ 
3.2    $L := \text{Min}\{V(P_t) : t \in J\};$ 
3.3    $RCL_J := \{t \in J : \bar{V} - \alpha(\bar{V} - L) \leq V(P_t) \leq \bar{V}\};$ 
3.4    $j := \text{Random}(RCL_J);$ 
3.5    $R := \text{Min}_{1 \leq k \leq m+1} \{R_k : B_k \supseteq P_j\};$ 
3.6    $RCL_B := \{k \in \{1, \dots, m+1\} : B_k \supseteq P_j, R \leq R_k \leq R + \theta(LAH - R)\};$ 
3.7    $i := \text{Random}(RCL_B);$ 
3.8   If  $i = m+1$ 
3.9     Then Encaje_Contenedor_Nuevo( $P_j, B_i$ );
        •  $m := m+1;$ 
3.10    Else Encaje_Contenedor_Usado( $P_j, B_i$ );
3.11     $J := J - \{i\};$ 
4. Return ( $m, B_i \forall i$ )
    
```

Figura 8: Algoritmo Construcción GRASP.

Se observa que, cuando  $\alpha \rightarrow 0$  y  $\theta \rightarrow 0$  los criterios GRASP de selección del paquete y del contenedor convergen al criterio goloso BFD. Esto es, el algoritmo goloso BFD constituye un caso particular del algoritmo ConstructionGRASP.

Los algoritmos GRASP son llamados de adaptativos porque incluyen un proceso de adaptación en el criterio de selección del requerimiento a atender. Para el algoritmo ConstructionGRASP, el proceso adaptativo es dado en el paso 3.11. El proceso adaptativo establece que el paquete empaquetado no debe ser incluido en los siguientes procesos de selección de los paquetes a empaquetar.

**4.2. Procedimiento LocalSearch**

La segunda componente de un algoritmo

GRASP es un algoritmo de búsqueda local (Local Search), el cual tiene por objetivo mejorar la solución encontrada por el algoritmo de construcción GRASP. Existen diversas técnicas para construir una mejor solución a partir de una solución dada, para una revisión de heurísticas y Metaheurísticas. Ver, por ejemplo, Glover y Kochemberger (2003).

Sea  $S = \{B_1, B_2, \dots, B_m\}$  una solución dada, deseamos encontrar una mejor solución a partir de S, esto es una solución que presente menor número de láminas. Ciertamente, no siempre es posible mejorar una solución dada. El siguiente teorema demostrado en Mauricio et al (2004) establece una condición suficiente de optimalidad.

**Teorema 1**

Sea  $S = \{B_1, B_2, \dots, B_m\}$  una solución del problema 3-BPP, y  $R_i$  el respectivo espacio no utilizado en el contenedor  $B_i \forall i = 1, \dots, m$ . Si  $\sum_{i=1}^m R_i < WHD$ , entonces S es una solución óptima de 3-BPP.

Un procedimiento de mejoría está conformado por tres pasos. Primero, se determina el contenedor que presentan mayor espacio no usado, el cual se denota por  $B_{r1}$ . Segundo, se determina el contenedor que presenta mayor espacio no usado sin considerar  $B_{r1}$ , el cual se denota por  $B_{r2}$ . Tercero, se realizan los intercambios de los paquetes entre el contenedor  $B_{r2}$  y los paquetes uno a uno de los contenedores restantes de S, de forma que el espacio no usado de  $B_{r2}$  o de los contenedores restantes permita empaquetar un paquete de  $B_{r1}$ . Los pasos 2 y 3 se repetirán hasta que todos los paquetes de  $B_{r1}$  hayan encajado en los otros contenedores, esto es, hasta que no sea necesario usar el contenedor  $B_{r1}$ . Si no es necesario usar  $B_{r1}$ , se habrá conseguido reducir el número de contenedores en una unidad, esto



es, se habrá conseguido mejorar la solución S, y el procedimiento podrá repetirse a fin de seguir mejorando la solución, de lo contrario el procedimiento deberá terminar.

## 5. EXPERIMENTOS NUMÉRICOS

### 5.1. Configuración

Grupos	Goloso	GRASP			
		N=100	N=500	N=1000	N=1500
1	4	3	3	3	3
2	3	3	3	3	3
3	4	4	4	3	3
4	8	8	8	8	8
5	1	1	1	1	1
6	2	2	1	1	1
7	2	2	2	1	1
8	2	2	2	1	1
Promedio	3.25	3.125	3.00	2.625	2.625

Tabla 1: Resultado Numéricos, Goloso BFD y GRASP con  $\alpha = 0.2$  y  $\theta = 0.5$ .

### 5.2. Instancias de pruebas

Se han generado 8 grupos de 10 instancias de pruebas cada uno considerando los criterios dados en Castro et al (2003), el detalle de estas instancias se describen en Rojas et al (2006). Existen varias técnicas de calibración del parámetro de relajación. Véase, por ejemplo, el capítulo 8 de Glover et al (2003); en este trabajo los parámetros de relajación han sido calibrado ejecutando el algoritmo ConstructionGRASP sobre las instancias de pruebas variando los valores de  $\alpha$  y  $\theta$  desde 0.1 hasta 0.5, con 1000 iteraciones, obteniéndose que los mejores valores para los parámetros son:  $\alpha = 0.2$  y  $\theta = 0.5$ . Con estos valores de los parámetros de relajación se ha ejecutado el algoritmo ConstructionGRASP para varios valores de números de iteraciones (N) y, para ver su eficiencia, se ha implementado el algoritmo goloso BFD obteniéndose los resultados mostrados en la Tabla 1.

## 6. CONCLUSIONES

El presente trabajo ha presentado un algoritmo GRASP para resolver el problema Three-Dimensional Bin Packing Problem (3D-BPP). El

Se ha implementado el algoritmo Construction GRASP en Visual Basic V.6.0 con manejador de base de datos Microsoft Access 2000. Y se han realizado experimentos sobre un procesador Intel Pentium IV, de 1.6 Ghz, 512 Mbytes de RAM, con sistema de video GetForce MX/MX 400 y con sistema operativo Windows 98.

algoritmo propuesto de Construcción GRASP está basado en el algoritmo goloso BFD, y considera dos parámetros de relajación. La introducción de dos parámetros de relajación tiene por objetivo generar mayor diversidad de las soluciones para el problema, esto significa teóricamente que se pueden obtener mejores soluciones que por los propuestos algoritmos GRASP que usan solo un parámetro de relajación. Los experimentos numéricos sobre 80 instancias de pruebas muestran mejores resultados (menor uso de contenedores) que el algoritmo goloso BFD, además este mejora a medida que aumenta el número de iteraciones. Se ha conseguido, con 1000 iteraciones, una mejora de la solución del orden del 19.53% respecto al algoritmo goloso.

## REFERENCIAS

Bischoff E., Dowsland W.V. (1982), An application of the microcomputer to produce design distribution. *Journal of Operation Research* 33, 271-280.

Cruz C., and Mauricio D. (1997), O problema de alocação de facilidades não desejadas: aplicações, algoritmos. *Anais do XXIX Simpósio*

*Brasileiro de Pesquisa Operacional*.  
Salvador, 22-24 de Octubre de 1997.

**Castro J., Soma N., Maculan N.** (2003), A greedy search for the tri-dimensional bin packing problem: the packing static stability case. *International Transactions in Operation Research*, 10, 141-153.

**Corcoran A.L., Wainwright R.L.** (1992), A genetic algorithm for packing in three dimensions. Proceeding of the 1992 ACM / SIGAPP Symposium on Applied Computing, 1021-1030.

**Dyckhoff H.** (1990), A typology of cutting and packing problems. *European Journal of Operation Research* 44, 145-159.

**Faina L.** (2000), A global optimization algorithm for three-dimensional packing problem. *European Journal of Operation Research*, 126, 340-354.

**Feo T., and Resende M.** (1989), A probabilistic heuristic for a computational difficult set covering problem. *Operation Research Letters*, 8, 67 – 71.

**Feo T., and Resende M.** (1995), Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6, 109 -133.

**Feo T., Resende M., and Smith S.** (1994), A greedy randomized adaptive search procedures for a maximum independent set. *Operation Research*, 42 (5), 860-878.

**Glover F. and Kochenberger G.** (2003), *Handbook of Metaheuristic*. Kluwer Academic Publisher.

**Garey M.R., Johnson D.S.** (1979), *Computer and intractability: A guide of theory of NP-Completeness*. W. H. Freeman, San Francisco.

**Ganoza D., Solano U.** (2004), *Un algoritmo de búsqueda adaptativo aleatoria y golosa para resolver el problema de cortes*. Tesis de Ingeniería, FISI - Universidad Nacional Mayor de San Marcos, Lima- Perú.

**Gilmore J.F., Williams E., Fossat E., Bohlander R., Eilling G., P., and Gomory R.** (1989), An expert system approach to palletizing unequaled containers, *SPIE* 1095, 933-942.

**George J. A., Robinson D.F.** (1980), A heuristic for packing boxes into a container. *Computers and Operational Research* 7, 147-156.

**Gehring H., Menschner K., Meyer M.** (1990), A computer based heuristic for packed and pooled shipment containers. *European Journal of Operational Research*, 44, 277-288.

**Han C.P., Knott K., Egbelu P.J.** (1989), A heuristic approach to the tri dimensional cargo – loading problem. *International Journal of Operation Research* 27 (5), 757-774.

**Li K., Cheng K.H.** (1992), Heuristic algorithms for one-line packing in three dimensions. *Journal of Algorithms* 13 (1992) 589-605.

**Lodi A., Martello S., and Vigo D.** (2002), Heuristic algorithms for the three-dimensional bin packing problem. *European Journal of Operation Research*, 114, 410–420.

**Morábito R.N., Arenales M.N.** (1993), Uma abordagem em grafos E/OU para o problema do cargamento de containeres. In *Anais do XXV Simposio Brasileiro de Pesquisa Operacional*. Campinas SP, UNICAMP.

**Mauricio D., Roja A., Calderon G.** (2004), Algoritmos GRASP para resolver el problema de cortes en 3D. *Reporte Técnico UPG – FISI N° 07/2004*, UNMSM, Lima – Perú.

**Miyazawa F.** (1993), *Algoritmos de empacotamento tridimensional: novas estratégias e análises de desempenho*. Tesis Msc. Instituto de Matemática e Estatística – USP.

**Martello S., Pisinger D., Vigo D.** (2000), The three-dimensional bin packing problem. *Operation Research*, 48, 256-267.

**Miyazawa F., Wakabayashi Y.** (1997), An algorithm for the three-dimensional packing problem with asymptotic performance analysis. *Algoritmica* 18, 122-144.

**Osogami T., Okano H.** (2003), Local search algorithms for the bin packing problem and their relationship to various constructions heuristics. *Journal of Heuristics* 9, 29-49.

**Pisinger D., Faroe O., Zachariasen M.** (1999),

local search for the three-dimensional bin packing problem. Technical Report, DIKU 99-13, Kobenhavns Universitet, Datalogisk Institute.

**Roja A., Calderon G.** (2006), Algoritmos GRASP para resolver el problema de Bin Packing con Restricciones de Estabilidad. Tesis de Ingeniero, Universidad Nacional Mayor de San Marcos, Lima, Perú.

**Roberts D., Taylor D.** (1972), A distributions and railcar loading system. Proceeding ORSA/TIM Conference.

**Haessler R.** (1979), Solving a boxcar loading problem. Proceeding ORSA/TIM Conference.

