

Fase de postprocesamiento de una Búsqueda Constructiva Adaptativa (GRASP)

Jesús David Beltrán, Jose Eduardo Calderón
Rayco Jorge Cabrera, J. Marcos Moreno Vega

Departamento de Estadística, I.O. y Computación
Centro Superior de Informática
Universidad de La Laguna, 38271 La Laguna, SPAIN
jmmoreno@ull.es

Resumen Tras obtener una solución para un problema de optimización, muchas técnicas heurísticas de resolución de problemas aplican un procedimiento de mejora a esta. La gran mayoría de los métodos de mejora se basan en Búsquedas Locales. Proponemos un procedimiento alternativo para el problema del empaquetado rectangular bidimensional no guillotina que consiste en reconstruir la solución. Para determinar la iteración a partir de la cual reconstruir, se realiza un análisis de la solución a la que se desea aplicar el procedimiento de mejora. Los resultados experimentales obtenidos muestran que se mejora la calidad de la solución y que la técnica es eficaz para resolver el problema.

1. Introducción

Muchas técnicas heurísticas de resolución de problemas aplican procedimientos de mejora a soluciones previamente obtenidas. Los procedimientos de mejora más conocidos son las Búsquedas Locales en las que, una vez definido el concepto de entorno de una solución, se escoge una solución del mismo que mejore a la solución inicial. Si esta solución existe, el procedimiento reitera el paso anterior con ella. En caso contrario, se finaliza la búsqueda. La gran mayoría de los procedimientos de mejora que pueden aplicarse a una solución son variantes de las Búsquedas Locales o emplean a estas como elementos importantes de su diseño.

Presentamos un procedimiento alternativo para mejorar la calidad de una solución del problema del empaquetado rectangular bidimensional no guillotina. El procedimiento tiene su origen en el análisis de una técnica constructiva GRASP [3] previamente propuesta para este problema. Se ha observado que esta técnica tiende a ubicar de forma incorrecta los rectángulos en las últimas iteraciones del método constructivo. Por ello, se propone reconstruir la solución obtenida por el método constructivo extrayendo los últimos rectángulos de la solución para ubicarlos usando una técnica heurística. Realizamos un análisis de la solución obtenida por el método constructivo para determinar qué rectángulos

deben extraerse de la solución. Con ello, adaptamos el procedimiento de mejora al problema y así aumentamos la eficiencia del mismo.

El presente trabajo se estructura de la siguiente forma. En la próxima sección se introduce el problema del empaquetado rectangular bidimensional no guillotina. En la sección 3 se expone la técnica constructiva GRASP y en la sección 4 se describe la particular implementación que se hace de la misma para el problema del empaquetado rectangular bidimensional no guillotina. Además, se describe el análisis de la solución que da lugar al procedimiento de mejora. Por último, se muestra la experiencia computacional realizada y se enumeran las conclusiones que se siguen de los resultados obtenidos.

2. Empaquetado rectangular bidimensional no guillotina

Los problemas de empaquetado constituyen una amplia clase de problemas en los que, de forma general, se desea empaquetar un conjunto de items (figuras geométricas pequeñas) en un objeto geométrico mayor (o conjunto de objetos) de tal forma que se optimice algún objetivo relativo al empaquetado obtenido.

La importancia de estos problemas en procesos industriales o de gestión financiera se refleja en la gran cantidad de trabajos aparecidos en la literatura científica. Algunos trabajos de revisión y clasificación en los que también se enumeran aplicaciones son [6] [7] [8] y [11].

Aquí consideramos el problema del empaquetado rectangular bidimensional no guillotina que se formula como sigue. Dado un objeto rectangular de amplitud fija w y altura infinita, y un conjunto, $\mathcal{R} = \{R(w_1, h_1), \dots, R(w_n, h_n)\}$, de rectángulos con al menos uno de sus lados, w_i , h_i , menor que w , se desea empaquetar el conjunto \mathcal{R} en el objeto rectangular utilizando el menor espacio posible (o lo que es lo mismo, se pretende minimizar la altura del empaquetado). En este problema se pueden rotar los objetos y los cortes pueden ser de tipo no guillotina (ver figura 1(a)).

3. Greedy Randomized Adaptive Search Procedures

GRASP (Greedy Randomized Adaptive Search Procedure) [9] [10] es un procedimiento heurístico que consta de varias etapas. A una fase constructiva, en la que se escoge al azar un elemento de la lista restringida de candidatos, le sigue una fase de postprocesamiento en la que se mejora la solución obtenida en la fase anterior. Los anteriores pasos se reiteran hasta que se cumpla el criterio de parada. La mejor solución obtenida es la propuesta por el algoritmo.

Los elementos que determinan completamente la técnica GRASP son: la función heurística, la forma en que se construye la lista restringida de candidatos, el método de postprocesamiento y el criterio de parada.

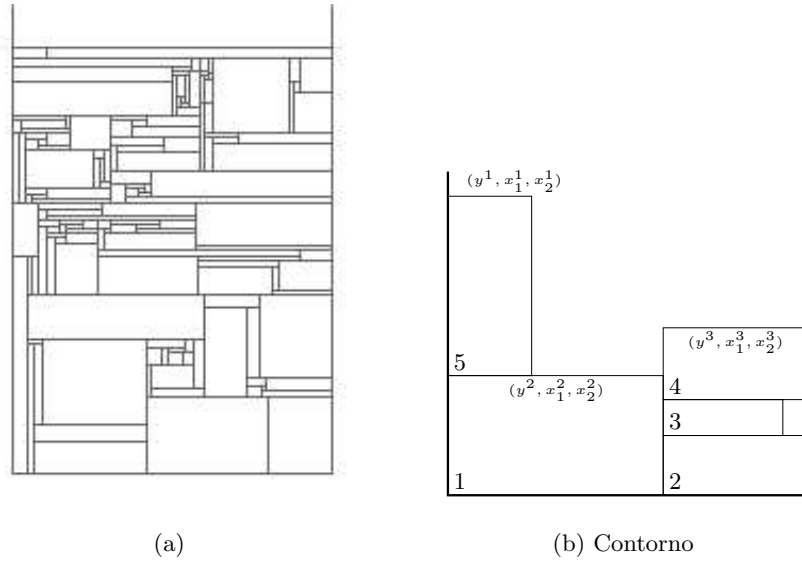


Figura 1.

3.1. Fase constructiva. Lista restringida de candidatos

Supongamos que para el problema particular que se está abordando se dispone de una función heurística, f , que mide la conveniencia de considerar cada elemento como parte de la solución. Supongamos además, sin pérdida de generalidad, que cuanto mayor sea la adecuación de un elemento, mayor es la evaluación. En un estrategia greedy se escoge, en cada iteración, el elemento que maximiza la función f . Como se ha mencionado anteriormente, esta estrategia suele suministrar pobres resultados y, en general, no puede asegurarse que con ella se obtenga la solución óptima del problema.

Una alternativa para superar este inconveniente consiste en escoger, no el mejor elemento, según la evaluación dada por f , sino uno de los mejores al azar. Para ello, se construye una lista restringida de candidatos que incluye a los mejores elementos. Esta lista se puede construir de varias formas. Algunas de las posibilidades son

1. *por cardinalidad*: la lista está formada por los k (parámetro fijado por el usuario) elementos con mayor evaluación de la función heurística;
2. *por rango*: la lista está formada por los elementos cuya evaluación está a una distancia no superior a un umbral fijado por el usuario. Esto es, dado un valor $\alpha \in [0, 1]$, la lista restringida de candidatos la forman los elementos

cuya evaluación está en el intervalo $[(1 - \alpha)MAX, MAX]$, siendo MAX la evaluación del mejor elemento;

3. *por intersección de las dos anteriores*: en cada iteración del proceso constructivo, la lista la forman los elementos que pertenecen simultáneamente a los dos conjuntos anteriores.

3.2. Fase de postprocesamiento

En un GRASP, a cada una de las soluciones obtenidas en la fase constructiva, se le aplica un procedimiento de mejora que devuelve una solución que mejora a la de partida. Como postprocesamiento suele emplearse una simple búsqueda local descendente, aunque también han sido exploradas otras alternativas. En [2] se enumeran algunas de estas alternativas: aplicar un Recocido Simulado desde la solución, aplicar un Algoritmo Genético con población inicial de soluciones construida mediante GRASP, etc.

4. GRASP para el problema del empaquetado de rectángulos

En la presente sección describimos un GRASP para el problema del empaquetado rectangular bidimensional no guillotina. En [3] se encuentra una descripción detallada de este GRASP junto al análisis de amplios resultados experimentales que indican que esta técnica es la más eficiente y eficaz para el problema.

La definición de la lista restringida de candidatos se realiza a partir del concepto de contorno superior.

4.1. Contorno

La inclusión de un rectángulo cualquiera en el objeto, determina un contorno superior rectangular como el que se muestra en la figura 1(b). Además, es posible que se obtengan áreas no aprovechables, llamadas desperdicios, como el que se obtiene al incluir el rectángulo 4 en el objeto de la figura 1(b). El contorno, C , puede representarse por medio del conjunto de segmentos horizontales (tomados de izquierda a derecha) que lo forman. Es decir:

$$C = \{(y^1, x_1^1, x_2^1), (y^2, x_1^2, x_2^2), \dots, (y^c, x_1^c, x_2^c)\}$$

con

$$\begin{aligned} y^i &\equiv \text{altura del } i\text{-ésimo segmento} \\ x_1^i &\equiv \text{punto inicial del } i\text{-ésimo segmento.} \\ x_2^i &\equiv \text{punto final del } i\text{-ésimo segmento} \end{aligned}$$

Además, $x_1^1 = 0$ y $x_2^c = w$. Nótese que, intuitivamente, es preferible un contorno formado por pocos niveles a otro con muchos niveles. Esto es así, ya que, en general, la posibilidad de obtener desperdicios aumenta con el número de niveles.

4.2. Lista restringida de candidatos

Sea t la iteración actual del proceso constructivo y supongamos que $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$, siendo \mathcal{R}_1 el conjunto de los rectángulos previamente incluidos en el objeto y $\mathcal{R}_2 = \mathcal{R} \setminus \mathcal{R}_1$. Sea $C(t)$ el contorno determinado por los rectángulos de \mathcal{R}_1 . Evaluaremos la conveniencia de incluir un rectángulo de \mathcal{R}_2 en el objeto por la forma que tendrá el contorno $C(t)$ tras su inclusión.

1. *Lista restringida de candidatos:* sea dado $\alpha \in [0, 1]$ y supongamos que el segmento del contorno con menor altura es (y^i, x_1^i, x_2^i) . La lista restringida de candidatos se construye como sigue:

$$LRC = \{R(w_j, h_j) \in \mathcal{R}_2 : (0 \leq x_2^i - x_1^i - w_j \leq \alpha) \vee (0 \leq x_2^i - x_1^i - h_j \leq \alpha)\}.$$

Es decir, la lista está formada por aquellos rectángulos que mejor se ajustan al ancho del segmento inferior del contorno. El ajuste viene determinado por el valor de α .

Para que la anterior definición tenga sentido, debe haber, al menos, un rectángulo de \mathcal{R}_2 , digamos $R(w_r, h_r)$, tal que $(0 \leq x_2^i - x_1^i - w_r \leq \alpha) \vee (0 \leq x_2^i - x_1^i - h_r \leq \alpha)$. Si ningún elemento de \mathcal{R}_2 cumple la anterior condición, se ubica dentro del objeto el rectángulo que mejor se ajusta a $x_2^i - x_1^i$, y se reconstruye el contorno. Si no existe tal rectángulo, se reconstruye el contorno eliminando, convenientemente, el segmento (y^i, x_1^i, x_2^i) .

4.3. Fase de postprocesamiento

Una de las situaciones anómalas que puede presentarse al aplicar los métodos constructivos anteriores se muestra en la figura 2. Consideremos la ubicación del rectángulo 6. Cualquiera de los métodos anteriores lo ubicaría según se indica en la figura 2(b). La bondad de esta nueva situación depende del instante en que se produce. En las primeras iteraciones del método, la situación es aconsejable. No obstante, en las últimas iteraciones puede producir soluciones de baja calidad. En particular, si nos encontramos en la última iteración, sería preferible ubicarlo como se muestra en la figura 2(c).

El anterior comportamiento es característico del método constructivo que empleamos. Por ello, en [3] se proponía el siguiente procedimiento de mejora, que se aplica a la solución obtenida en la fase constructiva.

Procedimiento de mejora: extraer los últimos m (parámetro fijado por el usuario) rectángulos de la solución. Supongamos, por simplicidad, que son $\{R_1, R_2, \dots, R_m\}$. Para cada permutación, $(R_{i_1}, R_{i_2}, \dots, R_{i_m})$, de los rectángulos:

- 1.) Hacer $j = 1$. Colocar el rectángulo R_{i_j} en la posición más profunda del objeto y con la orientación que suponga una menor altura relativa. Actualizar el contorno.
- 2.) Hacer $j = j + 1$. Tomar el rectángulo R_{i_j} de la permutación y empaquetarlo siguiendo el proceso anterior.
- 3.) Si $j = m$, parar; en caso contrario repetir el paso 2. Devolver la mejor de las soluciones obtenidas con el método anterior.

A pesar de que la anterior fase de postprocesamiento mejora la calidad de las soluciones obtenidas en la fase constructiva (ver [3]), tiene dos inconvenientes.

1. *Valores limitados del parámetro m* : dado que se realiza una búsqueda exhaustiva entre todas las posibles combinaciones de los últimos m rectángulos, hay que limitar esta búsqueda a valores pequeños de m .
2. *Obligación de fijar a priori el valor de m* : soluciones diferentes requerirán, posiblemente, reconstruir desde puntos distintos. Fijar a priori el valor de m impide que la fase de postprocesamiento se adapte a la solución obtenida en la fase constructiva.

Para subsanar estos inconvenientes, se propone analizar el contorno superior que se obtiene en cada iteración del proceso constructivo y determinar de esta forma el valor de m , y realizar una búsqueda heurística entre todas las combinaciones de los últimos m rectángulos.

Análisis del contorno La fase constructiva de un GRASP para el problema del empaquetado de rectángulos bidimensional no guillotina consta de n iteraciones (con n el número de rectángulos a empaquetar). Sea $C(t)$ el contorno superior que se obtiene al incluir un rectángulo en la t -ésima iteración.

Asociado al contorno $C(t)$ pueden considerarse varios valores que miden la suavidad del mismo. Uno de estos valores es la altura media de los niveles que se define como

$$\text{AlturaMedia}(C(t)) = \frac{1}{c} \sum_{i=1}^c (y^+(C(t)) - y^i) \quad t = 1, \dots, n$$

donde c es el número de niveles del contorno y

$$y^+(C(t)) = \max_{i=1, \dots, c} \{y^i\} \quad t = 1, \dots, n.$$

Convenimos que $\text{AlturaMedia}(C(0)) = 0$. Sea asimismo

$$\Delta \text{AlturaMedia}(t) = \text{AlturaMedia}(C(t-1)) - \text{AlturaMedia}(C(t)), \quad t = 1, \dots, n$$

el incremento que se produce en la altura media del contorno al incluir el rectángulo de la t -ésima iteración.

Proponemos dos formas de obtener la iteración (o equivalentemente el número de rectángulos) a partir de la cual aplicar el método de mejora.

1. *Mayor incremento en la altura media.* Aplicar el procedimiento de mejora a partir de la iteración en que se produce un mayor incremento en la altura media. Es decir, si t^* es la iteración que maximiza

$$\Delta \text{AlturaMedia}(t) = \text{AlturaMedia}(C(t-1)) - \text{AlturaMedia}(C(t)),$$

$$t = 1, \dots, n,$$

se extraen los rectángulos empaquetados en las iteraciones que van desde la t^* hasta la n .

2. *Menor altura media.* Aplicar el procedimiento de mejora a partir de la iteración en que el contorno es más suave. Es decir, si t^* es la iteración que minimiza

$$\text{AlturaMedia}(C(t)), \quad t = 1, \dots, n,$$

se extraen los rectángulos empaquetados en las iteraciones que van desde la $t^* + 1$ hasta la n .

Búsqueda heurística Nótese que una vez determinada la iteración, t^* , a partir de la cual reconstruir, se obtiene un nuevo problema de empaquetado de dimensión menor en el que la frontera inferior del objeto en que se deben incluir los rectángulos viene dada por el contorno $C(t^*)$. Este nuevo problema puede abordarse por cualquiera de las técnicas heurísticas de resolución de problemas. Hemos experimentado con una Búsqueda Local Descendente.

1. *Búsqueda Local Descendente.* Dada una permutación de los rectángulos y el movimiento consistente en intercambiar el orden de dos ellos, realizar el mejor de los movimientos mientras sea posible. Los rectángulos se empaquetan siguiendo los pasos 1, 2 y 3 del procedimiento de mejora descrito en la sección 4.3.

4.4. Criterio de parada

La búsqueda realizada con GRASP finaliza después de un número dado, $n_{iter} = 20$, de pasadas del bucle *Fase Constructiva*, *Fase de Postprocesamiento*.

5. Experiencia computacional

Para evaluar el comportamiento de nuestras propuestas de mejora de las soluciones obtenidas en la fase constructiva del GRASP, se resolvieron diferentes problemas generados aleatoriamente. Se implementó un generador de problemas que, dado el ancho del objeto rectangular, w , el número de rectángulos, n , y el valor objetivo óptimo, h_{opt} , suministra un conjunto de n rectángulos que pueden ubicarse en un objeto rectangular de amplitud w utilizando una altura h_{opt} . En la

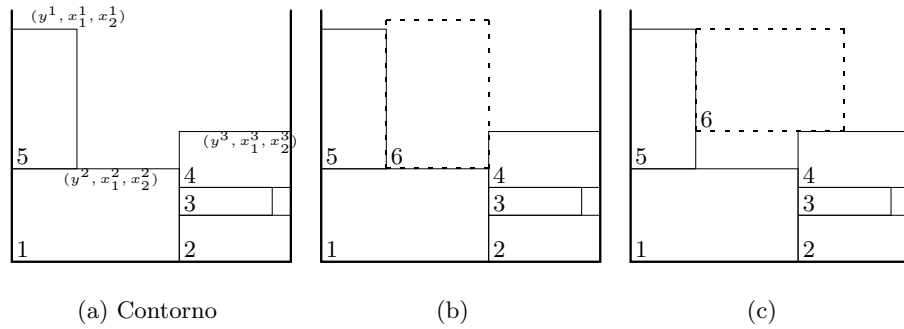


Figura 2. Procedimiento de mejora

figura 1(a) se muestra uno de los problemas obtenidos con este generador. Cada problema fué resuelto 5 veces realizando 20 pasadas del bucle *Fase Constructiva*, *Fase de Postprocesamiento*. El valor que determina el umbral de ajuste en el proceso constructivo del GRASP se fijó a $\alpha = 0$.

En la tabla 1 se muestran los resultados obtenidos en la experiencia computacional. Las tres primeras columnas describen el problema: número de rectángulos (n), ancho del objeto rectangular (w) y altura óptima (h_{opt}). A continuación aparecen 12 columnas agrupadas en dos grupos de 6 columnas. Cada grupo se corresponde con uno de los criterios propuestos para obtener la iteración a partir de la cual aplicar el método de mejora.

En las columnas 4 y 5 se recogen el mejor valor objetivo (Obj) y el valor objetivo medio obtenido en las 20 fases constructivas del GRASP. En las columnas 6 y 8 se muestran estos mismos valores tras la fase de postprocesamiento. Las columnas 7 y 9 almacenan las mejoras producidas tras la fase de postprocesamiento. La descripción de las columnas que van desde la 10 hasta la 15 es similar.

Por cada problema se muestran 5 filas de resultados (una por cada una de las 5 ejecuciones del GRASP realizadas) más una que recoge los valores medios de las mejoras. De los resultados obtenidos podemos concluir lo siguiente.

1. *La fase de postprocesamiento mejora la calidad de las soluciones.* La mejora se produce tanto en el mejor valor objetivo como en el valor objetivo medio. Aunque la mejora pueda parecer no significativa hay que señalar que las soluciones generadas en la fase constructiva son, en la gran mayoría de los casos, de alta calidad. Por ello, el efecto de la mejora puede parecer menor.
2. *Existe una ligera diferencia en el comportamiento de los criterios.* Aunque las diferencias no son significativas, se observa que, para problemas con el mismo número de rectángulos, el criterio basado en el *mayor incremento en la altura media* produce una mayor mejora en el mejor valor objetivo

Cuadro 1. Evaluación experimental del postprocesamiento

<i>n</i>	<i>w</i>	<i>h_{opt}</i>	Mayor incremento en la altura media						Menor altura media					
			Constr.			Postproc.			Constr.			Postproc.		
			<i>Obj</i>	\overline{Obj}	<i>Mej.</i>	<i>Obj</i>	\overline{Obj}	<i>Mej.</i>	<i>Obj</i>	\overline{Obj}	<i>Mej.</i>	<i>Obj</i>	\overline{Obj}	<i>Mej.</i>
50	50	50	51	52,6	51	0	52,45	0,15	52	52,3	52	0	52	0,3
			52	53,15	51	1	52,95	0,2	52	52,35	51	1	51,95	0,95
			52	55,7	51	1	55,15	0,55	52	52,3	52	0	52,3	0
			52	55,2	51	1	54,9	0,3	52	52,5	52	0	52,4	0,1
			52	52,55	52	0	52,45	0,10	53	56	52	1	55,5	0,5
50	40	60				0,6		0,26				0,4		0,26
			64	64,2	62	2	63,75	0,45	62	63,85	62	0	63,6	0,25
			62	65,55	62	0	63,15	0,85	64	65,2	62	2	64,25	0,95
			64	66,3	63	1	64,9	1,4	63	66,55	62	1	62,75	3,8
			64	65,9	64	0	64,4	0,4	63	63,5	62	1	63,1	0,4
100	50	50	63	65,45	63	0	64,45	1	63	65,95	63	0	63,65	2,3
						0,6		0,82				0,8		1,54
			52	52,55	52	0	52,55	0	51	51,7	51	0	51,4	0,3
			52	52	51	1	51,35	0,65	51	51,9	51	0	51,5	0,4
			52	52	52	0	52	0	52	52,1	52	0	52,1	0
100	50	75	51	51,65	51	0	51,65	0	52	52,7	52	0	52,35	0,35
			52	52	51	1	51,35	0,65	52	52,5	51	1	51,6	0,9
						0,4		0,26				0,2		0,39
			77	77,35	77	0	77,05	0,3	77	77	76	1	76,75	0,25
			76	76,75	76	0	76,75	0	77	78,05	77	0	78,05	0
200	100	100	77	77,3	77	0	77	0,3	76	76,95	76	0	76,7	0,25
			77	77	77	0	77	0	76	76,7	76	0	76,7	0
			77	77,55	77	0	77,25	0,3	77	77,25	76	1	76,85	0,4
						0		0,18				0,4		0,18
			101	101,8	101	0	101,8	0	101	101,65	101	0	101,65	0
200	120	160	101	101,8	101	0	101,5	0,3	101	103,7	101	0	103,7	0
			102	102,2	101	1	102,05	0,15	101	102	101	0	101,8	0,2
			101	103,85	101	0	102,65	1,2	102	102	102	0	102	0
			101	102,2	101	0	102,05	0,15	102	103,5	102	0	103,5	0
						0,2		0,36				0		0
200	120	160	163	164,75	162	1	163,7	1,05	162	163,05	162	0	163,05	0
			163	166,9	162	1	165,3	1,6	162	164,6	162	0	163,85	0,75
			162	169,25	162	0	169,25	0	162	164,2	162	0	163,8	0,6
			163	165,55	162	1	165,05	0,5	163	165,7	162	1	165,6	0,1
			165	167,1	163	2	165,75	1,35	164	169,2	164	0	169,2	0
			1		0,9				0,2		0,47			

de aquellos problemas con h_{opt} menor. Para los problemas con mayor h_{opt} , el criterio que suministra mayores mejoras es el basado en la *menor altura media*.

3. *La técnica GRASP propuesta es eficaz en la resolución del problema.* La distancia que existe entre el mejor valor objetivo encontrado por GRASP y el valor objetivo óptimo es inferior en todos los casos a 2 unidades (para cuatro problemas es inferior a 1 unidad y para los otros dos es inferior a 2 unidades). Este comportamiento ya se observó en la amplia experiencia computacional desarrollada en [3].

Referencias

1. Aarts, E, Korst, J. Simulated Annealing and Boltzmann Machines. John Wiley and Sons (1989)
2. Díaz, A. (coordinador) Optimización heurística y redes neuronales. Editorial Paraninfo (1996)
3. Beltrán, J. D, Eduardo, J., Jorge, R., Moreno-Vega, J.M. Procedimientos constructivos adaptativos (GRASP) para el problema del empaquetado bidimensional Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial **15** (2002) 26–33
4. Boender, C.G.E., Rinnooy Kan, A.H.G., Vercellis, C.: Stochastic Optimization Methods. Stochastics in Combinatorial Optimization (1986) 94–112
5. Boender, C.G.E., Rinnooy Kan: Bayesian Stopping Rules for Multistart Global Optimization Methods. Mathematical Programming **37** (1987) 59–80
6. Dowsland, K.A., Dowsland, W.B. Packing Problems European Journal of Operational Research **56** (1992) 2–14
7. Dyckhoff, H. Typology of Cutting and Packing Problems. European Journal of Operational Research **44** (1990) 145–159
8. Dyckhoff, H., Finke, U. Cutting and Packing in Production and Distribution. Springer Verlag, Berlin (1992)
9. Feo, T. A., Resende, M.G.C. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters **8** (1989) 67–71
10. Feo, T. A., Resende, M.G.C. Greedy Randomized Adaptive Search Procedures. Journal of Global Optimization **6** (1995) 109–133
11. Hopper, E., Turton, B.C.H. A Review of the Application of Meta-Heuristics Algorithms to 2D Strip Packing Problems. Artificial Intelligence Review **16** (2001) 257–300
12. Los, M., Lardinois, C. Combinatorial Programming, Statistical Optimization and the Optimal Transportation Network Problem. Transportation Research **2** (1982) 89–124
13. Moreno-Vega, J. M., Moreno, J. A. Una Regla de Parada para la Búsqueda con Arranque Múltiple. Actas de las I Jornadas de Informática. (1995) 271–280