SOLUCIÓN AL PROBLEMA DE EMPAQUETAMIENTO ÓPTIMO BIDIMENSIONAL EN ROLLOS INFINITOS USANDO UN ALGORITMO HÍBRIDO

A hybrid algorithm for the two-dimensional strip packing problem

RESUMEN

En este artículo se presenta un algoritmo para resolver problemas medianos y de gran escala de empaquetamiento óptimo en rollos infinitos. Este consiste en empacar en largos rollos con ancho fijo y longitud ilimitada pequeños rectángulos, sin violar los valores de demanda impuestos. El algoritmo propuesto transforma el problema original (*strip packing*) en pequeños problemas de empaquetamiento óptimo en contenedores (*bin packing*). A su vez, éstos son resueltos mediante una técnica metaheurística híbrida, búsqueda en vecindario variable y recocido simulado. La eficiencia del algoritmo propuesto es evaluada usando 34 casos de estudio, de mediano y gran tamaño; los resultados obtenidos son de excelente calidad al ser comparados con los de la literatura especializada.

PALABRAS CLAVES: empaquetamiento bidimensional en rollos infinitos, optimización combinatorial, problemas de corte y empaquetamiento.

ABSTRACT

In this paper we present an algorithm for solving approximately large-scale strip cutting/packing problems. The strip packing problem consists of packing a large strip with a fixed-width and unlimited length into smaller rectangles, without violating the demand values imposed on each rectangle. The strip packing problem is reduced to a series of single constrained cutting stock problem and solved by using hybrid metaheuristic, variable neighborhood search and simulated annealing. The performance of the proposed algorithm is evaluated on a set of medium and large size problem instances, the results are compared with literature, on this way the results are satisfy.

KEYWORDS: *combinatorial optimization, cutting and packing problems, two-dimensional strip packing.*

DAVID ÁLVAREZ MARTÍNEZ
Ingeniero de Sistemas y Computación.
Joven Investigador
COLCIENCIAS-Universidad Tecnológica de Pereira
davidalv@utp.edu.co

ELIANA TORO OCAMPO
Ingeniera Industrial, *M.Sc.*
Docente Asistente
Facultad de Ingeniería Industrial
Universidad Tecnológica de Pereira
elianam@utp.edu.co

RAMÓN GALLEGO RENDÓN
Ingeniero Electricista. *Ph.D.*
Docente Titular
Programa Ingeniería Eléctrica
Universidad Tecnológica de Pereira
ragr@utp.edu.co

1. INTRODUCCIÓN

El problema de strip packing es aplicado por profesionales de diferentes disciplinas entre los que se cuentan las ciencias computacionales, ingeniería industrial, logística, manufactura, administración y procesos de producción.

Los campos de aplicación en la industria son muy variados, siendo el principal en el que la materia prima utilizada viene en tiras cuya unidad de empaque es el rollo tales como acero, tela, papel, fibra de vidrio, plásticos, entre otros. En este tipo de problemas el objetivo es encontrar la longitud mínima de la tira ubicando la totalidad de las piezas requeridas y minimizando el desperdicio de material.

El problema de strip-packing en 2 dimensiones es considerado como NP-hard debido a que el espacio de solución crece de forma exponencial con relación al número de ítems a empacar [1].
Esta temática se considera clásica en la investigación de operaciones y ha sido estudiada ampliamente por la comunidad académica con resultados de interés como se comenta a continuación.

Este problema ha sido abordado a través de diferentes métodos clasificados como de solución exacta y aproximados. Algunos algoritmos heurísticos fueron desarrollados por Berkey and Wang [2], algoritmos híbridos combinando genéticos y métodos determinísticos para el problema de empaquetamiento ortogonal han sido propuestos por autores como Jakobs [3], Dagli y Phoshyanonda [4]. Una investigación empírica sobre algoritmos heurísticos y metaheurísticas para resolver el problema del strip packing fueron propuestos por Hopper y Turton en [5]. En [6] Belov propone modelos y algoritmos en una y dos dimensiones,

Fecha de Recepción: 8 de junio de 2009
Fecha de Aceptación: 28 de Agosto de 2009

Caprara y Monaci [7] y Fekete y Schepers [8] propusieron un algoritmo exacto basado en grafos.

En este trabajo se propone un algoritmo que transforma el problema original (strip packing) en pequeños problemas de empaquetamiento óptimo en contenedores (bin packing). A su vez, éstos son resueltos mediante una técnica metaheurística híbrida. La eficiencia del algoritmo propuesto es evaluada usando 34 casos de estudio, de mediano y gran tamaño; los resultados obtenidos son de excelente calidad al ser comparados con los de la literatura especializada.

Este documento está organizado de la siguiente forma: en la sección 2 se describe el problema de strip packing, en la sección 3 se presenta el modelo matemático, en la sección 4 se presenta el procedimiento empleado para la solución del problema usando el método propuesto, en la sección 5 se presentan el análisis de resultados obtenidos y finalmente en la sección 6 se presentan las conclusiones y trabajos futuros en esta área.

2. DESCRIPCIÓN DEL PROBLEMA

En la literatura especializada se conoce este problema bajo la denominación (2-SPP) del inglés two-dimensional strip packing problem, incluido en los problemas clásicos de corte y empaquetamiento e incluye las siguientes clasificaciones:

- RF Las piezas a ser ubicadas pueden ser rotadas 90° (R) y no se requiere que el corte sea del tipo guillotina (F).
- RG Las piezas a ser ubicadas pueden ser rotadas 90° (R) y se requiere que el corte sea del tipo guillotina (G).
- OF La piezas a ser ubicadas no se pueden rotar(O) y no se requiere que el corte sea del tipo guillotina (F).
- OG Las piezas a ser ubicadas no se pueden rotar(O) y se requiere que el corte sea del tipo guillotina (G).

De los problemas presentados anteriormente, en este trabajo se considera el problema RF 2-SPP [9].

Una forma grafica de explicar el problema es presentado a continuación:

En la figura 1 se presenta un ejemplo del tipo de piezas que pueden ser demandadas, en la figura 2 se muestra la materia prima a ser utilizada (rollo), en la figura 3 se presenta una disposición óptima de las piezas sobre el material, donde se observa que se utiliza la longitud mínima y además se evita el desperdicio del material. En las figuras 4, 5 y 6 se muestran los cortes tipo guillotina obtenidos, tanto horizontal como verticalmente a fin de generar las piezas demandadas.

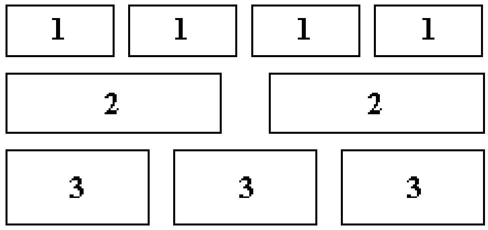

Figura 1. Ejemplo de piezas demandadas.

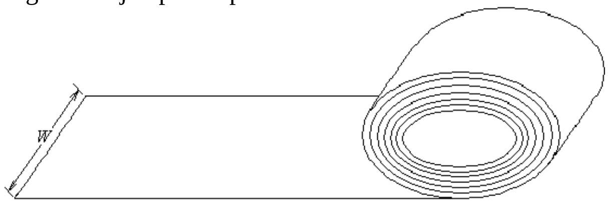

Figura 2. Ejemplo de rollo infinito con un ancho *W* fijo.

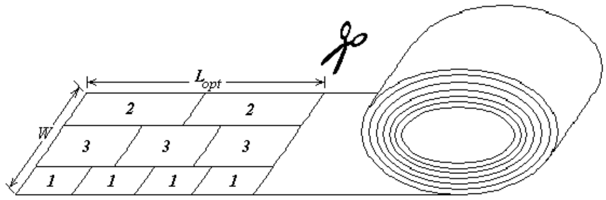

Figura 3. Ejemplo de empaquetamiento óptimo bidimensional en rollos infinitos.

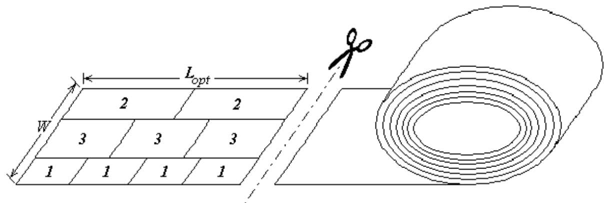

Figura 4. Corte del rollo.

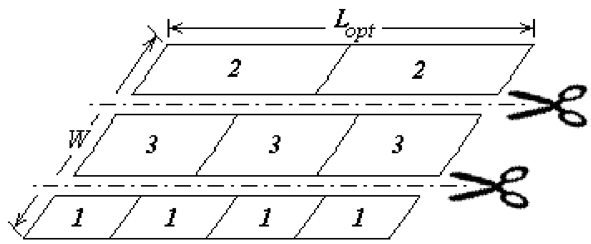

Figura 5. Cortes tipo guillotina para generar las piezas demandadas.

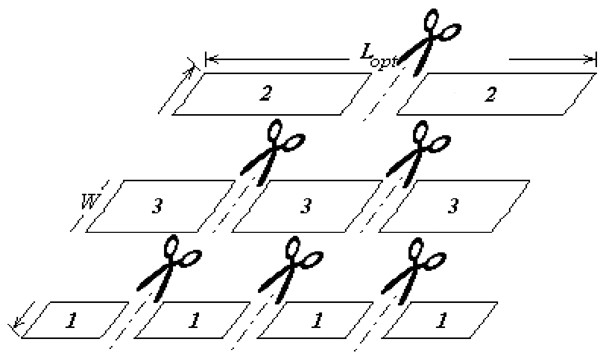

Figura 6. Cortes tipo guillotina para generar las piezas demandas.

3. MODELO MATEMÁTICO

Los problemas de empaquetamiento tienen un alta dificultad de ser formalizados, afortunadamente en [10] se desarrolla un modelo matemático lo bastante general, el cual es aceptado por la comunidad investigativa alrededor del tema. El modelo representa el problema empaquetamiento como un problema de programación lineal entera mixta, para este artículo el modelo es adaptado a fin de describir el problema de *strip packing*.

A continuación se describen algunas notaciones necesarias para comprender el modelo matemático del problema.

L : Largo de la hoja de material.
W : Ancho de la hoja de material.
N : Número disponible de rectángulos para posicionar en la hoja de material.
l_i : Largo del rectángulo i.
w_i : Ancho del rectángulo i.
(x_i, y_i) : Variables que indican la localización del rectángulo i teniendo como punto de referencia el vértice inferior izquierdo de la hoja de material.
s_i : Variable binaria que indica si el rectángulo i fue ubicado en la hoja de material. Cuando esto ocurre $s_i = 1$, en caso contrario $s_i = 0$.
Adicionalmente, existen otras variables que son usadas para indicar el posicionamiento de los rectángulos en relación a otros rectángulos.
a_{ik} : En caso de que sea 1, indica que el rectángulo i está a la izquierda del rectángulo k.
b_{ik} : En caso de que sea 1, indica que el rectángulo i está a la derecha del rectángulo k.
c_{ik} : En caso de que sea 1, indica que el rectángulo i está atrás del rectángulo k.
d_{ik} : En caso de que sea 1, indica que el rectángulo i está al frente del rectángulo k.
M : Número entero muy grande.

La función objetivo consiste en minimizar la cantidad de material desperdiciado tal como se expresa en la ecuación (1)

$$Min \quad L \tag{1}$$

Sujeto a

1. Evitar superposición de rectángulos en la hoja de material tal como se muestra en las ecuaciones (2), (3), (4) y (5).

$$x_i + l_i \leq x_k + (1 - a_{ik}) \cdot M, \forall i, k, i < k \tag{2}$$
$$x_k + l_k \leq x_i + (1 - b_{ik}) \cdot M, \forall i, k, i < k \tag{3}$$
$$y_i + w_i \leq y_k + (1 - c_{ik}) \cdot M, \forall i, k, i < k \tag{4}$$
$$y_k + w_k \leq y_i + (1 - d_{ik}) \cdot M, \forall i, k, i < k \tag{5}$$

2. Garantizar que el par de rectángulos evaluados con las ecuaciones anteriores estén dentro de la hoja de material, tal como se muestra en la ecuación (6).

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} \geq s_i + s_k - 1, \forall i, k, i < k \tag{6}$$

3. Garantizar que un posicionamiento de rectángulos, obedezca a las limitaciones físicas dadas por las dimensiones de la hoja de material, tal como se muestra en las ecuaciones (7) y (8).

$$x_i + l_i \leq L + (1 - s_i) \cdot M, \forall i \tag{7}$$
$$y_i + w_i \leq W + (1 - s_i) \cdot M, \forall i \tag{8}$$

Analizando el modelo resultante se tiene que el número de variables y de ecuaciones crece de forma exponencial con relación al número de piezas a empacar. La mayor preocupación que se observa respecto a este problema es el hecho de encontrarse en el peor escenario por ser un problema de programación lineal entera mixta. Dado lo anterior las técnicas de solución que más se adecuan para resolverlo son las metaheurísticas. En la siguiente sección se presenta una metodología para resolver el problema de empaquetamiento óptimo.

4. MÉTODO DE SOLUCIÓN

La restricción de cortes guillotina es garantizada bajo una estructura de árbol de cortes, este consiste en un árbol binario, donde cada nodo interno del árbol representa la forma en que se realiza el corte y los elementos a ubicarse en el subespacio generado por el corte [11]. La estructura de árbol de cortes se convierte en la codificación sobre la cual el algoritmo propuesto en este artículo trabajara.

El algoritmo propuesto tiene como fin convertir el problema de *strip packing* en pequeños problemas de *cutting stock* de forma tal que la suma de las longitudes de los *bins* generados sea el largo del *strip*. Un esquema general de la metodología es representada por la figura 7.

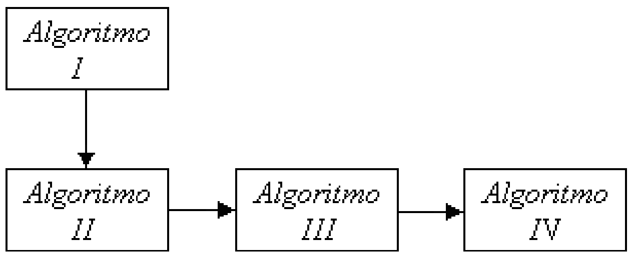

Figura 7. Esquema de la metodología

A continuación se presenta un resumen de la metodología propuesta.

4.1 Algoritmo de transformación s*trip-bins*

El algoritmo I consiste en determinar el número total de *bins* y las dimensiones de cada uno, donde luego se resuelve uno por uno como un problema de *cutting stock* a través de los algoritmos II, III y IV.
Las dimensiones de los *bins* están limitadas en todo momento a las restricciones tecnológicas. El procedimiento para calcular el número de *bins* se resume en los siguientes pasos y la ecuación (9):

1. Calcular la sumatoria de las áreas de las piezas demandas y dividir esta por el ancho *W* del *strip,* este resultado es llamado L_{opt}.
2. Definir l_{max} como la longitud de la pieza de mayor largor del conjunto de piezas demandadas.
3. Dividir el resultado del paso 1 entre l_{max}.

$$NúmeroBins = \frac{\dfrac{\sum_{i=1}^{N} l_i \cdot w_i}{W}}{l_{\max}} \qquad (9)$$

Teniendo el número y las dimensiones de los *bins* se resuelven como problemas de *cutting stock* secuenciales y dependientes, donde la demanda de piezas ira disminuyendo mientras evoluciona el proceso. Debido a que el largo del strip es un valor estimado, durante el proceso se aumentará en caso de que no se logre satisfacer la demanda.

4.2 Algoritmo de búsqueda aleatoria binaria

El algoritmo I determina los tipos de cada corte a utilizar sea horizontal o vertical, para este algoritmo se usa un esquema de vecindario variable [12]. Inicialmente el vecindario de búsqueda consiste en modificar de forma aleatoria la orientación de un corte, posteriormente, si esta modificación no genera una mejora en la función objetivo se procede a modificar dos cortes y así sucesivamente, en el momento en que la incumbente sea actualizada se regresa nuevamente al primer vecindario. Esta metodología de búsqueda local permite diversificar cuando la función objetivo no es mejorada y hacer una búsqueda local más detallada en el momento de actualizarse la incumbente, de esta forma se logra un algoritmo computacionalmente eficiente.

4.3 Algoritmo de búsqueda aleatoria real

El segundo algoritmo determina la distancia a la que se debe realizar cada corte, para ello se utiliza un algoritmo de búsqueda aleatoria que modifica iterativamente los cortes seleccionados por el algoritmo I reduciendo paulatinamente el ancho de búsqueda, mediante un factor de recocido que a medida que el algoritmo evoluciona se hace más determinístico por lo cual este factor se acerca a cero. El algoritmo II está inspirado en la filosofía del recocido simulado [13] en donde a medida que el proceso evoluciona el grado de aleatoriedad disminuye.

4.4 Algoritmo constructivo

El algoritmo constructivo ubica las piezas en los diferentes sub-espacios definidos por los tipos de los cortes y las distancias de estos. Cada sub-espacio ubica arreglos de la misma clase de piezas, para esto se determina el desperdicio para cada tipo de arreglo asegurando se cumpla la restricción de demanda, de tal forma que si el arreglo esta conformado por piezas duplicadas se modifica la demanda de la pieza original y la duplicada.

5. ANÁLISIS DE RESULTADOS

Para verificar la eficiencia de la metodología presentada en este artículo se tomo de la literatura especializada bases de datos con casos de prueba para el tipo de problema desarrollado en este trabajo. En la librería de Mhand Hifi [14] se presentan 34 casos de estudio de diferentes tamaños y complejidad matemática, los 25 primeros tienen entre 7 y 22 piezas con una complejidad media y los 9 restantes tienen entre 56 y 198 piezas con una alta complejidad.

El algoritmo fue implementado en DELPHI 7.0 ® en una computadora con unas especificaciones cercanas a un procesador Pentium ® 4 de 3.0 GHz y 1 GB de memoria RAM. Los parámetros usados para este estudio fueron: el número de iteraciones globales para el conjunto de problemas de complejidad media fue igual a 500 y para los de complejidad alta fue igual a 1000, el número de iteraciones parciales para los de complejidad media fue igual a 100 y para los de complejidad alta fue igual a 500, por ultimo el número de capas se ajusto dependiendo del número de piezas diferentes de cada instancia de prueba con relación a la ecuación (10).

$$NumerodeCapas = \lceil \log_2 NumerodePiezas \rceil \qquad (10)$$

En la tabla 1 se presentan las soluciones obtenidas por la metodología propuesta para el conjunto de problemas de complejidad media, además se ilustran las mejores respuestas reportadas en la literatura especializada [14].

Problema	Largo del Strip Reportado en la Literatura [15]	Largo del Strip Obtenido por la Metodología Propuesta
SCP1	13	13
SCP2	40	42,50
SCP3	14	15
SCP4	20	20
SCP5	20	20
SCP6	38	40,87
SCP7	14	12
SCP8	17	17
SCP9	68	76
SCP10	80	80
SCP11	48	55
SCP12	34	37
SCP13	50	47,51
SCP14	69	67
SCP15	34	35
SCP16	33	33
SCP17	39	35
SCP18	101	106,47
SCP19	26	26
SCP20	21	21
SCP21	145	139
SCP22	34	39
SCP23	35	32
SCP24	114	132
SCP25	36	36

Tabla 1. Resultados computacionales obtenidos comparados frente a la referencia [15].

En la tabla 2 se presentan las soluciones encontradas por la metodología propuesta para las instancias de alta complejidad, además se ilustran las mejores respuestas reportadas en la literatura especializada.

Problema	Largo del Strip Reportado en la Literatura [16]	Largo del Strip Obtenido por la Metodología Propuesta
SCPL1	57	62,08
SCPL2	165	154
SCPL3	181	179
SCPL4	67	74,26
SCPL5	147	146,88
SCPL6	24	27,73
SCPL7	126	130,97
SCPL8	77	84
SCPL9	102	101

Tabla 2. Resultados computacionales obtenidos comparados frente a la referencia [16].

En la tabla 3 se ilustra un resumen de la calidad de respuestas obtenidas por el algoritmo propuesto en todas los casos de estudio y comparadas con las reportadas en [15] y [16].

Resumen de Respuestas	Respecto a SCP1-SCP25	Respecto a SCPL1-SCPL9
Soluciones no superadas	10	5
Soluciones igualadas	9	0
Soluciones superadas	6	4
Número de casos	25	9

Tabla 3. Número de casos superados respecto a [15] y [16].

Se lograron identificar soluciones de muy alta calidad, obteniéndose mejores respuestas en algunas de las instancias que las reportadas en la literatura especializada. En los casos propuestos en la referencia [15] considerados de mediana complejidad matemática se iguala en 9, se supera en 6 y no se logra el objetivo en 10 instancias. En los casos propuestos en [16] considerados de alta complejidad matemática, se supera en 4 y no se logra el objetivo en 5 instancias.

6. CONCLUSIONES Y RECOMENDACIONES

Se resolvió el problema de empaquetamiento óptimo bidimensional en rollos infinitos, tipo guillotina sin rotación de piezas, mediante la transformación en pequeños problemas de bin packing y solucionado con un algoritmo híbrido constructivo de búsqueda en vecindad variable y recocido simulado, obteniendo resultados de buena calidad.

El tipo de codificación en árbol y el método de solución propuesto son adecuados en la solución de problemas de empaquetamiento de mercancías entre los que se contempla el strip packing.

Se lograron identificar soluciones de muy alta calidad, obteniéndose en algunas de las instancias una mejora respecto a las reportadas en la literatura especializada [14] y [15].

Como trabajos futuros se proponen incluir el concepto de rotación de piezas y realizar una extensión de la metodología a problemas de empaquetamiento óptimo tridimensional.

7. BIBLIOGRAFÍA

[1] A. Lodi, S. Martello S and D. Vigo, “Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems”, *INFORMS Journal of Computing*, vol. 11, 1999, pp. 345-57.

[2] JO. Berkey, PY. Wang, “Two-dimensional finite bin packing algorithms”, *Journal of the Operational Research Society*, vol. 38, 1987, pp. 423–429.

[3] S. Jakobs, “On genetic algorithms for the packing of polygons”, *European Journal of Operational Research*, vol. 88, 1996, pp. 165–81.

[4] CH. Dagli, P. Poshyanonda, “Newapproaches to nesting rectangular patterns”, *Journal of Intelligent Manufacturing*, vol. 8, 1997, pp. 177–190.

[5] E. Hopper, BCH. Turton, “An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem”, *European Journal of Operational Research*, vol. 128, 2001, pp. 34–57.

[6] G. Belov, *Problems, models and algorithms in one- and two-dimensional cutting*, PhD thesis, Technischen Universitat Dresden, 2003. [Online]. Available: http://www.math.tu-dresden.de/belov/publ/text030908_SUBMIT.pdf.

[7] A. Caprara, M. Monaci, “On the two-dimensional Knapsack Problem”, *Operations Research Letters*, vol. 32, 2004, pp. 5–14.

[8] SP. Fekete, J. Schepers, “On higher-dimensional packing III: exact algorithms”, *Operations Research*, Technical Report ZPR 97-290, 2004.

[9] D. Zhang, Y. Kang, A. Deng, “A new heuristic recursive algorithm for the strip rectangular packing problem”, *Computers and Operation research*, vol. 33, 2006, pp. 2209-2217.

[10] C. S. Chen, S. M. Lee, Q. S. Shen, “An analytical model for the container loading problem”, *European Journal of Operational Research*, vol. 80, 1995, pp. 68-76.

[11] K. Berthold, “Guillotineable bin packing: A genetic approach”, *European Journal of Operational Research*, vol. 84, 1995, pp. 645-661.

[12] P. Hansen, M. Nenad, J. Moreno, “Búsqueda de entorno variable”, *Revista Iberoamericana de Inteligencia Artificial*, vol. 19, 2003, pp. 77-92.

[13] R. Gallego, A. Escobar, E. Toro, *Técnicas Metaheurísticas de Optimización*, Universidad Tecnológica de Pereira, Julio 2008.

[14] M. Hifi, *Problem instances for the Strip Cutting/Packing Problem*, [Online]. Available: ftp://cermsem.univ-paris1.fr/pub/CERMSEM/hifi/Strip-cutting/

[15] M. Hifi, “Exact Algorithms for the Guillotine Strip Cutting/Packing Problem”, *Computers & Operations Research*, vol. 25, No 11, 1998, pp. 925-940.

[16] M. Hifi, “The Strip Cutting/Packing Problem: Incremental Substrip Algorithms-Based Heuristics”, *Pesquisa Operacional*, Special Issue on Cutting and Packing Problems, 1999, pp. 169-188.