

Problema de la mochila irrestricta bidimensional guillotizada¹

Unconstrained Two-Dimensional Knapsack Problem²

Problema da mochila irrestrita bidimensional guilhotizada³

David Álvarez-Martínez⁴

Eliana Mirledy Toro-Ocampo⁵

Ramón Alfonso Gallego-Rendón⁶

¹ Fecha de recepción: 24 de abril de 2010. Fecha de aceptación para publicación: 12 de agosto de 2010. Este artículo es resultado del proyecto de investigación *Solución del problema de empaquetamiento óptimo de contenedores*, financiado por Colciencias y la Universidad Tecnológica de Pereira, Colombia, a través del convenio Jóvenes Investigadores, con código de radicación CIEC 859-2009.

² Submitted on: April 24, 2010. Accepted on: August 12, 2010. This article results from the research project *Solution of the Tridimensional Bin Packing Problem*, financed by Colciencias and the Universidad Tecnológica de Pereira, Colombia, agreement Jovenes Investigadores, registration CIEC 859-2009.

³ Data de recepção: 24 de abril de 2010. Data de aceitação para publicação: 12 de agosto de 2010. Este artigo é resultado do projeto de pesquisa *Solução do problema de empacotamento ótimo de containers*, financiado por Colciencias e pela Universidade Tecnológica de Pereira, Colômbia, através do convenio Jovens Pesquisadores, com código de inscrição CIEC 859-2009.

⁴ Ingeniero de sistemas y computación. Magíster en Ingeniería Eléctrica, Universidad Tecnológica de Pereira, Pereira, Colombia. Joven Investigador Grupo DINOP-Universidad Tecnológica de Pereira. Correo electrónico: davidalv@utp.edu.co.

⁵ Ingeniera industrial. Magíster en Investigación de Operaciones y Estadística y en Ingeniería Eléctrica. Profesora de la Facultad de Ingeniería Industrial, Universidad Tecnológica de Pereira, Pereira, Colombia. Correo electrónico: elianam@utp.edu.co.

⁶ Ingeniero electricista. Magíster en Sistemas de Potencia. PhD en Ingeniería Eléctrica. Profesor de la Facultad de Ingenierías, Universidad Tecnológica de Pereira, Pereira, Colombia. Correo electrónico: ragr@utp.edu.co.

Resumen

Los problemas de empaquetamiento y corte óptimo son considerados clásicos dentro de la investigación de operaciones, debido a su gran espectro de aplicación en la industria y su alta complejidad tanto matemática como computacional. En este trabajo se presenta el problema de empaquetamiento óptimo bidimensional irrestricto de piezas rectangulares en una sola placa, con pesos asociados a las piezas y sin estos, al tiempo que se considera la posibilidad de rotar 90° las piezas y con restricciones de corte tipo guillotina (problema de la mochila bidimensional irrestricta guillotizada). Se describe el modelo matemático aplicado por diferentes grupos de investigación que estudian esta temática. Se propone, además, un tipo de codificación para aplicarla en este problema y se resuelve mediante un algoritmo de optimización que combina las principales características de cúmulo de partículas, recocido simulado y algoritmos genéticos. Para comprobar la eficiencia de la metodología presentada se tomaron casos de prueba de la literatura especializada, se analizan y comparan los métodos de solución presentados con los últimos avances del problema y se obtienen resultados de excelente calidad y nunca antes reportados en la literatura especializada.

Palabras clave

Mochila bidimensional irrestricta, cúmulo de partículas, recocido simulado.

Abstract

Cutting and packing problems are common in operations research, due to their big spectrum of application in the industry and its highly mathematical and computational complexity for the academy. In this study we present the unconstrained two-dimensional cutting stock problem of rectangular items, with and without weights associated to the items, bearing in mind the possibility to rotate items at 90° , and with guillotine cuts (also known as unconstrained two-dimensional guillotineable single knapsack problem). For this problem, we describe the mathematical model recognized by the academic community. We develop an appropriate encoding of the problem so it is possible to work on it using metaheuristic hybrid algorithm particle swarm optimization and simulated annealing. To check the efficiency of this methodology, case studies were taken from specialized literature, where the presented solution method could be analyzed and compared with current problems. The results obtained had an excellent quality and had never been reported.

Key words

Unconstrained two-dimensional knapsack problem, particle swarm optimization, simulated annealing.

Resumo

Os problemas de empacotamento e corte ótimo são considerados clássicos dentro da pesquisa de operações, devido a seu grande espectro de aplicação na indústria e sua alta complexidade tanto matemática como computacional. Neste trabalho apresenta-se o problema de empacotamento ótimo bidimensional irrestricto de peças retangulares em uma só placa, com pesos associados às peças e sem estes, ao mesmo tempo em que se considera a possibilidade de rotar 90° as peças e com restrições de corte tipo guilhotina (problema da mochila bidimensional irrestricta guilhotizada). Descreve-se o modelo matemático aplicado por diferentes grupos de pesquisa que estudam esta temática. Além disso, propõe-se também um tipo de codificação para aplicá-la neste problema e se resolve mediante um algoritmo de otimização que combina as principais características de cúmulo de partículas, recocimento simulado e algoritmos genéticos. Para comprovar a eficiência da metodologia apresentada se tomaron casos de prova da literatura especializada, analisam-se e comparam-se os métodos de solução apresentados com os últimos avanços do problema e se obtêm resultados de excelente qualidade e nunca antes relatados na literatura especializada.

Palavras chave

Mochila bidimensional irrestricta, particle swarm optimization, simulated annealing.

Introducción

El problema de la mochila irrestricta bidimensional guillotizada se usa para resolver problemas de corte que se presentan cuando el material es una pieza rectangular donde se deben ubicar piezas rectangulares más pequeñas de las que se conoce el tamaño y un costo asociado. El objetivo es maximizar el valor de las piezas cortadas. Las características de este problema son las siguientes:

- El costo asociado puede o no estar relacionado con el área de la pieza que se va a ubicar. Si el costo es igual al área de la pieza, se está resolviendo el problema sin pesos (*unweighted version*), y si el costo es diferente del área del ítem, el problema que se va a resolver es el problema con pesos (*weighted version*).
- La orientación de las piezas, es decir, una pieza de alto h y ancho w es diferente de una pieza de longitud w y alto h (*without rotation*). Si se considera que las dimensiones (b, w) y (w, b) representan las dimensiones de la misma pieza, se está abordando un problema con rotación (*with rotation*).
- Los patrones de corte son del tipo guillotina. Cada corte produce dos subrectángulos. Los cortes van de un extremo al otro del rectángulo original.
- No existe un límite máximo del número de piezas que se vayan a cortar de cada tipo (*unconstrained version*).

Gilmore y Gomory (1965 y 1966) proponen un algoritmo recursivo exacto sobre la base de la programación dinámica para resolver el problema. Este algoritmo es aplicable a las versiones con y sin pesos. Herz (1972) propone un método de búsqueda recursiva de árbol, que es más eficaz que el algoritmo de Gilmore y Gomory para el problema sin pesos, pero no se aplica a los casos con pesos. Entre tanto, Beasley (1985) plantea un algoritmo que es una versión modificada del algoritmo de Gilmore y Gomory. Hifi y Zissimopoulos (1996) sugieren un algoritmo recursivo exacto que use programación dinámica basada en eficientes límites inferiores y superiores para resolver el problema irrestricto.

Por último, G y Kang (2002) proponen una mejora al algoritmo de Hifi y Zisismpoulos usando una cota superior más eficiente. Este es actualmente uno de los algoritmos exactos con mejor desempeño para resolver el problema irrestricto.

Existen dos técnicas generales usadas para resolver los problemas restringidos: *top-down* y *bottom-up*. Christofides y Whitlock (1977) propusieron originalmente el enfoque de *top-down*; el otro requiere una gran cantidad de memoria, razón por la cual la implementación de un algoritmo de este tipo es poco atractivo. Sin embargo, G, Kang y Seong (2003) plantearon un algoritmo que parte de una solución inicial de buena calidad y utiliza el algoritmo constructivo *bottom-up* como estrategia para generar las ramas y así disminuir el número de nodos por explorar.

En este artículo se utiliza una codificación en árbol binario, llamado árbol de cortes, que convierte el problema de empaquetamiento en uno de optimización irrestricta y se resuelve mediante dos algoritmos. El primero es una búsqueda exhaustiva sobre el árbol para encontrar los tipos de corte (horizontal o vertical); el segundo, un algoritmo metaheurístico híbrido que combina las principales características de cúmulo de partículas, recocido simulado y algoritmos genéticos para determinar las distancias de corte óptimas.

Para comprobar la eficiencia y la calidad de las respuestas de la metodología propuesta se usaron casos de prueba de la literatura especializada, de donde se obtuvieron resultados de excelente calidad y de los cuales no se conoce reporte. La estructura de este artículo es la siguiente: descripción del problema, modelo matemático, metodología de solución, análisis de resultados y conclusiones.

1. Descripción del problema

El problema de la mochila irrestricta tiene dos variantes: sin pesos y con estos (*unweighted version* y *weighted version*). La primera instancia del problema (*weighted version with rotation*) se define como: cortar de un rectángulo que se denomina placa (objeto) de alto h y ancho w un conjunto de rectángulos de cardinalidad n que se denominan piezas (ítems) de alto h_i , ancho w_i y costo asociado c_i (donde $i=1, \dots, n$). Una pieza (h, w) es equivalente a una pieza (w, h) . Aquí el objetivo está dado por la ecuación (1) y consiste en maximizar el costo asociado del conjunto de piezas cortadas. Así, z_i es una variable binaria que indica si la pieza i fue o no cortada.

$$\max \sum_{i=1}^n c_i \cdot z_i \quad (1)$$

Sujeto a:

- Las piezas empacadas no deben superar los límites de la placa.
- Las piezas no deben superponerse entre ellas.
- Sólo está permitido el uso de cortes tipo guillotina.

La segunda instancia del problema (*weighted version without rotation*) sólo difiere en la definición anterior por la condición de orientación de las piezas; por lo tanto, una pieza (b, w) no es equivalente a una pieza (w, b) .

La tercera instancia del problema (*unweighted version with rotation*) únicamente difiere de la primera en que el costo asociado c_i es igual al alto h_i por el ancho w_i . De este modo, la función objetivo de la ecuación (2) consiste en maximizar el área del conjunto de piezas cortadas.

$$\max \sum_{i=1}^n h_i \cdot w_i \cdot z_i \quad (2)$$

La cuarta instancia del problema (*unweighted version without rotation*) sólo difiere de la definición anterior por la condición de orientación de las piezas; por lo tanto, una pieza (b, w) no es equivalente a una pieza (w, b) .

2. Modelo matemático

Se presenta un modelo de programación lineal entera mixta para el problema de empaquetamiento óptimo de la mochila bidimensional irrestricta basado en la caracterización de los patrones de corte tipo guillotina y el uso de coordenadas donde pueden ser ubicadas las piezas (Figura 1). Esta es una adaptación del modelo presentado por (Ben, Chu y Espinouse, 2008) para el problema de empaquetamiento óptimo guillotinado en rollos infinitos (*guillotinable strip packing problem*).

A fin de obtener un modelo lineal entero, se usa el siguiente conjunto de variables binarias que representan la ubicación de las piezas en la placa:

$$z_{i,j,k} = \begin{cases} 1, & \text{si la pieza } k \text{ es empacada en } (i, j) \\ 0, & \text{de lo contrario} \end{cases}$$

Las siguientes variables de decisión intermedia también son necesarias para garantizar que no existan traslapes entre piezas:

$$u_{i,j,i'} = \begin{cases} 1, & \text{si la pieza en } (i, j), \text{ no excede (horizontalmente) } x_{i'} \text{ con } i' > i \\ 0, & \text{de lo contrario} \end{cases}$$

$$v_{i,j,j'} = \begin{cases} 1, & \text{si la pieza en } (i, j), \text{ no excede (verticalmente) } y_{j'}, \text{ con } j' > j \\ 0, & \text{de lo contrario} \end{cases}$$

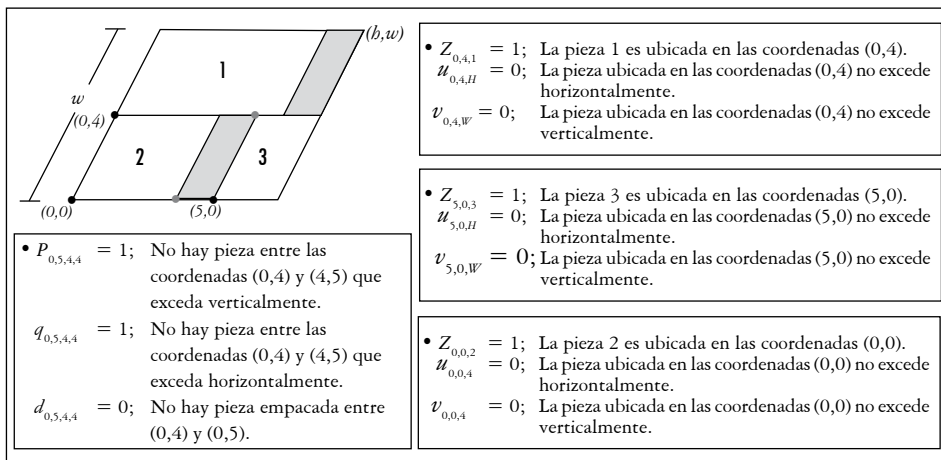
Los siguientes tres conjuntos de variables binarias son necesarios para garantizar las restricciones guillotina:

$$p_{i_1,i',j_1,j_2} = \begin{cases} 1, & \text{si no hay pieza entre } (i_1, j_1) \text{ y } (i'-1, j_2) \text{ que exceda } x_{j'} \text{ (consecuentemente,} \\ & \text{un corte vertical en } x_{j'} \text{ no cruza ninguna pieza empacada entre } (i_1, j_1) \text{ y} \\ & (i'-1, j_2)), i' > i_1 \\ 0, & \text{de lo contrario} \end{cases}$$

$$q_{i_1,i_2,j_1,j'} = \begin{cases} 1, & \text{si no hay pieza entre } (i_1, j_1) \text{ y } (i_2, j'-1) \text{ que exceda } y_{j'} \text{ (consecuentemente,} \\ & \text{un corte horizontal en } y_{j'} \text{ no cruza ninguna pieza empacada entre } (i_1, j_1) \text{ y} \\ & (i_2, j'-1)), j' > j_1 \\ 0, & \text{de lo contrario} \end{cases}$$

$$d_{i_1,i_2,j_1,j_2} = \begin{cases} 1, & \text{si existe mínimo una pieza empacada entre } (i_1, j_1) \text{ y } (i_2, j_2) \\ 0, & \text{de lo contrario} \end{cases}$$

Figura 1. Modelamiento y disposición de piezas sobre la mochila



Fuente: presentación propia de los autores.

La formulación completa del problema de empaquetamiento óptimo guillotinado de la mochila bidimensional irrestricta (*weight version without rotation*) es la siguiente:

$$\text{Maximizar } \sum_{i=1}^n \sum_{j=1}^n c_k z_{i,j,k} \quad \forall k, \tag{3}$$

$$\text{sujeto a } 0 \leq x_1 \leq x_2 \leq \dots \leq x_n, \tag{4}$$

$$0 \leq y_1 \leq y_2 \leq \dots \leq y_n, \tag{5}$$

$$\sum_{i=1}^n \sum_{j=1}^n z_{i,j,k} \leq 1 \quad \forall k, \tag{6}$$

$$\sum_{i=1}^n \sum_{k=1}^n z_{i,j,k} \leq 1 \quad \forall j, \tag{7}$$

$$\sum_{j=1}^n \sum_{k=1}^n z_{i,j,k} \leq 1 \quad \forall i, \tag{8}$$

$$x_{i'} - x_i - \sum_{k=1}^n w_k z_{i,j,k} \geq (u_{i,j,i'} - 1)W \quad \forall i, \forall j, \forall i' > i, \tag{9}$$

$$x_{i'} - x_i - \sum_{k=1}^n w_k z_{i,j,k} \geq u_{i,j,i'}W \quad \forall i, \forall j, \forall i' > i, \tag{10}$$

$$y_{j'} - y_j - \sum_{k=1}^n h_k z_{i,j,k} \geq (v_{i,j,j'} - 1)H \quad \forall i, \forall j, \forall j' > j, \tag{11}$$

$$y_{j'} - y_j - \sum_{k=1}^n h_k z_{i,j,k} \geq u_{i,j,j'}H \quad \forall i, \forall j, \forall j' > j, \tag{12}$$

$$(1 - d_{i_1, i_2, j_1, j_2})n \geq \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \sum_{k=1}^n z_{i,j,k} - 1 \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j_2 > j_1, \tag{13}$$

$$p_{i_1, i', j_1, j_2} \leq \sum_{j=j_1}^{j_2} \sum_{k=1}^n z_{i',j,k} \quad \forall i_1, \forall j_1, \forall j_2 > j_1, \forall i' > i_1, \tag{14}$$

$$p_{i_1, i', j_1, j_2} \leq \sum_{i=i_1}^{i'-1} \sum_{j=j_1}^{j_2} \sum_{k=1}^n z_{i,j,k} \quad \forall i_1, \forall j_1, \forall j_2 > j_1, \forall i' > i_1, \tag{15}$$

$$(i' - i_1)(j_2 - j_1 + 1)p_{i_1, i', j_1, j_2} \leq \sum_{i=i_1}^{i'-1} \sum_{j=j_1}^{j_2} u_{i,j,i'} \quad \forall i_1, \forall j_1, \forall j_2 > j_1, \forall i' > i_1, \tag{16}$$

$$q_{i_1, i_2, j_1, j'} \leq \sum_{i=i_1}^{i_2} \sum_{k=1}^n z_{i,j',k} \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j' > j_1, \tag{17}$$

$$q_{i_1, i_2, j_1, j'} \leq \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j'-1} \sum_{k=1}^n z_{i,j,k} \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j' > j_1, \tag{18}$$

$$(j' - j_1)(i_2 - i_1 + 1)q_{i_1, i_2, j_1, j'} \leq \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j'-1} v_{i,j,i'} \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j' > j_1, \tag{19}$$

$$d_{i_1, i_2, j_1, j_2} + \sum_{i'=i_1+1}^{i_2} p_{i_1, i', j_1, j_2} + \sum_{j'=j_1+1}^{j_2} q_{i_1, i_2, j_1, j'} \geq 1 \quad \forall i_1, \forall j_1, \forall i_2 > i_1, \forall j_2 > j_1, \tag{20}$$

$$W \geq x_i + \sum_{j=1}^n \sum_{k=1}^n w_k z_{i,j,k} \quad \forall i, \tag{21}$$

$$H \geq y_j + \sum_{i=1}^n \sum_{k=1}^n h_k z_{i,j,k} \quad \forall j, \quad (22)$$

$$z_{i,j,k} \in \{0,1\} \quad \forall i, \forall j, \forall k, \quad (23)$$

$$u_{i,j,i'} \in \{0,1\} \quad \forall i, \forall j, \forall i' > i, \quad (24)$$

$$v_{i,j,j'} \in \{0,1\} \quad \forall i, \forall j, \forall j' > j, \quad (25)$$

$$d_{i_1,i_2,j_1,j_2}, p_{i_1,i_2,j_1,j_2}, q_{i_1,i_2,j_1,j_2} \in \{0,1\} \quad \forall i, \forall j, \forall i_2 > i_1, \forall j_2 > j_1, \quad (26)$$

En la formulación anterior, las restricciones (6)-(8) aseguran que cada posición horizontal o vertical sea ocupada exactamente por una pieza, y cada pieza es empacada exactamente una vez.

Las restricciones (9)-(12) garantizan que no existan traslapes entre piezas. Las restricciones (13)-(20) son para cada área rectangular. Si las restricciones de guillotina son satisfechas para cada área rectangular, también se cumple para todo el patrón de corte. Finalmente, las restricciones (21) y (22) garantizan que ninguna pieza exceda horizontalmente el ancho w y que ninguna pieza exceda verticalmente la altura h , considerando que la función objetivo es maximizar la sumatoria del costo asociado de las piezas empacadas en la placa (véase ecuación [3]).

El modelo que representa la versión sin pesos (*unweight version*) del problema es igual al anterior, modificado únicamente en la función objetivo, es decir, la ecuación (3) es reemplazada por la ecuación (27).

$$\text{Maximizar} \quad \sum_{i=1}^n \sum_{j=1}^n w_k l_k z_{i,j,k} \quad \forall k \quad (27)$$

En este modelo existen muchas restricciones y variables binarias (cerca de $3n^4/4$ variables binarias y cerca de $2n^4$ restricciones). Por otra parte, un grupo grande de restricciones presenta un comportamiento donde sólo una restricción está activa y las otras son redundantes (por ejemplo, existe siempre una restricción redundante entre (9) y (10)).

Por esta razón, la relajación de la programación lineal (PL) del modelo obtiene un límite inferior de mala calidad. El modelo presenta una alta complejidad matemática que en la práctica lo hace inexplotable mediante el *software* y el *hardware* actual disponible para la programación entera mixta (más detalle de este modelo en [Ben, Chu y Espinouse, 2008]).

3. Metodología de solución

3.1 Codificación

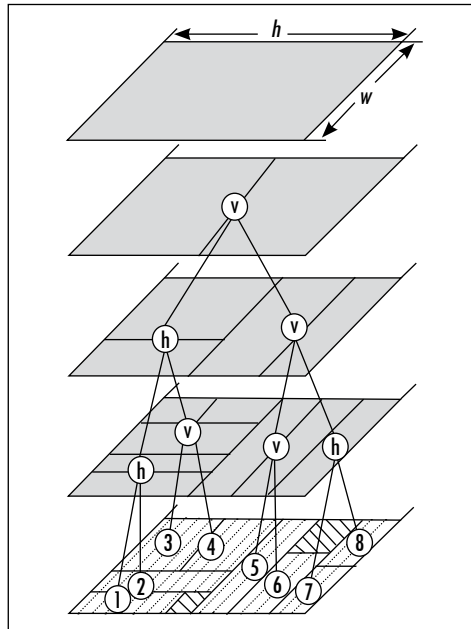
Wong, Leong y Liu (1988) presentan una estructura de codificación de datos para el problema de empaquetamiento, denominada *árbol de cortes*. Una de las grandes ventajas de la representación en árbol de cortes es la generación de patrones de corte tipo guillotina. Diferentes metodologías propuestas han corroborado la efectividad de la codificación en árbol de cortes, en especial la presentada por (Cui, 2007 y Toro, Garcés y Ruiz, 2008).

Un árbol de cortes se define como un árbol con raíz, donde cada nodo interno (padre) representa la posición y la forma como se realiza el corte sobre el material (horizontal o vertical); mientras los nodos hoja (nodos terminales) representan las dimensiones de los subespacios generados para cortar las piezas agrupadas.

Diferentes propuestas que usan la codificación en árbol de cortes sugieren encontrar el árbol óptimo, que es de difícil solución, a diferencia de (Toro, Garcés y Ruiz, 2008), que delimitan y reducen el número de árboles en el proceso de optimización. A la vez, Toro, Garcés y Ruiz (2008) sugieren, después de realizar un estudio estadístico, delimitar el árbol de cortes al uso de árboles binarios completos con tres niveles.

La Figura 2 ilustra un árbol para el problema de empaquetamiento. En este, el nodo raíz establece un corte vertical sobre la placa de material. La jerarquía del árbol indica que el hijo izquierdo efectúa un corte horizontal sobre el rectángulo izquierdo, mientras el hijo derecho realiza un nuevo corte vertical sobre el rectángulo derecho; esto para el primer nivel. En el segundo nivel se realiza un corte horizontal al rectángulo inferior izquierdo y se obtienen para el tercer nivel dos subespacios 1 y 2, mientras que en el rectángulo superior izquierdo se efectúa un corte vertical y se logran en el tercer nivel los subespacios 3 y 4. De la misma forma, los subespacios 5, 6, 7 y 8 son el resultado de los cortes realizados en el segundo nivel. En cada subespacio generado se ubican las piezas con igual forma que generen mayor rentabilidad.

Figura 2. Representación mediante árbol de cortes



Fuente: presentación propia de los autores.

3.2 Cálculo de la función objetivo

Luego de obtener los subespacios, se deben ubicar las piezas. Este proceso se realiza mediante el algoritmo constructivo de mejor ajuste (*best-fit*), conservando las restricciones tipo guillotina y embalando la mayor cantidad de piezas por subespacio.

El algoritmo constructivo *best-fit* consiste en encontrar el conjunto de piezas idénticas que maximice el área del subespacio j y, a su vez, genere la mayor rentabilidad, sujeto a la restricción del número de piezas i disponibles. La ecuación (28) expresa formalmente el algoritmo *best-fit*. El cálculo de la función objetivo consiste en calcular la ecuación (28) para cada subespacio.

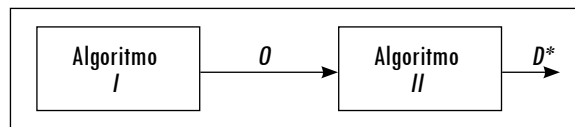
$$\max \left\{ c_i \cdot \min \left(\left\lfloor \frac{W_j}{w_i} \right\rfloor \cdot \left\lfloor \frac{L_j}{l_i} \right\rfloor, \text{piezas disponibles tipo } i \right) \cdot b_i; \forall i; i = 1, 2, \dots, n \right\} \quad (28)$$

3.3 Metodología de optimización

La codificación propuesta en este estudio garantiza la factibilidad, en cuanto a las restricciones de corte tipo guillotina. El árbol de orientación de cortes y el árbol de distancias de los cortes son independientes entre sí (representados por las variables O y D). Esto significa que para cada conjunto de valores de O existe una

solución óptima D^* . El esquema de optimización para los problemas de mochila se ilustra en la Figura 3. El algoritmo *I* realiza una búsqueda exhaustiva sobre el árbol O , mientras que el algoritmo *II* recibe todos árboles O posibles, donde debe encontrar las distancias óptimas para cada uno de estos. El algoritmo *II* corresponde al escenario de las técnicas metaheurísticas.

Figura 3. Esquema de optimización



Fuente: presentación propia de los autores.

El algoritmo *I* genera los posibles árboles de orientación de los cortes. Los problemas de la mochila se restringen a sólo árboles completos de tres niveles. El resultado de este algoritmo siempre dará como resultado 128 diferentes árboles de orientación, y cada uno de éstos se usa como dato de entrada para el algoritmo *II*.

En el algoritmo *II* se usan las técnicas metaheurísticas de optimización. Este combina las principales características de cúmulo de partículas, recocido simulado y algoritmos genéticos. El primero es el algoritmo principal, el segundo y tercero se utilizan como mecanismo de perturbación especializado para realizar búsquedas locales, a fin de efectuar cambios en las posiciones de las partículas usando la filosofía de la mutación de los genéticos y la temperatura del recocido simulado.

Así mismo, en este algoritmo se incluye el operador de mutación propio de los algoritmos genéticos (Gallego, Escobar y Toro, 2008) en el algoritmo de cúmulo de partículas, donde la mutación se define como la modificación del valor de un nodo del árbol de distancias a través del mecanismo de transición de la ecuación (29).

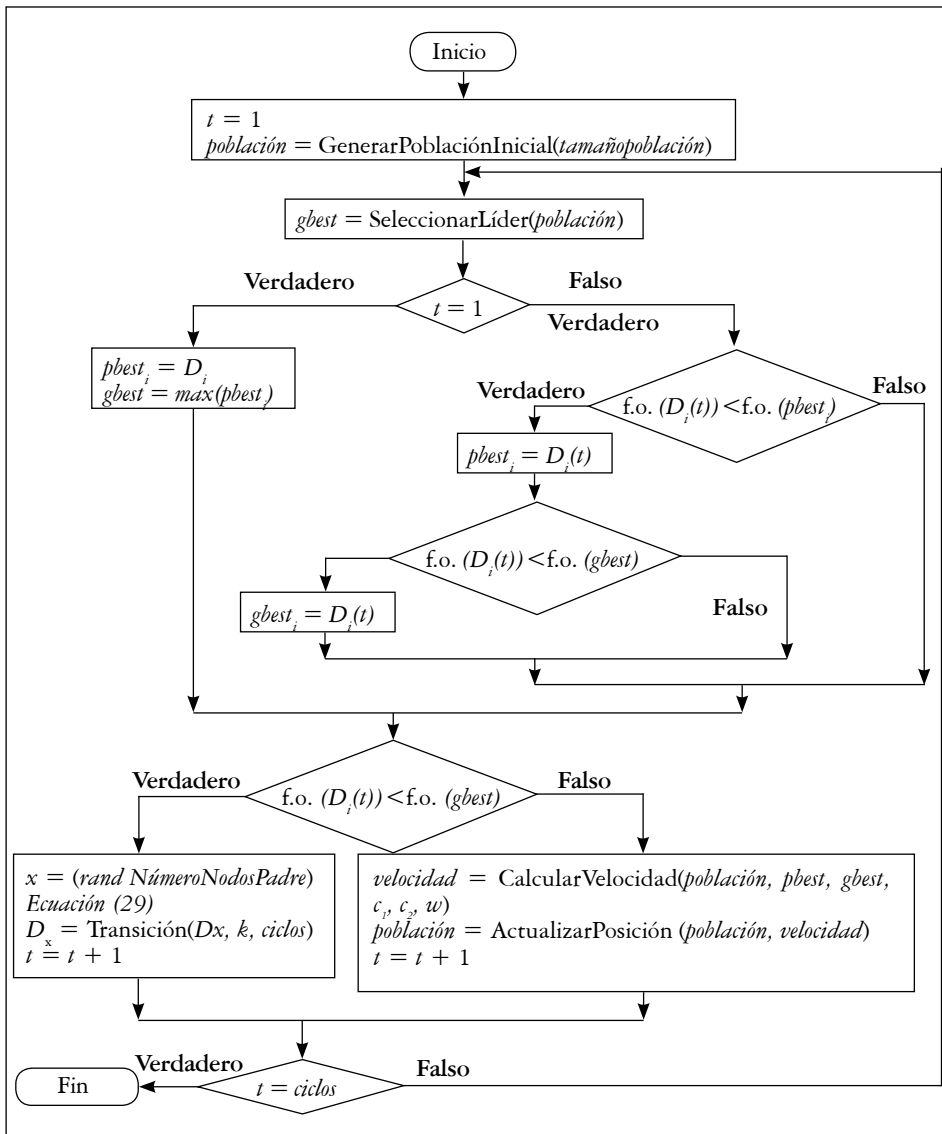
$$\text{nodo } i = \text{nodo } i + \left(\text{rand} - \frac{1}{2} \right) \cdot \left(\sqrt{1 - \frac{k}{\text{Iteraciones totales}}} + \varepsilon \right) \quad (29)$$

El mecanismo de transición consiste en permitir grandes cambios en las distancias de los cortes durante las primeras iteraciones, al igual que el recocido simulado permite empeoramientos de la función objetivo al comienzo del proceso. A medida que avanzan las iteraciones, los cambios se vuelven más sensibles y determinísticos. La ecuación (29) está compuesta por el valor actual del nodo i del árbol de distancias, el número de iteración actual, el número de

iteraciones totales y un ϵ (donde ϵ es el mínimo porcentaje para generar un cambio en la distancias) y donde, $\epsilon = 100/\max(L, W)$.

La Figura 4 ilustra el diagrama de flujo de datos del algoritmo de cúmulo de partículas más el recocido simulado ($A_{\text{PSO+SA}}$). El nombre de este algoritmo está dado, ya que utiliza el mecanismo de transición inspirado en la técnica de recocido simulado.

Figura 4. Algoritmo híbrido de cúmulos de partículas y recocido simulado



Fuente: presentación propia de los autores.

3.4 Calibración de parámetros

En general, no existe un método exacto y eficiente para calibrar parámetros de las diferentes técnicas metaheurísticas. Comúnmente estos algoritmos son parametrizados combinando una búsqueda exhaustiva y un análisis estadístico de la calidad de los resultados. Zhi-Hui, Jun y Yun (2009) presentan un rango de valores reducido para los parámetros del algoritmo de cúmulo de partículas. Teniendo como base los rangos presentados por los anteriores autores, el tamaño de la malla se reduce considerablemente.

En este estudio se conserva la filosofía de los operadores de mutación de los algoritmos genéticos, donde la probabilidad de que ocurra una mutación en la población es muy baja. Entonces se realizó el mismo proceso de calibración para los rangos propuestos del parámetro de mutación en (Gallego, Escobar y Toro, 2008). Los valores resultantes de la calibración de parámetros son ilustrados en la Tabla 1.

Tabla 1. Valores de parámetros del algoritmo A_{PSO+SA}

Parámetros	Valor
Tamaño de la población	100
Número de ciclos	500
c_1 (conocimiento individual)	2,05
c_2 (conocimiento grupal)	2,05
w (inercia)	0,70
Tasa de mutación	0,03
Número de niveles	3

Fuente: presentación propia de los autores.

4. Análisis de resultados

Los sistemas de prueba usados en este estudio se tomaron de la literatura especializada, pues metodologías aproximadas y exactas se emplean en la solución de dichos problemas. Los problemas seleccionados son variados, en cuanto a la complejidad matemática y son diseñados especialmente para cada tipo de problema.

Fueron seleccionados 30 casos de prueba para el problema de la mochila bidimensional guillotizada, 15 para la versión con pesos ({UW1-UW11} y {UWL1-UWL4}) y 15 para la versión sin pesos ({UU1-UU11} y {UUL1-UUL4}). Estos casos presentan 30 tipos de mochilas diferentes con distribuciones entre 25 y

200 piezas. La base de datos es presentada por (Hifi, 2001) y disponible en línea en (Hifi, 1997). De la totalidad de casos, 22 ({UW1-UW11} y {UU1-UU11}) pertenecen a la categoría de problemas de empaquetamiento de mediana complejidad matemática, y ocho ({UWL1-UWL4} y {UUL1-UUL4}), a la de alta complejidad matemática. Diferentes estudios han utilizado estos casos de prueba para realizar una especie de *benchmark* de las metodologías propuestas.

Todos los algoritmos fueron desarrollados en Delphi 7.0[®] y ejecutados en un computador con un procesador Pentium IV de 3 GHz y una memoria RAM de una giga. Se presentan a continuación para todos los casos de prueba de cada tipo de problema la mejor solución reportada en la literatura especializada (*best known solution*). Los resultados de (Hifi, 1998) presentan las mejores soluciones, aunque trabajos como los de (G, Kang y Seong, 2003) alcanzan buenos resultados.

Casos como UWL1, UWL2, UWL3, UWL4, UUL1, UUL2, UUL3 y UUL4 para el problema de la mochila sin rotación no tienen respuesta reportada, debido a que la mayoría de aproximaciones a estos casos de gran complejidad matemática se hicieron mediante el uso de técnicas exactas (Gallego, Escobar y Romero, 2007), donde el esfuerzo computacional es demasiado alto para su solución. Por lo tanto, algunos autores omiten reportarlas. Las tablas 2 y 3 presentan los resultados obtenidos por el algoritmo propuesto en este estudio y los tiempos empleados para alcanzar estas respuestas.

Tabla 2. Resultados A_{PSO+SA} para la mochila con pesos, con rotación y sin esta

Caso	Mejor solución reportada		Tiempo utilizado (segundos)		Solución propuesta A_{PSO+SA}		Tiempo utilizado (segundos)	
	Sin rotación	Con rotación	Sin rotación	Con rotación	Sin rotación	Con rotación	Sin rotación	Con rotación
UW1	6036	6696	13	29	6036	6696	33	65
UW11	15747	18200	84	138	15747	18200	37	72
UW2	8468	9732	53	124	8468	9732	45	88
UW3	6226	7188	32	111	6226	7188	56	100
UW4	8326	8452	135	4870	8326	8452	59	116
UW5	7780	8398	58	196	7780	8398	45	89
UW6	6615	6937	177	635	6615	6937	59	116
UW7	10464	11585	301	1104	10464	11585	62	122

Continúa

Caso	Mejor solución reportada		Tiempo utilizado (segundos)		Solución propuesta A_{PSO+SA}		Tiempo utilizado (segundos)	
	Sin rotación	Con rotación	Sin rotación	Con rotación	Sin rotación	Con rotación	Sin rotación	Con rotación
UW8	7692	8088	452	921	7692	8088	69	136
UW9	7038	7527	258	1570	7038	7527	56	110
UW10	7507	8172	932	1283	7507	8172	74	147
UWL1	...	171079321	...	913	158363374	171836653	155	302
UWL2	...	325725070	...	1281	257372669	326399541	158	299
UWL3	...	433859655	...	3095	261252077	433882185	320	610
UWL4	...	568436545	...	6017	473754165	568690356	450	700

Fuente: presentación propia de los autores.

Tabla 3. Resultados A_{PSO+SA} para la mochila sin pesos, con rotación y sin esta

Caso	Mejor solución reportada $H \times W$		Tiempo utilizado (segundos)		Solución A_{PSO+SA} $H \cdot W$		Tiempo utilizado (segundos)	
	Sin rotación	Con rotación	Sin rotación	Con rotación	Sin rotación	Con rotación	Sin rotación	Con rotación
UU1	97,17	98,42	372	17	97,17	98,42	106	210
UU2	99,21	99,28	13	49	99,21	99,28	120	238
UU3	97,52	99,03	19	3437	96,82	99,03	106	210
UU4	98,25	99,05	67	186	97,39	98,97	145	288
UU5	99,15	99,64	106	352	99,15	99,64	186	374
UU6	98,79	98,81	43	263	98,79	98,81	145	288
UU7	98,84	99,46	255	>10800	98,61	98,67	186	374
UU8	98,98	99,37	168	444	98,98	98,98	212	422
UU9	99,20	99,36	160	>10800	99,20	99,20	231	460
UU10	99,01	99,35	>10800	2239	98,30	99,16	212	422
UU11	99,85	99,88	>10800	>10800	99,73	99,86	105	210
UUL1		99,99		704	99,95	99,99	280	500
UUL2		99,99		1336	99,96	99,99	282	503

Continúa

Caso	Mejor solución reportada $H \times W$		Tiempo utilizado (segundos)		$\frac{\text{Solución } A_{PSO+SA}}{H \cdot W}$		Tiempo utilizado (segundos)	
	Sin rotación	Con rotación	Sin rotación	Con rotación	Sin rotación	Con rotación	Sin rotación	Con rotación
UUL3		99,99		4165	99,98	99,99	320	604
UUL4		98,42		6170	99,98	98,42	420	750

Fuente: presentación propia de los autores.

Cuando se soluciona el problema de la mochila irrestricta con pesos, con rotación de piezas y sin estas para instancias de mediana complejidad matemática usando el algoritmo propuesto, se alcanzan respuestas de excelente calidad y en tiempos computacionalmente razonables. Para problemas de gran complejidad se mejoran ostensiblemente las respuestas reportadas en la literatura. Sin embargo, cuando se soluciona el problema de la mochila irrestricta sin pesos usando el algoritmo propuesto, se obtiene un comportamiento regular, en cuanto a las respuestas encontradas por la literatura para instancias de mediana complejidad; entre tanto, para problemas de gran complejidad las repuestas son de mejor calidad respecto a las reportadas en la literatura (Tabla 4).

Tabla 4. Comparación de resultados del algoritmo A_{PSO+SA} respecto a lo que se conoce

	Con pesos				Sin pesos			
	Sin rotación		Con rotación		Sin rotación		Con rotación	
	Calidad	Tiempo	Calidad	Tiempo	Calidad	Tiempo	Calidad	Tiempo
Igualados	11		11		6		9	
Superados	4	13	4	14	4	8		11
Inferiores		2		1	5	7	6	4

Fuente: presentación propia de los autores.

5. Conclusiones

Se resolvió el problema de empaquetamiento óptimo bidimensional guillotinado irrestricto con pesos y sin estos, con rotación de las piezas y sin esta mediante un algoritmo híbrido de cúmulo de partículas, recocido simulado y algoritmo genético, y con ello se obtuvieron excelentes resultados.

Se utilizó un tipo de codificación en árbol, llamada árbol de cortes, que combina un árbol de valores binarios, en el cual se orientan los cortes, con un árbol de valores reales y el cual determina las distancias de los cortes. Esto presenta un gran desempeño para este tipo de problemas.

Además, se implementó un algoritmo de optimización que combina las principales características de cúmulo de partículas, recocido simulado y algoritmos genéticos. El primero se considera el algoritmo principal y el segundo y tercero se usan como mecanismo de perturbación especializado para realizar búsquedas locales, a fin de efectuar cambios en las posiciones de las partículas usando la filosofía de la mutación de los genéticos y la temperatura del recocido simulado.

El método de solución usado presentó un gran desempeño en la solución del problema que asume pesos en la función objetivo y que considera dos variantes en la orientación de las piezas. En la primera instancia no permite rotación de piezas y en la segunda permite su rotación. En la solución del problema que no asume pesos en la función objetivo, su desempeño es aceptable, ya que la tercera parte de las instancias no fueron igualadas o superadas.

Los tiempos de cómputo son menores a los reportados en la literatura especializada; pero debido a las diferencias entre las arquitecturas de cómputo y lenguajes de programación, no se puede concluir entre metodologías. En particular, los tiempos de cómputo utilizados por la metodología propuesta son razonables.

Este problema es aplicable en diversos sectores de la economía, entre los que destacan los sectores de industria, transporte y comercio. Así, por ejemplo, su aplicación se presenta en las industrias textil, metalmecánica, papelera, vidriera, transporte y almacenaje de mercancías.

La metodología propuesta es aplicable a problemas como la mochila restringida, *bin packing*, *strip packing*, entre otros. También puede ser extendible a problemas de empaquetamiento en tres dimensiones.

Referencias

- BEASLEY, J. E. Algorithms for unconstrained two-dimensional guillotine cutting. *Journal of the Operational Research Society*, 1985, vol. 36, pp. 297-306.
- BEN, S.; CHU, C. y ESPINOUSE, M. L. Characterization and modelling of guillotine constraints. *European Journal of Operational Research*, 2008, vol. 191, pp. 112-126.
- CHRISTOFIDES, N. y WHITLOCK, C. An algorithm for two-dimensional cutting problems. *Operations Research*, 1977, vol. 25, pp. 30-44.
- CUI, Y. An exact algorithm for generating homogenous T-shape cutting patterns. *Computers & Operations Research*, 2007, vol. 34, pp. 1107-1120.

- GALLEGO, R.; ESCOBAR, A. H. y ROMERO, R. A. *Programación lineal entera*. Pereira: Universidad Tecnológica de Pereira, 2007.
- GALLEGO, R.; ESCOBAR, A. H. y TORO, E. M. *Técnicas metaheurísticas de optimización*. Pereira: Universidad Tecnológica de Pereira, 2008.
- G, Y.-G. y KANG, M.-K. A new upper bound for unconstrained two-dimensional cutting and packing. *Journal of the Operational Research Society*, 2002, vol. 53, pp. 587-591.
- G, Y.-G.; KANG, M.-K. y SEONG, J. A best-first branch and bound algorithm for unconstrained two-dimensional cutting problems. *Operations Research Letters*, 2003, vol. 31, pp. 301-307.
- GILMORE, P. C. y GOMORY, R. E. Multistage cutting problems of two and more dimensions. *Operations Research*, 1965, vol. 13, pp. 94-120.
- GILMORE, P. C. y GOMORY, R. E. The theory and computation of knapsack functions. *Operational Research*, 1966, vol. 14, pp. 1045-1074.
- HERZ, J. C. A recursive computing procedure for two-dimensional stock cutting. *IBM Journal of Research and Development*, 1972, vol. 16, pp. 462-469.
- HIFI, M. Exact algorithms for the guillotine strip cutting/packing problem. *Computers & Operations Research*, 1998, vol. 25, pp. 925-940.
- HIFI, M. Exact algorithms for large-scale unconstrained two and three staged cutting problems. *Computational Optimization and Applications*, 2001, vol. 18, pp. 63-88.
- HIFI, M. *Problem instances for the 2D Cutting/Packing Problems* [web en línea], 1997. <ftp://cermse.univ-paris1.fr/pub/CERMSEM/hifi/2Dcutting/> [Consultado 07-09-2010].
- HIFI, M y ZISSIMOPOULOS, V. A recursive exact algorithm for weighted two-dimensional cutting. *European Journal of Operational Research*, 1996, vol. 91, pp. 553-564.
- TORO, E.; GARCÉS, A. y RUIZ, H. Solución al problema de empaquetamiento bidimensional usando un algoritmo híbrido constructivo de búsqueda en vecindad variable y recocido simulado. *Revista Facultad de Ingeniería de la Universidad de Antioquia*, 2008, vol. 46, pp. 119-131.
- WONG, D. F.; LEONG, H. W. y LIU, C. L. *Simulated annealing for VLSI design*. New York: Kluwer Academic Publishers, 1988.
- ZHI-HUI, Z.; JUN, Z.; YUN, L. y HENRY, S. Adaptive Particle swarm optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 2009, vol. 39, pp. 1362-1381.