

## ALGORITMO DE OPTIMIZACIÓN CÚMULO DE PARTÍCULAS APLICADO EN LA SOLUCIÓN DEL PROBLEMA DE EMPAQUETAMIENTO ÓPTIMO BIDIMENSIONAL CON Y SIN ROTACIÓN

*Particle Swarm Optimization algorithm for the two-dimensional rectangular guillotine bin packing problem with and without rotations*

### RESUMEN

En este documento se presenta la técnica de optimización cúmulo de partículas la cual utiliza una propuesta de codificación basada en árbol binario de cortes para resolver el problema de corte bidimensional de piezas rectangulares. Este es un problema clásico de optimización caracterizado por su alta complejidad computacional y de gran aplicación en el área de la producción. En este artículo se trataron dos de sus variantes, denominados: RG y OG 2D-BPP. Con el fin de evaluar el desempeño del algoritmo se usaron sistemas de prueba de la literatura especializada obteniéndose excelentes resultados.

**PALABRAS CLAVES:** cúmulo de partículas, empaquetamiento bidimensional, optimización combinatorial.

### ABSTRACT

*In this document we present the algorithm Particle Swarm Optimization, using an encoding based in binary trees of cutting to solve the two-dimensional rectangular guillotine bin packing problem.*

*This is a well known classic optimization problem for its high computational complexity and its big applications in the field of production. We consider the problem with and without 90° rotations, called RG and OG 2D-BPP. The computational results on a kind of benchmark problems indicate that the algorithm performs better than several recently published algorithms.*

**KEYWORDS:** combinatorial optimization, particle swarm optimization, two dimensional bin packing.

### 1. INTRODUCCIÓN

Los problemas de corte y empaquetamiento, están estrechamente relacionados, es así como Vendramini [1] resuelve el problema de cargue de contenedores relajándolo inicialmente a un problema bidimensional y luego construyendo columnas sobre el patrón obtenido para llevarlo a tres dimensiones. Adicionalmente en [2] se muestra como el problema de embalaje en estibas o *pallets* se puede reducir a uno de dos dimensiones, este problema es habitual en las fábricas donde se reciben o entregan cajas de diferentes dimensiones.

Para resolver el problema la mayor parte de procedimientos reportados en la literatura se basan en técnicas metaheurísticas. En 1977 Christofides y Withlock [3], propusieron un algoritmo exacto de búsqueda en árbol para resolverlo, utilizando para ello el algoritmo previamente propuesto en 1966 por Gilmore y Gomory [4] quienes resuelven un problema de

### 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

características similares. Por otro lado en 1983 Wang [5], propone un algoritmo de desarrollo incremental. Tal algoritmo es posteriormente mejorado en los trabajos de Vasko [6] en 1989, Oliveira y Ferreira [7] en 1990. En 1997, Lai y Chan [8] presentan un procedimiento basado en *Simulated Annealing*.

En 1998 Parada [9] presenta la solución del problema mediante el *Simulated Annealing* usando una codificación de árbol binario. En 2001 Leung [10] utiliza un algoritmo evolutivo en el que usa la representación de las soluciones de Lai y Chan. En 2003 Beasley [11] presenta un algoritmo genético para el caso general, basado en una nueva formulación no lineal del problema. La formulación también admite extensiones para el problema con más de un tablero, el problema con algunas zonas del tablero que no se pueden utilizar y el problema donde las piezas se pueden rotar.

En 2007 Yaodong Cui [12] presenta un algoritmo exacto que genera cortes homogéneos en dos segmentos. Allí se

reportan 50 casos de prueba que son utilizados en este artículo y con los que se está validando la metodología propuesta.

En este artículo se está proponiendo la técnica de optimización cúmulo de partículas para resolver el problema de empaquetamiento óptimo bidimensional donde son conocidos el ancho y el alto del tablero donde se deben ubicar rectángulos de menor tamaño considerando dos opciones: posibilidad de rotación de las piezas y sin permitir la rotación de las piezas.

La presentación de este documento es como sigue: En la siguiente sección se hace una descripción del problema, en la sección 3 se presenta el modelo matemático, en la sección 4 se presenta el algoritmo de solución, en la sección 5 se presentan los resultados, finalmente, se plantean las conclusiones y recomendaciones.

## 2. DESCRIPCION

El problema de empaquetamiento óptimo bidimensional en contenedores (2D-BPP) del inglés *two-dimensional bin packing problem*, considera un conjunto de  $N$  piezas rectangulares, cada pieza  $i$  tiene un ancho  $q_i$  y un largo  $p_i$  (donde  $i \in \{1, \dots, N\}$ ). Este problema consiste en empacar las piezas en hojas de material de ancho  $W$  y longitud  $L$ . Sin pérdida de la generalidad, se asume que solo se cuenta con una hoja de material y las dimensiones de las piezas y la hoja de material son enteras. El objetivo es maximizar el área total usada por las piezas sin superponer los ítems. Debido a las restricciones tecnológicas de corte o empaquetamiento se presentan dos tipos de patrones: guillotina y no guillotina. Un corte es de tipo guillotina si cuando se aplica sobre un rectángulo produce dos nuevos rectángulos, es decir, si el corte va de un extremo a otro del rectángulo original; en otro caso se denomina del tipo no guillotina. Este problema se clasifica por los patrones de corte y las restricciones de orientación de los ítems:

RF: Las piezas a ser ubicadas pueden ser rotadas  $90^\circ$  (R) y no se requiere que el corte sea del tipo guillotina (F).  
 RG: Las piezas a ser ubicadas pueden ser rotadas  $90^\circ$  (R) y se requiere que el corte sea del tipo guillotina (G).  
 OF: Las 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).

En este trabajo se consideran los problemas de empaquetamiento óptimo bidimensional en contenedores (cortes) tipo guillotina con y sin rotación de piezas, reconocidos en la literatura como RG y OG 2D-BPP.

## 3. MODELO MATEMÁTICO

Los problemas de empaquetamiento tienen una alta dificultad de ser formalizados. En [13] se desarrolla un

modelo matemático general, el cual es aceptado por la comunidad académica que investigativa el tema. El modelo representa el problema de empaquetamiento como un problema de programación lineal entera mixta. En este artículo el modelo es adaptado a fin de describir el problema de *bin packing*.

A continuación se describen algunas notaciones necesarias para comprender los modelos matemáticos de los problemas trabajados en este artículo.

$N$ : Número disponible de rectángulos para posicionar en la hoja de material.

$p_i$ : Largo del rectángulo  $i$ .

$q_i$ : 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  $s_i=0$ .

$(l_{xi}, l_{yi})$ : Variables binarias que indican el eje de la hoja de material, el lado  $p_i$  del rectángulo  $i$  está en paralelo.

$(w_{xi}, w_{yi})$ : Variables binarias que indican el eje de la hoja de material, el lado  $q_i$  del rectángulo  $i$  está en paralelo.

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).

$$\text{Min } L \cdot W - \sum_{i=1}^N p_i \cdot q_i \cdot s_i \quad (1)$$

Sujeto a:

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

$$x_i + p_i \cdot l_{xi} + q_i w_{xi} \leq x_k + (1 - a_{ik}) \cdot M, \forall i, k, i < k \quad (2)$$

$$x_k + p_k \cdot l_{xk} + q_k w_{xk} \leq x_i + (1 - b_{ik}) \cdot M, \forall i, k, i < k \quad (3)$$

$$y_i + p_i \cdot l_{yi} + q_i w_{yi} \leq y_k + (1 - c_{ik}) \cdot M, \forall i, k, i < k \quad (4)$$

$$y_k + p_k \cdot l_{yk} + q_k w_{yk} \leq y_i + (1 - d_{ik}) \cdot M, \forall i, k, i < k \quad (5)$$

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).

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

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).

$$x_i + p_i \cdot l_{xi} + q_i w_{xi} \leq L + (1 - s_i) \cdot M, \forall i \quad (7)$$

$$y_i + p_i \cdot l_{yi} + q_i w_{yi} \leq W + (1 - s_i) \cdot M, \forall i \quad (8)$$

El modelo matemático del problema de empaquetamiento óptimo en contenedores donde no se permite rotación en las piezas se encuentra al reemplazar las ecuaciones 2, 3, 4, 5, 7, 8 por las ecuaciones 9, 10, 11, 12, 13, 14 respectivamente.

$$x_i + p_i \leq x_k + (1 - a_{ik}) \cdot M, \forall i, k, i < k \quad (9)$$

$$x_k + p_k \leq x_i + (1 - b_{ik}) \cdot M, \forall i, k, i < k \quad (10)$$

$$y_i + q_i \leq y_k + (1 - c_{ik}) \cdot M, \forall i, k, i < k \quad (11)$$

$$y_k + q_k \leq y_i + (1 - d_{ik}) \cdot M, \forall i, k, i < k \quad (12)$$

$$x_i + p_i + q_i \leq L + (1 - s_i) \cdot M, \forall i \quad (13)$$

$$y_i + p_i + q_i \leq W + (1 - s_i) \cdot M, \forall i \quad (14)$$

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. ALGORITMO DE SOLUCIÓN

### 4.1 Particle Swarm Optimization (PSO)

PSO usa un simple mecanismo que imita el comportamiento de las bandadas de pájaros y los cardumes de peces en cuanto a la forma en que guían sus desplazamientos a fin de encontrar alimento y refugio, considerados como su objetivo primordial [14].

PSO es fácil de implementar y ha sido ampliamente usado en variadas aplicaciones con resultados excelentes resolviendo problemas reales de optimización. Este algoritmo puede ser computacionalmente ineficiente por que puede quedar atrapado fácilmente en óptimos locales cuando resuelve problemas cuyo espacio de solución es multimodal; estas debilidades han hecho que el campo de aplicación de la metodología esté un poco restringido. Sin embargo, acelerar la velocidad de convergencia y evitar los óptimos locales se han convertido en los dos más importantes objetivos en la investigación de PSO [15].

En este desarrollo, el control de parámetros del algoritmo y la combinación de los operadores de búsqueda se han convertido en auxiliares de dos de los tres enfoques más destacados y prometedores, el otro es la mejora de la estructura topológica [16].

Para mejorar la eficiencia y acelerar el proceso de búsqueda, es fundamental determinar el estado de evolución y los mejores valores para los parámetros, con el fin de evitar posibles óptimos locales en el estado de convergencia. Algunas técnicas se han planteado introduciendo operadores como la selección [17], cruzamiento [18], mutación [19], búsqueda local [20].

Estas operaciones híbridas normalmente se implementan en cada generación [19], o en un intervalo prefijado [20], o son controladas por una función de adaptación definida para el caso específico. En este trabajo se realizó una búsqueda exhaustiva para la calibración de los parámetros usando como base los rangos propuestos en [14].

### 4.2 Estructura de PSO

En PSO, un conjunto de partículas representan potenciales soluciones, donde cada partícula  $i$  está asociada a dos vectores, el vector de velocidades:

$$V_i = [V_i^1, V_i^2, \dots, V_i^D]$$

y a la posición del vector:

$$X_i = [x_i^1, x_i^2, \dots, x_i^D]$$

donde  $D$  determina el tamaño del espacio de soluciones. La velocidad y la posición de cada partícula son inicializadas por vectores aleatorios con sus correspondientes rangos. Durante el proceso de evolución, la velocidad y la posición de la partícula  $i$  se actualizan mediante la siguiente expresión.

$$v_i^d = w v_{i-1}^d + c_1 \text{rand}_1^d (pbest_i^d - x_i^d) + c_2 \text{rand}_2^d (nbest^d - x_i^d) \quad (15)$$

$$x_i^d = x_{i-1}^d + v_i^d \quad (16)$$

Donde  $w$  es el peso de inercia,  $c_1$  y  $c_2$  son los coeficientes de aceleración y  $\text{rand}_1^d$  y  $\text{rand}_2^d$  son números aleatorios uniformemente distribuidos y generados en el intervalo [0-1] para la  $d$ -ésima dimensión.

Los pasos del algoritmo son los siguientes:

- i) Se inicia con una población de partículas, con posiciones y velocidades en el espacio del problema  $d$  dimensional generado de forma aleatoria.
- ii) A cada partícula se le evalúa la función objetivo.
- iii) Se compara el valor de la función de adaptación (*fitness*) de la partícula con su *pbest* (la mejor posición obtenida por la partícula). Si su valor corriente es mejor, el *pbest* pasa a ser igual al valor de la *fitness* de la partícula y la localización de la *pbest* pasa a ser igual a la localización actual del espacio  $d$  dimensional.
- iv) Se compara el valor de la función *fitness* con el mejor valor de adaptación de la población. Si el valor actual es mejor que el *gbest* (la mejor posición alcanzada por el cúmulo), actualizar el valor del *gbest*.
- v) Modificar la velocidad y la posición de acuerdo a las ecuaciones (15) y (16), respectivamente.

vi) Volver al paso ii hasta que el criterio de parada sea satisfecho. Uno de los más utilizados es el número máximo de ciclos generacionales.

**4.3 Codificación en árbol de cortes**

La codificación presentada se basa en la adecuación de la metodología propuesta en [21] denominada codificación de árbol binario de cortes.

El área de trabajo se divide en subespacios. Para garantizar que los subespacios creados presenten cortes de tipo guillotina, se define una codificación de árbol binario complementario de la siguiente forma:

Se define el número de capas, con base en este valor se determina el número de cortes. El número de capas  $c$  es definido de forma aleatoria y determina el número de variables en la codificación, así como el número de subespacios  $N_s$  creados. El número de cortes  $p$  está dado por la ecuación 17.

$$p = 2^c - 1 \tag{17}$$

El número de subespacios está dado por la ecuación 18:

$$N_s = 2^c \tag{18}$$

Para representar la estructura que genera el subespacio son definidos dos vectores de tamaño igual al número de cortes: un vector  $T$  de tipo binario que define el tipo de corte (vertical u horizontal); el segundo,  $H$ , es un vector real con valores entre 0 y 1 que determina la distancia porcentual desde la esquina inferior del tablero disponible para determinar la posición del corte.

Por ejemplo, si se decide que el número de capas es 3 entonces los cortes a obtener serán 7. Los vectores  $T$  y  $H$  deberán proporcionar la información de los tipos de corte y las distancias de los cortes, respectivamente. En la figura 1 se muestra una alternativa de distribución de un tablero con 8 subespacios.

$$\text{Número Cortes} = 2^3 - 1 = 7$$

$$\text{Número Subespacios} = 2^3 = 8$$

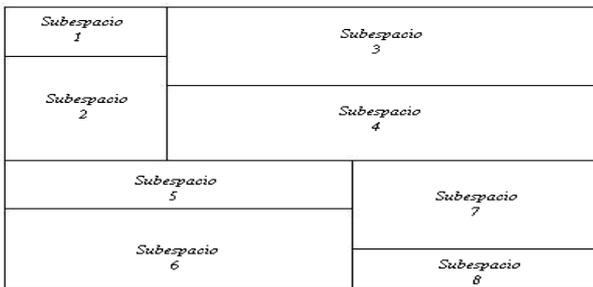


Figura 1. Distribución de un tablero con tres capas.

**4.4 Implementación de la técnica al problema**

La metodología propuesta se basa en dos algoritmos centrales, la figura 2 representa el esquema relacional de la misma.

La metodología se basa en encontrar los valores óptimos de los vectores  $T$  y  $H$ , el algoritmo  $I$  realiza la búsqueda exhaustiva de todas las posibles combinaciones del vector  $T$ , mientras el algoritmo  $II$  hace la exploración usando la técnica PSO para determinar el valor óptimo del vector  $H$ .

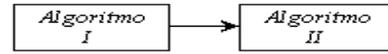


Figura 2. Esquema de la metodología.

**5. RESULTADOS**

A continuación se presentan los resultados obtenidos de la solución de 50 casos de estudio referenciados en [12]. En la tabla 1 se presentan los valores de la función objetivo obtenidas de solucionar los casos con 3 metodologías diferentes: la primera corresponde a los resultados reportados en la referencia [12], la segunda a los resultados reportados en la referencia [21] y la tercera corresponde a los resultados obtenidos con el algoritmo propuesto en este artículo.

En la tabla 3 se presentan las respuestas obtenidas de solucionar los casos con 2 metodologías diferentes: la primera corresponde a los resultados reportados en la referencia [22] y la segunda a los resultados obtenidos con el algoritmo propuesto en este artículo. De estas tablas se observa que los mejores resultados fueron obtenidos con el algoritmo propuesto y logra igualar o superar a los tres algoritmos referenciados en los casos estudiados.

**6. CONCLUSIONES**

Se resolvió el problema de empaquetamiento bidimensional con y sin rotación de piezas mediante el algoritmo de *Particle Swarm Optimization*, obteniendo resultados de excelente calidad.

Se superaron el 86% de los casos reportados en [12] y el 60% de las respuestas reportadas en [21], lo cual demuestra que la metodología propuesta es lo suficientemente robusta y puede ser aplicada a cualquier problema que esté enmarcado dentro del tipo OG 2D-BPP.

Evaluando los resultados para los problemas del tipo RG 2D-BPP, se corrobora que la posibilidad de rotar 90° las piezas aumenta el grado de complejidad del problema, pero mejora ostensiblemente la calidad de las respuestas.

En trabajos futuros se podrían analizar diferentes estrategias de calibración de parámetros como las propuestas en las referencias [14], [17], [18], [19], [20] a fin de validar su eficiencia sobre las respuestas finales.

Otra posible aplicación de la metodología propuesta sería implementarla para resolver los problemas de empaquetamiento óptimo tridimensional.

Caso	Ref. [12]	Ref. [21]	PSO Propuesto
1	98.03	99.31	99.43
2	97.48	98.17	97.9
3	98.04	98.88	98.92
4	95.59	98.53	98.53
5	98.36	98.12	98.1
6	98.06	98.49	97.85
7	98.31	98.91	98.65
8	98.49	98.9	98.9
9	97.31	98.3	98.53
10	94.14	98.3	98.64
11	96.28	97.74	97.98
12	97.13	97.79	97.63
13	97.95	98.91	98.69
14	96.17	96.49	96.98
15	98.19	98.67	98.71
16	96.70	98.27	98.06
17	94.78	96.49	97.14
18	96.79	97.33	97.78
19	98.32	98.55	99.19
20	94.83	96.61	96.61
21	98.20	97.94	98.07
22	95.68	96.32	96.15
23	96.82	97.22	97.73
24	94.95	95.15	95.98
25	96.01	98.27	97.89
26	96.06	97.97	98.05
27	94.33	98.16	98.31
28	96.63	98.34	98.51
29	96.76	98.04	98.39
30	98.27	97.92	98.02
31	94.71	97.78	97.78
32	97.00	97.37	97.57
33	97.96	98.31	98.31
34	97.82	98.04	98.09
35	96.70	97.09	97.31
36	94.35	97.38	97.52
37	94.67	97.21	97.36
38	98.03	98.07	98.08
39	96.74	98.60	98.98
40	97.78	97.44	98.21
41	98.84	98.49	98.53
42	96.92	98.30	97.96
43	96.05	97.41	97.39
44	96.81	97.33	97.17
45	96.83	96.67	96.71
46	95.60	97.63	97.61
47	96.11	97.18	97.29
48	97.85	97.54	98.76
49	98.26	98.12	98.06
50	96.74	98.22	98.22

Tabla 1. Resultados sin rotación

Resumen	Ref. [12]	Ref. [21]
Superados	43	30
Igualados	0	6
No superados	7	14

Tabla 2. Resumen de resultados sin rotación

Caso	Ref. [22]	PSO Propuesto
1	99.31	99.43
2	98.55	98.55
3	98.88	98.92
4	98.53	98.53
5	98.61	98.1
6	99.43	99.43
7	99.06	99.06
8	98.93	98.90
9	98.54	98.53
10	98.69	98.64
11	98.32	98.32
12	98.41	98.54
13	98.91	98.69
14	96.78	96.98
15	99.15	99.15
16	98.39	98.06
17	97.55	98.02
18	97.91	97.78
19	98.86	99.19
20	97.17	97.17
21	98.49	98.07
22	97.36	97.36
23	98.81	98.81
24	96.45	96.45
25	98.35	98.49
26	98.32	98.05
27	98.22	98.31
28	98.45	98.51
29	98.51	98.70
30	98.36	98.02
31	98.42	98.42
32	98.37	98.67
33	98.51	98.55
34	98.55	98.09
35	97.71	97.71
36	98.37	98.37
37	98.15	98.15
38	98.96	98.98
39	98.68	98.98
40	97.96	98.21
41	99.40	99.40
42	99.08	99.08
43	97.64	97.39
44	98.20	98.20
45	97.84	97.99
46	98.13	98.13
47	97.89	98.17
48	97.90	98.76
49	98.67	98.06
50	98.43	98.22

Tabla 3. Resultados con rotación

Resumen	Ref. [22]
Superados	18
Igualados	18
No superados	14

Tabla 4. Resumen de resultados con rotación

## 7. BIBLIOGRAFÍA

- [1] Vendramini E., “*Otimização do Problema de Carregamento de Container Usando uma Metaheurística Eficiente*”, Trabajo de Grado, UNESP, Brasil, 2007.
- [2] Parreño F. “*Algoritmos Heurísticos Y Exactos Para Problemas De Corte No Guillotina En Dos Dimensiones*”, Tesis Doctoral. Universidad De Valencia, Servei De Publicacions, España, 2004.
- [3] Christofides N., Whitlock A., “An algorithm for twodimensional cutting problems”, *Operational Research*, vol 25, pp 30 -44, 1977.
- [4]. Gilmore P.C. y Gomory R.E., “The theory and computation of knapsack functions”, *Operations Research*, vol 17, pp 1045-1074, 1966.
- [5] Wang, P., “Two algorithms for constrained twodimensional cutting stock problems”, *Operations Research*, vol 31, pp. 573-586, 1983.
- [6] Vasko, F.A., “Computational improvement to Wang's two-dimensional cutting stock algorithm”, *Computers and Industrial Engineering*, vol 16, pp. 109-115, 1989.
- [7] Oliveira J. F. y Ferreira J. S. “An improved version of Wang's algorithm for two - dimensional Cutting Problems”, *European Journal Operations Research*, vol 44, pp. 256-266, 1990.
- [8] Lai K y Chan J., “A evolutionary algorithm for the rectangular cutting stock problem”, *International Journal of Industrial Engineering*, vol 4, pp. 130-139, 1997.
- [9] Parada V., Sepúlveda M. y Gómez A., “Solution for the Constrained Guillotine Cutting Problem by Simulated Annealing”, *Journal on Computers and Operations Research*, vol 25, pp. 37-47, 1998.
- [10] Leung T.W., Yung C.H. y Troutt M.D., “Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem”, *Computers and Industrial Engineering*, vol 40, pp.201-214, 2001.
- [11] Beasley J.E., “A population heuristic for constrained two-dimensional nonguillotine cutting”, *European Journal of Operational Research*, vol 156, pp. 601-627, 2004.
- [12] Cui Y., “An exact algorithm for generating homogenous T-shape cutting patterns”, *Computers & Operations Research*, vol 34, pp. 1107-1120, 2007. Disponible en Internet en : <http://www.gxnu.edu.cn/Personal/ydcui/English/Paper.htm>
- [13] Chen C. S., Lee S. M. y Shen Q. S., “An analytical model for the container loading Problem”, *European Journal of Operational Research*, vol 80, pp. 68-76, 1995.
- [14] Zhi-Hui Z., Jun Z., Yun L. y Henry S., “Adaptive Particle Swarm Optimization”. *IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics*, This article has been accepted for inclusion in a future issue of this journal 2009.
- [15] Liang J. J., Qin A. K., Suganthan P. N. y S. Baskar, “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions”, *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.
- [16] Eberhart R. C. y Shi Y., “Guest editorial”, *IEEE Trans. Evol. Comput.—Special Issue Particle Swarm Optimization*, vol. 8, no. 3, pp. 201–203, 2004.
- [17] Angeline P. J., “Using selection to improve particle swarm optimization”, in *Proc. IEEE Congr. Evol. Comput.*, Anchorage, AK, pp. 84–89 , 1998.
- [18] Chen Y. P., Peng W. C. y Jian M. C., “Particle swarm optimization with recombination and dynamic linkage discovery”, *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 6, pp. 1460–1470, 2007.
- [19] Andrews P. S., “An investigation into mutation operators for particle swarm optimization”, in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, pp. 1044–1051, 2006.
- [20] Liang J. J. y Suganthan P. N., “Dynamic multi-swarm particle swarm optimizer with local search,” in *Proc. IEEE Congr. Evol. Comput.*, pp. 522–528, 2005.
- [21] Toro E., Garces 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 Universidad de Antioquia*, vol 46, pp. 119-131, 2008.
- [22] Álvarez D, Toro E y Gallego R., “Empaquetamiento óptimo bidimensional con rotación de piezas usando un algoritmo híbrido de búsqueda en vecindad variable y recocido simulado”, *Revista Scientia et Technica*, vol 41, pp. 279-284, 2009.