# EMPAQUETAMIENTO ÓPTIMO EN TIRAS CON Y SIN ROTACIÓN RESUELTO CON CÚMULO DE PARTÍCULAS

*Two-dimensional strip packing problem with and without rotations using Particle Swarm Optimization*

**RESUMEN**

En este artículo se resuelve el problema de empaquetamiento óptimo en tiras con y sin rotación de piezas usando la técnica de optimización cúmulo de partículas. Para su solución se usó un esquema de codificación en árbol, el cual fue adaptado a la metodología de solución, dándole gran potencialidad al proceso de solución. Este problema es de amplia aplicación en el sector de la manufactura y de interés tanto para la comunidad académica como para el sector productivo. Se compararon las respuestas obtenidas con casos de prueba de la literatura especializada con los que se validó la estrategia propuesta a fin de mostrar la buena calidad de los resultados.

**PALABRAS CLAVES:** Empaquetamiento óptimo en tiras, optimización con cúmulo de partículas, rotación ortogonal de piezas.

**ABSTRACT**

*In this paper we solve strip cutting/packing problem with and without items rotation using Particle Swarm Optimization. We used a binary tree of cutting encoding, giving potentiality at the process of solution. This problem has a wide application in the industry and it is such studied by the academic community as by the private sector. The computational results on a class of benchmark problems indicate that the algorithm performs better than several recently published algorithms.*

**KEYWORDS:** *Orthogonal rotations, particle swarm optimization, 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

Dentro de la familia de los problemas de empaquetamiento bidimensional está el de empaquetamiento óptimo en tiras (2D-SPP) del inglés *two-dimensional strip packing problem*, que consiste en un conjunto de rectángulos que deben ser ubicados en un área rectangular con un ancho limitado y una altura de tamaño variable. El objetivo es ubicar todas las piezas demandadas, sin sobreponer las piezas y utilizando la menor altura.
Este problema es considerado NP-Duro debido al tamaño de su espacio de solución [1]. Existen cuatro variantes del mismo dependiendo de la posibilidad de rotar o no las piezas y del tipo de corte generado, guillotina o no guillotina. Esta problemática es de amplia aplicación en el sector textil, metalmecánico, papelero y en general en todos los procesos productivos donde la materia prima viene empacada en rollo.

Dentro de la literatura especializada aparecen numerosas heurísticas que intentan resolver el problema, en [2] aparece un compendio muy completo que las resume. La comunidad académica, en su interés por resolver este problema tan relevante, está permanentemente en la búsqueda de estrategias que permitan mejorar la calidad de las respuestas y los tiempos computacionales de procesamiento.

El propósito de este artículo es presentar un algoritmo que usa estrategias del algoritmo de optimización cúmulo de partículas (PSO), que permitan resolver el problema 2D-SPP con y sin rotación de las piezas. Este algoritmo será validado con casos de prueba de la literatura especializada.

Este documento tiene la siguiente presentación: inicialmente se hace una descripción del problema, en la sección 3 se presenta el modelo matemático, en la sección 4 se presenta el método de solución, así como codificación de árbol binario de cortes y su adecuación a la técnica PSO, en la sección 5 se presentan el análisis de los resultados obtenidos de las distintas instancias evaluadas y finalmente en 6 se muestran las conclusiones, recomendaciones y propuestas para trabajos futuros.

Fecha de Recepción: 15 de Septiembre de 2009.
Fecha de Aceptación: 12 de Octubre de 2009

## 2. DESCRIPCION

El problema de empaquetamiento óptimo en tiras (2D-SPP) considera un conjunto de $N$ piezas rectangulares, cada pieza $i$ tiene un ancho $q_i$ y un largo $p_i$ (donde $i \in \{1,2,\ldots, N\}$). Este problema consiste en empacar todas las piezas en una tira de ancho $W$ y longitud infinita. Sin perdida de la generalidad, se asume que las dimensiones de las piezas y la tira son enteras. El objetivo es minimizar el largo total usado para empacar las piezas sin sobreponer 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° (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: 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 tiras (cortes) tipo guillotina con y sin rotación de piezas, reconocidos en la literatura como RG y OG 2D-SPP.

## 3. MODELO MATEMÁTICO

Los problemas de empaquetamiento presentan una alta dificultad de ser formalizados, en [3] se desarrolla un modelo matemático general, el cual es aceptado por la comunidad investigativa alrededor del tema. El modelo representa el problema de 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* con y sin rotación.

A continuación se describen algunas notaciones necesarias para comprender los modelos matemáticos de los problemas resueltos en este artículo.
$L$: Largo de la hoja de material.
$N$: Número disponible de rectángulos para posicionar en la hoja de material.
$(x_i , y_i)$: 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 al cual esta paralelo el largo de la pieza $i$.
$(w_{xi} , w_{yi})$: Variables binarias que indican el eje de la hoja de material al cual esta paralelo el ancho de la pieza $i$.
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.

El modelo matemático del problema de empaquetamiento óptimo bidimensional en tiras sin permitir rotación de piezas se describe a continuación:

La función objetivo consiste en minimizar el largo total usado tal como se expresa en la ecuación (1).

$$Min\ L \qquad (1)$$

Sujeto a:

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

$$x_i + p_i \le x_k + (1 - a_{ik}) \cdot M , \forall i, k, i < k \qquad (2)$$

$$x_k + p_k \le x_i + (1 - b_{ik}) \cdot M , \forall i, k, i < k \qquad (3)$$

$$y_i + q_i \le y_k + (1 - c_{ik}) \cdot M , \forall i, k, i < k \qquad (4)$$

$$y_k + q_k \le y_i + (1 - d_{ik}) \cdot M , \forall i, k, i < k \qquad (5)$$

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

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

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

$$x_i + p_i + q_i \le L + (1 - s_i) \cdot M , \forall i \qquad (7)$$

$$y_i + p_i + q_i \le W + (1 - s_i) \cdot M , \forall i \qquad (8)$$

El modelo matemático del problema de empaquetamiento óptimo en tiras permitiendo rotación de piezas se encuentra al reemplazar las ecuaciones 2, 3, 4, 5, 7, 8 por las ecuaciones 9, 10, 11, 12, 13, 14 respectivamente.

$$x_i + p_i \cdot l_{xi} + q_i w_{xi} \le x_k + (1 - a_{ik}) \cdot M, \forall i,k,i<k \quad (9)$$

$$x_k + p_k \cdot l_{xk} + q_k w_{xk} \le x_i + (1 - b_{ik}) \cdot M, \forall i,k,i<k \quad (10)$$

$$y_i + p_i \cdot l_{yi} + q_i w_{yi} \le y_k + (1 - c_{ik}) \cdot M, \forall i,k,i<k \quad (11)$$

$$y_k + p_k \cdot l_{yk} + q_k w_{yk} \le y_i + (1 - d_{ik}) \cdot M, \forall i,k,i<k \quad (12)$$

$$x_i + p_i \cdot l_{xi} + q_i w_{xi} \le L + (1 - s_i) \cdot M, \forall i \quad (13)$$

$$y_i + p_i \cdot l_{yi} + q_i w_{yi} \le W + (1 - s_i) \cdot M, \forall i \quad (14)$$

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. Los modelos planteados son del tipo de programación lineal entera mixta. Dado lo anterior, las técnicas de solución que más se adecuan para resolverlo son las metaheurísticas; una técnica exacta presentaría tiempos computacionales prohibitivos.

## 4. MÉTODO DE SOLUCIÓN

### 4.1 Optimización con cúmulo de partículas

Los primeros estudios del comportamiento social de los individuos datan a principios del siglo XX. Reynolds [4] presenta una investigación realizada sobre el comportamiento de las aves, en este se estudió el comportamiento individual y colectivo, lo que dio origen a la técnica combinatorial denominada PSO. Basados en estos conceptos, Kennedy y Eberhart en 1995 presentaron la formulación matemática del modelo PSO [5].

El método PSO es una metaheurística del grupo de los algoritmos evolutivos, su estrategia se fundamenta en el comportamiento social de los animales tales como las bandadas de aves, bancos de peces o enjambres de abejas, los cuales actúan como si fueran un solo individuo. En estos grupos de animales se establecen relaciones entre los individuos y se crean jerarquías dependiendo de las características de los mismos. El líder guía a los individuos a regiones promisorias, sin embargo, los miembros del grupo durante su recorrido pueden inferir sobre una nueva dirección. El papel de líder puede cambiar si existe otro individuo con mejores características.

Esta técnica presenta alguna similitud con los algoritmos genéticos, ya que el proceso incluye una población. El PSO cuenta con operadores que permiten realizar una búsqueda en el espacio de solución. En PSO la exploración del espacio de solución se realiza a través de una población de individuos conocidos como partículas, donde cada una de ellas representa una posible solución del problema. La ubicación de cada partícula sobre la región de búsqueda está determinada mediante su posición, la cual, representa el valor de las variables de decisión del problema.

Cada partícula cambia de posición de acuerdo a su velocidad teniendo en cuenta la mejor solución encontrada por ésta a lo largo del proceso (*Pbest*) y a la información del líder del cúmulo (*Gbest*). El operador *Pbest* (individual best) compara la posición actual de una partícula con la mejor posición que ha presentado durante el proceso de búsqueda. Mientras que el operador *Gbest* (global best) estudia el comportamiento del grupo, almacenando la posición actual del líder.

#### 4.1.1 Población inicial

Dependiendo de la complejidad matemática del problema, la población inicial es generada de forma aleatoria o utilizando métodos constructivos basados en factores de sensibilidad. Para este estudio la población inicia con un conjunto de partículas generadas de forma aleatoria.

Una población está conformada por *k* partículas y cada una de estas en el estado *i* representa una alternativa de solución, la cual es interpretada como la posición de la partícula. Cada partícula se representa mediante un vector cuyas posiciones indican los valores de las variables del problema (ver figura 1) y simbolizan un punto dentro del espacio de solución.

$\boldsymbol{x}_i =$ | $\boldsymbol{x}_{i1}$ | $\boldsymbol{x}_{i2}$ | ... | $\boldsymbol{x}_{in}$ |

Figura 1. Posición de la i-ésima partícula

#### 4.1.2 Selección del líder

En cada iteración el PSO debe seleccionar el líder del grupo y para esto compara los valores de la función objetivo de cada partícula con la función objetivo del líder. El líder es aquella partícula que posea la mejor función objetivo (incumbente global). Durante el proceso de optimización se actualiza la mejor ubicación local *Pbest* y el valor de la mejor ubicación global *Gbest*.

#### 4.1.3 Función de velocidad

La velocidad permite actualizar la posición de cada partícula y representa el gradiente de cada individuo dentro del cúmulo. Al igual que la posición, la velocidad se representa a través de un vector cuyas dimensiones deben ser iguales (ver figura 2). La velocidad contiene información de la experiencia local y colectiva del grupo. Para calcular la velocidad de cada partícula se usa la siguiente expresión (15).

$\boldsymbol{v}_i =$ | $\boldsymbol{v}_{i1}$ | $\boldsymbol{v}_{i2}$ | ... | $\boldsymbol{v}_{in}$ |

Figura 2. Velocidad de la i-ésima partícula

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot rand(Pbest - x_i(t)) + c_2 \cdot Rand(Gbest - x_i(t)) \quad (15)$$

Donde:

Velocidad actual ($v_i(t)$): Dirección de vuelo que presenta una partícula. Al inicio del proceso las partículas parten del reposo.
Factor de inercia ($w$): Factor de ponderación que actúa sobre la velocidad de la partícula. Un valor elevado puede ocasionar una búsqueda muy exhaustiva mientras que un valor demasiado pequeño conlleva a que la exploración se realice de manera fugaz.
Constantes de aceleración $c_1$ y $c_2$: Valores que orientan a las partículas hacia su mejor ubicación local o global respectivamente. La adecuada calibración de estos valores permite que la población no se homogenice durante el proceso.
*Rand* y *rand*: Valores aleatorios pertenecientes a una distribución de probabilidad uniforme.

#### 4.1.4 Actualización de la posición

La nueva posición de las partículas sigue la siguiente ecuación (16):

$$x(t+1) = x(t) + v(t+1) \quad (16)$$

La posición actualizada de las partículas debe satisfacer las restricciones de las variables del problema.

#### 4.1.5 Criterio de Parada

Los algoritmos de optimización requieren de un criterio que les permita decidir cuando finalizar la exploración del espacio de solución. Entre los criterios de parada más empleados en dichos algoritmos se encuentran:
Si la incumbente no mejora después de $n$ iteraciones se escoge está solución, cumplir un número máximo de ciclos generacionales. En este caso se empleó el segundo.

### 4.2 Codificación en árbol de cortes

La codificación presentada se basa en la adecuación de la metodología propuesta en [6], 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 el valor a calibrar y determina el número de variables en la codificación, así como el número de subespacios $s$ creados. El número de cortes $p$ está dado por la ecuación 17.

$$p = 2^c - 1 \quad (17)$$

El número de subespacios está dado por la ecuación 18:

$$N_s = 2^c \quad (18)$$

Para representar la estructura que genera el subespacio son definidos dos vectores de tamaño igual al número de cortes: el primero $T$ es un vector 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 a la cual se produce el corte con respecto al tablero disponible.
Un ejemplo de árbol de cortes de dos capas, con valores de $H$ y $T$ iguales a:

$$H = \begin{pmatrix} 0,65 \\ 0,40 \\ 0,30 \end{pmatrix} T = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

*Cortes* = $2^2$-1 = 3; *Subespacios* = $2^2$ = 4

Generaría un patrón de corte como el presentado en la figura 3.

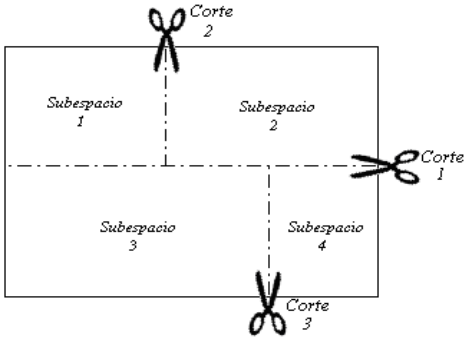


Figura 3. Ejemplo de cortes de un tablero con dos capas.

### 4.3 Implementación de la técnica al problema

La metodología propuesta se basa en tres algoritmos. La figura 4 representa el esquema relacional de la misma.

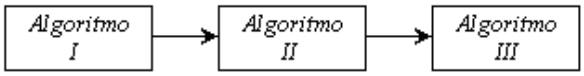


Figura 4. Esquema de la metodología.

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

**Algoritmo de transformación *Strip-Bins***

El algoritmo $I$ consiste en determinar el número total de placas y las dimensiones de cada una, donde luego se resuelve una por una como un problema de *cutting stock* a través de los algoritmos $II$ y $III$. Las dimensiones de las placas están limitadas en todo momento a las restricciones tecnológicas. El procedimiento para calcular el número de placas (*bins*) se resume en los siguientes pasos y la ecuación (19):

1. Calcular la sumatoria de las áreas de las piezas demandas y dividir esta por el ancho $W$ de la tira (*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 $l_{max}$.

$$NúmeroBins = \frac{\sum_{i=1}^{N} p_i \cdot q_i}{\frac{W}{l_{max}}} \quad (19)$$

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 incrementará en caso de que no se logre satisfacer la demanda.

## 5. ANÁLISIS DE RESULTADOS

Para verificar la eficiencia de la metodología presentada en este artículo se tomó 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 [7] 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 a un procesador Pentium ® 4 de 3.0 GHz y 1 GB de memoria RAM. El número de capas se ajustó dependiendo del número de piezas diferentes de cada instancia de prueba con relación a la ecuación (20). Los parámetros de la técnica PSO fueron calibrados conforme a los rangos presentados en [8]. El tamaño de la población fue 100 partículas, el número de ciclos generacionales fueron 100, los valores de $c_1$, $c_2$ y *w* fueron 2.05, 2.05 y 0.4, respectivamente.

$$NúmerodeCapas = \lceil \log_2 NúmerodePiezas \rceil \quad (20)$$

En la tabla 1 y 2 se presentan los valores de la función objetivo obtenidas de solucionar los casos de complejidad media con y sin rotación de piezas respectivamente, mediante 3 metodologías diferentes: la primera corresponde a los resultados reportados por Hifi [9], la segunda a los resultados reportados por Alvarez *et al.* [11] y la tercera corresponde a los resultados obtenidos con el algoritmo propuesto en este artículo. Las tablas 1 y 2 son resumidas en las tablas 3 y 4.

En la tabla 5 y 6 se presentan las respuestas obtenidas al solucionar los casos de mayor complejidad con y sin rotación 90° de las piezas, respectivamente, mediante 3 metodologías diferentes: la primera corresponde a los resultados reportados por Hifi [10], la segunda a los resultados reportados por Alvarez et al. [11] y la tercera corresponde a los resultados obtenidos con el algoritmo propuesto en este artículo. Las tablas 5 y 6 son resumidas en las tablas 7 y 8.

| Problema | Hifi [9] | Alvarez *et al.* [11] | PSO Propuesto |
|---|---|---|---|
| **SCP1** | 13 | 13 | 13 |
| **SCP2** | 40 | 42,50 | 43 |
| **SCP3** | 14 | 15 | 15 |
| **SCP4** | 20 | 20 | 20 |
| **SCP5** | 20 | 20 | 20 |
| **SCP6** | 38 | 40,87 | 36 |
| **SCP7** | 14 | 12 | 14 |
| **SCP8** | 17 | 17 | 17 |
| **SCP9** | 68 | 76 | 73 |
| **SCP10** | 80 | 80 | 80 |
| **SCP11** | 48 | 55 | 55 |
| **SCP12** | 34 | 37 | 37 |
| **SCP13** | 50 | 47,51 | 49 |
| **SCP14** | 69 | 67 | 68 |
| **SCP15** | 34 | 35 | 35 |
| **SCP16** | 33 | 33 | 33 |
| **SCP17** | 39 | 35 | 38 |
| **SCP18** | 101 | 106,47 | 104 |
| **SCP19** | 26 | 26 | 26 |
| **SCP20** | 21 | 21 | 21 |
| **SCP21** | 145 | 139 | 145 |
| **SCP22** | 34 | 39 | 39 |
| **SCP23** | 35 | 32 | 34 |
| **SCP24** | 114 | 132 | 130 |
| **SCP25** | 36 | 36 | 36 |

Tabla 1. Resultados obtenidos sin rotación.

| Problema | Hifi [9] | Alvarez *et al.* [11] | PSO Propuesto |
|---|---|---|---|
| **SCP1** | 13 | 13 | 13 |
| **SCP2** | 40 | 42,50 | 43 |
| **SCP3** | 14 | 15 | 14 |
| **SCP4** | 20 | 20 | 20 |
| **SCP5** | 20 | 20 | 11 |
| **SCP6** | 38 | 40,87 | 35 |
| **SCP7** | 14 | 12 | 13 |
| **SCP8** | 17 | 17 | 17 |
| **SCP9** | 68 | 76 | 73 |
| **SCP10** | 80 | 80 | 80 |
| **SCP11** | 48 | 55 | 55 |
| **SCP12** | 34 | 37 | 37 |
| **SCP13** | 50 | 47,51 | 48 |
| **SCP14** | 69 | 67 | 67 |
| **SCP15** | 34 | 35 | 35 |
| **SCP16** | 33 | 33 | 33 |
| **SCP17** | 39 | 35 | 35 |
| **SCP18** | 101 | 106,47 | 101 |
| **SCP19** | 26 | 26 | 26 |
| **SCP20** | 21 | 21 | 21 |
| **SCP21** | 145 | 139 | 144 |
| **SCP22** | 34 | 39 | 38 |
| **SCP23** | 35 | 32 | 34 |
| **SCP24** | 114 | 132 | 120 |
| **SCP25** | 36 | 36 | 36 |

Tabla 2. Resultados obtenidos con rotación.

| Resumen | Hifi [9] | Alvarez *et al.* [11] |
|---|---|---|
| **Superados** | 4 | 4 |
| **Igualados** | 11 | 14 |
| **No superados** | 10 | 7 |

Tabla 3. Resumen de resultados de la tabla 1.

| Resumen | Hifi [9] | Alvarez *et al.* [11] |
|---|---|---|
| **Superados** | 8 | 7 |
| **Igualados** | 10 | 13 |
| **No superados** | 7 | 5 |

Tabla 4. Resumen de resultados de la tabla 2.

| Problema | Hifi [10] | Alvarez *et al.* [11] | PSO Propuesto |
|---|---|---|---|
| **SCPL1** | 57 | 62,08 | 56 |
| **SCPL2** | 165 | 154 | 166 |
| **SCPL3** | 181 | 179 | 180 |
| **SCPL4** | 67 | 74,26 | 66 |
| **SCPL5** | 147 | 146,88 | 146 |
| **SCPL6** | 24 | 27,73 | 24 |
| **SCPL7** | 126 | 130,97 | 125 |
| **SCPL8** | 77 | 84 | 76 |
| **SCPL9** | 102 | 101 | 101 |

Tabla 5. Resultados obtenidos sin rotación.

| Problema | Hifi [10] | Alvarez *et al.*[11] | PSO Propuesto |
|---|---|---|---|
| **SCPL1** | 57 | 62,08 | 55 |
| **SCPL2** | 165 | 154 | 154 |
| **SCPL3** | 181 | 179 | 179 |
| **SCPL4** | 67 | 74,26 | 66 |
| **SCPL5** | 147 | 146,88 | 146 |
| **SCPL6** | 24 | 27,73 | 23 |
| **SCPL7** | 126 | 130,97 | 125 |
| **SCPL8** | 77 | 84 | 76 |
| **SCPL9** | 102 | 101 | 101 |

Tabla 6. Resultados obtenidos con rotación.

| Resumen | Hifi [10] | Alvarez *et al.*[11] |
|---|---|---|
| **Superados** | 7 | 6 |
| **Igualados** | 1 | 1 |
| **No superados** | 1 | 2 |

Tabla 7. Resumen de resultados de la tabla 5.

| Resumen | Hifi [10] | Alvarez *et al.*[11] |
|---|---|---|
| **Superados** | 9 | 6 |
| **Igualados** | 0 | 3 |
| **No superados** | 0 | 0 |

Tabla 8. Resumen de resultados de la tabla 6.

## 6. CONCLUSIONES

Se resolvió el problema de empaquetamiento bidimensional en tiras con y sin rotación de piezas mediante el algoritmo de *Particle Swarm Optimization* obteniendo resultados de excelente calidad para los problemas de gran escala y de buena calidad para los problemas de mediana escala.

La metodología propuesta es lo suficientemente robusta y puede ser aplicada a cualquier problema que esté enmarcado dentro del tipo OG 2D-SPP.

Evaluando los resultados para los problemas del tipo RG 2D-SPP, 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.

La metodología presentada en este artículo muestra que en casos de estudio con un alto número de piezas se obtienen resultados de excelente calidad.

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

## 7. BIBLIOGRAFÍA

[1] Martello S., Monaci M. y Vigo D., "An exact approach to the strip-packing problem", *INFORMS Journal of Computing*, vol. 15, pp. 310-319, 2003.

[2] Hopper E., "*Two-Dimensional Packing Utilising Evolutionary Algorithms and other Meta-Heuristic Methods*". PhD. Thesis Cardiff University, 2000.

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

[4] Reynolds W., "Flocks, Herds y Schools: A distributed behavioral model", *Computer graphics*. vol. 21, 1987, pp. 25-34.

[5] Kennedy, J. y Eberhart, R., "PSO optimization", *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, Perth, Australia: IEEE Service Center, pp. 1942-1948.

[6] 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 Universidad de Antioquia*, vol. 46, pp. 119-131, 2008.

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

[8] Zhi-Hui Zhan, Jun Zhang, Yun Li y Henry Shu-Hung Chung, "Adaptive Particle Swarm Optimization", *IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics*, pendiente de publicación, 2009.

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

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

[11] Alvarez D., Toro E. y Gallego R., "Solución al problema de empaquetamiento óptimo bidimensional en rollos infinitos usando un algoritmo híbrido", *Scientia et Technica*, vol. 42, 2009, pp. 205-211.