PROBLEMA DE EMPAQUETAMIENTO RECTANGULAR BIDIMENSIONAL TIPO GUILLOTINA RESUELTO POR ALGORITMOS GENÉTICOS

Two dimensional guillotine cutting packing problem using Genetic algorithm

ELIANA MIRLEDY TORO O
Ingeniera Industrial, Ms.C
Profesora catedrática.
Facultad de Ingeniería Industrial
Universidad Tecnológica de Pereira
elianam@utp.edu.co

MAURICIOGRANADA ECHEVERRI
Ingeniero Electricista, Ms.C
Profesor
Facultad de Ingeniería Eléctrica
Universidad Tecnológica de Pereira
magra@utp.edu.co

RESUMEN

En este artículo, se considera el problema de encontrar un patrón de empaquetamiento de rectángulos de diferentes tamaños, dentro de un rectángulo de mayor tamaño de forma que el área desperdiciada sea mínima. Las aplicaciones se pueden observar en industrias de perfiles metálicos, corte de maderas, papel, plástico o vidrio en donde los componentes rectangulares tienen que ser cortados desde grandes hojas de material.

Es un problema de tipo NP-completo, dado que los patrones de empaquetamiento (alternativas de solución) incrementan exponencialmente con el número de rectángulos que deben ser empacados. Como técnica de solución se presenta el algoritmo genético modificado y se sugiere una estrategia de codificación del problema basada en cortes por secciones.

PALABRAS CLAVES: Problema de empaquetamiento, corte guillotina, optimización combinatorial, metaheurísticas.

ABSTRACT

We consider problem to find a pattern to allocate a set of rectangular items of different sizes, within a rectangle of greater size so that the wasted area is minimum. The applications can be observed in metallic profiles industries, wood cut, paper, plastic or glass in where the rectangular components must be cut from great leaves of material.

It is a problem NP-hard, since the packing patterns increase exponentially with the number of rectangles that must be allocated. A Genetic algorithm is used and codification strategy of the problem based on cuts by sections is suggested

KEYWORDS: Packing problem, guillotine cutting, combinatorial optimization, metaheuristics.

1. INTRODUCCIÓN

El problema de corte de piezas rectangulares pertenece a la familia de problemas de corte y empaque y sus aplicaciones se pueden observar en industrias de perfiles metálicos, corte de maderas, papel, plástico o vidrio en donde los componentes rectangulares tienen que ser cortados desde grandes hojas de material.

En este artículo, se considera el problema de encontrar un patrón de empaquetamiento de rectángulos de diferentes tamaños, dentro de un rectángulo de mayor tamaño cuyo objetivo consiste en que el área desperdiciada sea mínima. Este problema de empaquetamiento en dos dimensiones es de interés científico y académico ya que sirve como plataforma para problemas de mayor complejidad, como el de tres dimensiones. Es un problema de tipo NP-completo, dado que los patrones de empaquetamiento incrementan exponencialmente con el número de rectángulos que deben ser empacados. La magnitud del espacio de búsqueda es mayor que el del problema del agente viajero. Por ejemplo, si el número de placas a ser ubicadas es de 25, el tamaño del espacio de soluciones está dado por $2^{25} \cdot 25!$. Mientras que el espacio de soluciones del agente viajero es del orden de 10^{31} para un caso con igual número de ciudades considerando que todas las ciudades estén conectadas entre si. $2^{25} \cdot 25! > 10^{31}$. Esta es una de las razones para usar las técnicas metaheurísticas como herramienta de solución[1].

El problema descrito considera un alto grado de diversidad en los tipos de *placas* o piezas rectangulares y una única *paleta* en las que deben ser empacadas. Además, sin pérdida de generalización, se asume que todas las entradas son enteros positivos. La posibilidad de rotar las placas un ángulo de 90º no es considerada en este trabajo. Sin embargo, para una placa en particular es

Fecha de Recepción: 30 Mayo de 2007
Fecha de Aceptación: 23 Julio de 2007

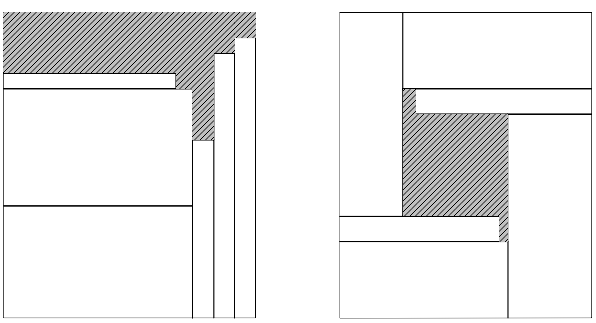
posible resolver este problema generando un nuevo tipo de placa con dimensiones equivalentes a ésta rotada 90º.

Este documento se presenta de la siguiente manera. El problema de empaquetamiento bidimensional es presentado en la sección 2. lLa Sección 3 explica el modelo matemático , en la sección 4 se presentan el algoritmo de solución adaptado al problema específico, en la sección 5 se muestran las pruebas y resultados y en la sección 6 se presentan las conclusiones con base a los resultados obtenidos.

2. PLANTEAMIENTO DEL PROBLEMA

En un problema de empaquetado de rectángulos se dispone de un conjunto de piezas rectangulares o *placas* con longitudes conocidas y se desean distribuir estas en un objeto rectangular de mayor tamaño, denominado *paleta*, de forma que se minimice el área desperdiciada en la paleta. Una forma alternativa de interpretar el problema supone que se dispone de un objeto rectangular desde el que hay que obtener un conjunto de piezas realizando cortes perpendiculares a los ejes.
Este problema se conoce comúnmente como el problema restringido bi-dimensional de corte 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 la figura 1 se muestran los diferentes tipos de problema, el área sombreada representa las áreas sin utilizar [2].

La consideración de diferentes tipos de restricciones puede originar problemas de empaquetamiento totalmente distintos.



1(a) Corte Guillotina 1(b) Corte no guillotina

Figura 1. Tipos de Cortes

Pueden considerarse las siguientes tres situaciones referidas al conocimiento que se tiene del objeto en que hay que distribuir las piezas[3]:

1. El largo y el ancho de la paleta son conocidos. Se pretende, por tanto, distribuir en el objeto aquellas piezas que optimicen la función objetiva considerada. Esta situación es la que se aborda en este artículo.

2. Sólo es conocido el ancho de la paleta mientras que el largo se considera flexible. En este tipo de problema se desea encontrar la distribución de piezas que minimice el largo de la paleta. En [4] y [5] se plantean algunos ejemplos de este problema.

3. No son conocidos ni el largo ni el ancho de la paleta. El propósito es distribuir las piezas de tal forma que el rectángulo que esta distribución determina sea el de menor área. En [6] se plantean algunos ejemplos de este problema.

3. MODELO MATEMÁTICO

El problema ha sido formulado como un problema lineal entero en [7], en este trabajo se aborda un nuevo planteamiento. Para una adecuada formulación matemática del problema, se implementa la siguiente notación:
L: altura de la *paleta* rectangular de almacenamiento.
W: ancho de la *paleta*.
M: número total de piezas rectangulares o *placas* a ser ubicadas.
$i \in \{1,2,\ldots,m\}$: conjunto de índices de los tipos de *placas*.
l_i: altura de la *placa* rectangular de tipo i.
w_i: ancho de la *placa* de tipo i.
π_k: alternativa de solución k conformada por dos vectores como se muestra en la figura 3.
S_k: número máximo de secciones de la alternativa de solución k.

La expresión matemática de la función objetivo para una alternativa de solución k es:

$$Fobj_k = L \times W - \sum_{s=1}^{S_k} \sum_{m=\pi_k.\text{secciones}(s)+1}^{\pi_k.\text{secciones}(s+1)} l_{\pi_k.\text{placas}(m)} \cdot w_{\pi_k.\text{placas}(m)} \qquad (1)$$

La minimización de la función objetivo mostrada en (1) está sujeta al cumplimiento de las siguientes restricciones, a fin de obtener una solución factible:

1. Todas las piezas tienen una orientación fija, es decir, no se permiten rotaciones de 90ª. Esto significa que una placa de ancho w y altura l es distinta a una pieza de ancho l y altura w.
2. La placas deben ser ubicadas dentro de la paleta de forma ortogonal. Esto quiere decir que los lados de las placas son paralelos a los lados de la paleta.
3. El ancho y la altura de cada placa no debe exceder las dimensiones de la paleta.
4. Las dimensiones de las placas y las de la paleta son números enteros positivos.
5. Todos los cortes realizados sobre la paleta se consideran perfectos, sin grosor.
6. Pueden haber varias placas del mismo tipo.

El objetivo del problema consiste en ubicar las placas sobre la paleta de forma que la diferencia entre el área de la paleta y el área total ocupada por las placas sea mínima (minimizar el desperdicio)[8]. Una alternativa factible de solución será aquella en la que todas las placas quedan contenidas dentro de la paleta, ubicadas ortogonalmente y sin traslaparse unas con otras. Por lo tanto, se pretende encontrar una solución factible que minimice el área desperdiciada.

4 ALGORITMO GENÉTICO DE CHU-BEASLEY

Las principales características del Algoritmo Genético (AG) de Chu-Beasley consisten en mantener constante el tamaño de la población de alternativas de solución, de manera que en cada iteración se reemplaza una alternativa de la población usando un eficiente mecanismo de modificación de la misma pero teniendo en cuenta que no se admiten configuraciones repetidas dentro de la población. En cada iteración la población es reemplazada sistemáticamente por un único descendiente generado[9,10]. Esta estrategia tiene la ventaja de permitir encontrar múltiples soluciones y además conservar la diversidad del conjunto de alternativas. El algoritmo que describe el método es el siguiente:

1. Se genera, aleatoriamente, una población inicial de soluciones.
2. Se obtienen las funciones objetivo, penalizando las soluciones infactibles.
3. Se seleccionan dos padres usando el método de selección por torneo. En este método, dos configuraciones son elegidas aleatoriamente y se elige uno de los padres considerando el mejor valor en función objetivo, el otro es descartado. El proceso se ejecuta dos veces para obtener dos padres.
4. El proceso de recombinación y mutación se realiza de forma integrada ya que se deben tener en cuenta consideraciones especiales del problema. Estos operadores actúan sobre los padres escogidos en el paso anterior. La idea se amplia en la sección 4.2. En esta fase se realizan mejoras en la factibilidad y en la optimalidad, teniendo en cuenta la filosofía de bloques constructivos.
5. Se reemplaza un individuo de la población por la solución encontrada tomando como base lo siguiente:
   a. Si la alternativa actual es infactible y a su vez es menos infactible que la peor infactible de la población, entonces reemplazar la peor infactible por la alternativa actual. Es decir, entre dos soluciones infactibles se prefiere la solución menos infactible.
   b. Si la configuración es factible y existe por lo menos una infactible en la población actual, entonces reemplazar la peor infactible por la alternativa actual. Es decir, entre una solución factible y una infactible se prefiere la solución factible.
   c. Si la configuración es factible y todas las alternativas de la población actual son factibles, entonces reemplazar la alternativa con peor función objetivo por la alternativa actual. Lo anterior se realiza sólo si la alternativa actual es de mejor calidad que la peor de la población. Es decir, entre dos soluciones factibles se prefiere la solución con mejor función objetivo.

4.1 Representación del Cromosoma

Se tiene una paleta rectangular de 24x38, donde se desean ubicar *m* placas rectangulares de menor tamaño y de diferentes tipos, en la tabla 1 se presentan los datos de largo y ancho de cada una de las piezas y la cantidad de piezas demandadas, una configuración factible es aquella donde se ubiquen algunas piezas de las demandadas de forma que se minimice el área sin utilizar.

Una alternativa de solución π está codificada por dos vectores. El primero, denominado "*Placas*", contiene una secuencia de ubicación de placas según el tipo. Dicho vector es de un tamaño igual al número total de placas. El segundo vector se denomina "Secciones" y contiene los intervalos para los cuales el primer vector ubica las placas en una misma sección. El número total de secciones de cada alternativa es determinado de forma aleatoria. Por lo anterior, una alternativa π es un arreglo de dos vectores a los cuales se puede hacer referencia como *π.Placas* y *π.Secciones*. La altura de una sección está determinada por la placa de mayor altura.

	Largo	Ancho	Cantidad demandada
Paleta	24	38	
Pieza tipo 1	14	19	2
Pieza tipo2	9	6	2
Pieza tipo 3	3	6	2
Pieza tipo 4	6	6	4
Pieza tipo 5	9	3	2
Pieza tipo 6	10	1	4

Tabla 1.Datos del ejemplo

Así, por ejemplo, la alternativa infactible mostrada en la figura 2 corresponde a la ubicación de placas mostrada en la figura 3.

π.Placas=	2	0	1	5	1	0	6	6	4	0	7	3	4	7	2	7	0	6	4	5

π.Secciones =	0	6	15	20

Figura 2. Alternativa de solución π.

El vector *Secciones* indica que esta alternativa posee 3 secciones, la primera comprendida entre las posiciones 0 y 6 del vector *Placas*, la segunda entre 7 y 15 y la tercera entre 16 y 20. Esto implica que, por ejemplo, la segunda sección contiene las placas 6, 6, 4, 7, 3, 4, 7 y 2. De esta forma, una alternativa de solución está completamente definida por los vectores *Placas* y *Secciones* los cuales representan la forma de ubicación de las placas en la paleta principal. La secuencia de llenado de las placas siempre se realiza de izquierda a derecha en cada sección.

Nótese que en el vector *Placas* existe la posibilidad de encontrar campos con un valor 0. Esta posibilidad permite que el algoritmo de optimización pueda, eventualmente, excluir placas de la alternativa de solución. Es decir, no es necesario que todas las placas tengan que ser ubicadas en la paleta para una alternativa de solución dada.

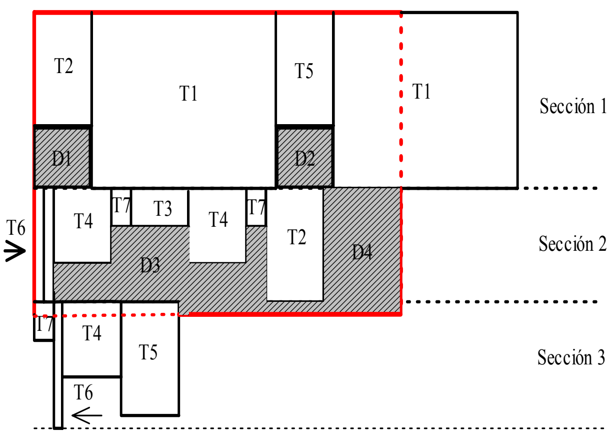


Figura 3. Representación de una alternativa de solución

A cada sección se asigna un área utilizada por las placas contenidas en la misma. Además, si existen placas que se salen de la paleta, la sección es penalizada incrementando el área desperdiciada en un área igual a la suma de las áreas de las placas que están por fuera de la paleta. Esto sirve como factor de sensibilidad para orientar la búsqueda de mejores soluciones. En otras palabras, es necesario calcular el área desperdiciada en cada sección con el fin de tener un indicador de las peores secciones y por lo tanto preferir cambios, de forma probabilística, en estas secciones.

En la figura 3 se muestra una alternativa de solución dónde el área disponible que corresponde a la paleta está representada por líneas rojas, y las áreas sombreadas corresponden a los desperdicios, dentro de cada uno de los rectángulos se indica el tipo de pieza que se está ubicando, y las líneas punteadas en negro indican las secciones de corte; en la sección 1 de la alternativa mostrada se puede apreciar que el área desperdiciada es la suma de los rectángulos nombrados como D1 y D2 , D1 tiene dimensiones de 5 por 6, y D2 tiene dimensiones de 5 por 3, lo que da un total de área sin utilizar en esta sección de 45 unidades.

Sin embargo, la segunda pieza tipo 1 está ubicada por fuera de la paleta por lo cual esta sección es penalizada en un valor igual al área de la placa que causa la violación y corresponde a 14x19= 266. El área total desperdiciada en la sección 1 es 45 +266=311.

4.2 Proceso de recombinación y mutación

El proceso de recombinación, en este problema en particular, debe ser realizado cuidadosamente ya que es posible obtener soluciones infactibles al implementar este operador de la forma tradicional. Si se aplicara este tipo de recombinación se corre el riesgo de obtener una configuración en la cual una placa podría ocupar dos posiciones diferentes sobre la paleta.

En este artículo se propone incorporar la filosofía de la recombinación a través de la conservación de bloques constructivos. El proceso consiste en usar dos puntos de corte sobre los padres seleccionados por torneo los cuales son representados por el vector π.*placas*, como el mostrado en la figura 2. Posteriormente se aplica un operador de cruzamiento especial denominado cruce parcialmente combinado o PMX por su significado en inglés (Partially Matched Crossover). Este operador permite recombinar dos cromosomas sin perder los bloques del cromosoma que ya se había construido. En el algoritmo tradicional PMX los puntos de corte son escogidos aleatoriamente. Para este problema los puntos de corte son determinados por el vector π.*placas* de una de las alternativas escogida aleatoriamente, en donde se selecciona una sección para que la recombinación se realice conservando la sección escogida. La escogencia de la sección se hace usando selección por ruleta y se da mayor probabilidad a la sección de mejor calidad, es decir, se trata de conservar las secciones con menor desperdicio o de buena calidad. Así por ejemplo, la recombinación entre dos alternativas *i* y *j* se lleva a cabo como se muestra a continuación:

π_i.secciones					
	0	4	7	10	13

Sección escogida por ruleta

	1	2	3	4	5	6	7	8	9	10	11	12	13
$\pi_i.placas$	5	8	11	6	7	9	10	12	1	3	2	4	13
$\pi_j.placas$	13	5	1	10	6	2	3	4	7	11	12	8	9

Puntos de recombinación

Resultan dos descendientes, donde el primero conserva la sección conformada por las placas (7, 9, 10):

13	5	1	6	7	9	10	4	2	11	12	8	3

El segundo descendiente conserva la sección conformada por las placas (6, 2, 3):

5	8	11	7	6	2	3	12	1	9	10	4	13

El proceso de mutación en el algoritmo genético tradicional es bastante simple. Consiste en intercambiar aleatoriamente dos genes de los cromosomas obtenidos anteriormente. En el problema que se maneja en este artículo, el proceso de mutación es un poco distinto, debido a que es posible afectar cualquiera de los dos cromosomas que conforman una alternativa de solución (ver figura 2).

La mutación consiste en escoger por ruleta una de las secciones del hijo seleccionado. Se da mayor probabilidad a las secciones de peor calidad y se modifica la sección escogida siguiendo las siguientes estrategias:

1. Modificar el vector π.*placas* original haciendo permutaciones suaves usando el operador de mutación tradicional. Es decir, escogiendo de forma aleatoria dos elementos del vector π.*placas*, de la sección ganadora en la ruleta, e intercambiándolos entre si.
2. Modificar el vector π.*placas* original quitando de la sección uno de sus elementos de forma aleatoria.
3. Modificar el vector π.*placas* original reemplazando uno de sus elementos por una de las placas que no pertenecen a la alternativa actual. Esto se hace de forma aleatoria y afectando la sección actual.
4. Modificar el vector π.*secciones* original reduciendo o aumentando el tamaño de la sección actual. Este procedimiento es de gran importancia porque permite realizar búsquedas redimesionando las secciones. El impacto de realizar este paso sobre la alternativa de solución es el de pasar placas de una sección a otra.

De los descendientes anteriores, se escoge uno usando selección proporcional (ruleta) dando mayor probabilidad a las alternativas con mejor valor de función objetivo, incluyendo la alternativa original.

5. PRUEBAS Y RESULTADOS

Para demostrar la eficiencia del algoritmo, se probó con diferentes casos de prueba y se obtuvieron los siguientes resultados.

Caso 1. Paleta de altura=24 y ancho=38.

Tipo	Alto	Ancho	Cantidad
1	12	19	2
2	14	19	2
3	2	4	25
4	3	38	1
5	5	19	2
6	2	2	1

Tabla 2. Dimensiones y cantidad de placas para caso 1.

Para este caso se utilizó una población de 10 individuos generando aleatoriamente tanto el vector de ubicación de placas como el vector de secciones El algoritmo propuesto penaliza la infactibilidad de modo que la respuesta final contiene todas las placas dentro de la paleta

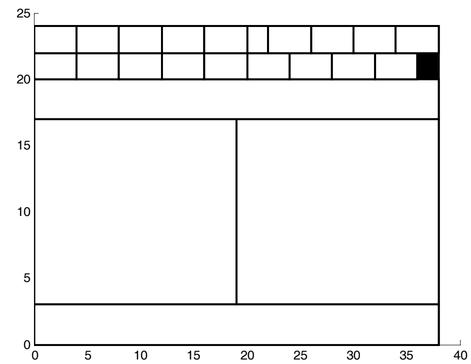


Figura 4. Solución obtenida para el caso 1.

El factor de penalización usado es igual a 2.

Caso 2. Paleta de altura=10 y ancho=10.

Tipo	Alto	Ancho	Cantidad
1	3	2	4
2	7	2	4
3	4	2	4
4	6	2	4
5	9	1	4
6	8	4	4
7	4	1	4
8	1	10	4
9	3	7	4
10	4	5	4

Tabla 3. Dimensiones y cantidad de placas para caso 2.

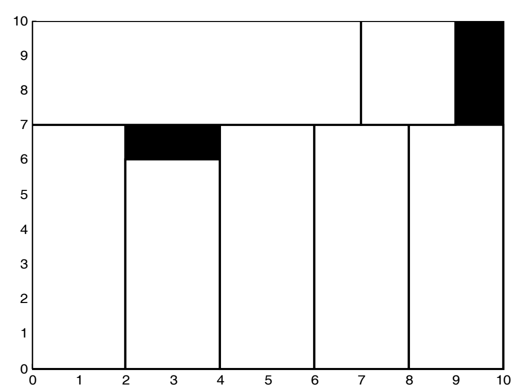


Figura 5. Solución obtenida para el caso 2.

Esta respuesta se obtuvo con parámetros idénticos a los del caso 1. Cada caso se corrió 20 veces y en el 99% de ellas se llegó a las soluciones mostradas. Para el caso 1 se obtuvo un porcentaje de utilización del material disponible de 99.5% y para el caso 2 el porcentaje de utilización fue del 95 %.

El caso 1 tiene un tamaño del espacio de soluciones del orden de $2^{33} \cdot 33!$ y el caso 2 de $2^{40} \cdot 40!$. Lo cual los hace problemas de difícil solución con un grado mayor de complejidad que el problema del agente viajero.

Los tiempos computacionales involucrados en la solución de estos casos, para un equipo con procesador Intel Pentium de 1.6 GHz, fueron de 20 y 28 segundos, respectivamente.

6. CONCLUSIONES

La codificación presentada es de fácil implementación y muestra un buen comportamiento cuando el problema es resuelto por algoritmos genéticos.

Los bloques constructivos son beneficiados con este tipo de codificación ya que se preserva, a través de selección proporcional, al manejar división por secciones. La optimalidad total se mejora cuando el área desperdiciada en una sección es reducida. Por esta razón, el algoritmo presenta un mejor desempeño cuando el factor de penalización por secciones es escogido adecuadamente.

La optimalidad se mejora haciendo mutaciones suaves a los bloques de mala calidad hasta que mejore o hasta un número de intentos determinado. Este paso es similar a una búsqueda intensiva. Cuando una sección no tiene área desperdiciada se debe procurar mantener sus elementos unidos como un bloque. Esto se logra dandole a esta sección una baja probabilidad de ser modificada.

En el proceso de mutación se considera agregar, quitar y modificar tanto el vector de ubicación de placas como el de secciones.

Es posible mejorar el resultado de los datos obtenidos empleando otros tipos de codificación para una alternativa de solución. Una propuesta para trabajos futuros consiste en implementar una codificación basada en grafos[11] o en ubicación estratégica, como lo propone el algoritmo de BL[8,12,13]. La codificación de este problema usando grafos permite abordar su solución usando optimización por colonia de hormigas.

7. AGRADECIMIENTOS

Los autores expresan su agradecimiento a la Universidad Tecnológica de Pereira por el apoyo prestado al grupo de desarrollo en investigación operativa (DINOP).

8. BIBLIOGRAFÍA

[1] S. Jakobs, Theory And Methodology On Genetic Algorithms For The Packing Of Polygons. European Journal Of Operational Research. Vol.88, p87-100. 1996.

[2]F. Parreño. Algoritmos Heurísticos Y Exactos Para Problemas De Corte No Guillotina En Dos Dimensiones. Tesis Docotoral. Universidad De Valencia, Servei De Publicacions. España 2004

[3] C. Beltrán E. Calderón Procedimientos Constructivos Adaptativos (Grasp) Para El Problema Del Empaquetado Bidimensional. IX Conferencia De La Asociación Española Para La Inteligencia Artificial. VI Jornadas De Transferencia Tecnológica De Inteligencia Artificial (Volumen II)), pp. 755-764. Gijón, España. Del 14 Al 16 De Nov. De 2001

[4] **S. Benati.** An Algorithm For A Cutting Stock Problem On A Strip. Journal Of Operational Research Society Vol. 48 Pp. 288- 294 (1997)

[5] E. Hopper, B. Turton. An Empirical Investigation Of Meta-Heuristic And Heuristic Algorithms For A 2d Packing Problem. European Journal Of Operational Research Vol. 128 Pp. 34-57 (2001)

[6] I. Hwang. An Efficient Processor Allocation Algorithm Using Two Dimensional Packing.Journal Of Parallel And Distributed Computing Vol. 42 Pp. 75-81 (1997)

[7] Beasley, J.E., An exact two-dimensional non-guillotine cutting tree search procedure. Operations Research 33, 49–64. 1985.

[8] T.W. Leung , Chi Kin Chan, M. Troutt Application Of A Mixed Simulated Annealing-Genetic Algorithm Heuristic For The Two-Dimensional Orthogonal Packing Problem. European Journal Of Operational Research 145 (2003) 530–542

[9] Vendramini E. Optimización del problema de cargamento de contenedores usando una metaheur+istica eficiente".Tesis Doctoral. UNESP.Brasil.2007

[10] Toro E., Granada M., Romero R. Algoritmo Memético Aplicado A La Solución Del Problema Deasignación Generalizada.Revista Tecnura. Año 8 No 16 Semestre I -2005.

[11] Diego-Mas.J.E. "Optiomización de la distribución en planta de instalaciones industriales mediante algoritmos genéticos. Aportación al control de la geometría de actividades". Tesis doctoral. Universidad Politécnica de Valencia.España. 2006

[12] J.E. Beasley, A population heuristic for constrained two-dimensional nonguillotine cutting, European Journal of Operational Research 156 (2004) 601-627.

[13] Andrea Lodi , Silvano Martello, Michele Monaci. Two-dimensional packing problems: A survey. European Journal of Operational Research 141 (2002) 241–252