

# HEURISTICA DE DOS-ETAPAS PARA EL PROBLEMA DE CORTE DE PIEZAS CON GUILLOTINADO BIDIMENSIONAL<sup>▲</sup>

## TWO-STAGE HEURISTIC FOR THE CUTTING PARTS PROBLEM USING TWO-DIMENSIONAL GUILLOTINE

Ma. Loecelia Guadalupe Ruvalcaba Sánchez<sup>1,♦</sup>, Juan Gabriel Correa Medina<sup>1</sup>, Vittorio Zanella Palacios<sup>2</sup>

### RESUMEN

En un ambiente altamente competitivo, el problema de corte de guillotina bidimensional es un elemento clave en la reducción de costos. Este problema tiene una amplia gama de aplicaciones en industrias cuyos procesos de corte de materiales se realizan con máquinas que sólo permiten cortes de un extremo a otro. En este trabajo se presenta un algoritmo de dos etapas usando metaheurísticas para acomodar en una sola placa de ancho conocido y longitud infinita, un conjunto de ítems rectangulares fuertemente heterogéneos que pueden ser rotados 90°. El objetivo es minimizar la longitud requerida de la placa procurando la acumulación del desperdicio. En la primera etapa se aplica un algoritmo de búsqueda tabú para determinar el orden en que se acomodan los ítems. En la segunda, se busca determinar el mejor acomodo de los ítems en la placa mediante un algoritmo de recocido simulado. Se experimenta con un conjunto de instancias conocidas. Los resultados muestran que la rotación de piezas favorece la obtención de soluciones que igualan al menos las reportadas previamente en la literatura y que la concentración de los desperdicios incrementa su posibilidad de reutilización.

**Palabras clave:** Corte y empaquetado, búsqueda tabú, recocido simulado

### ABSTRACT

Into a highly competitive environment, two-dimensional guillotine's cut problem is an elementary key to cost reduction. This problem has a wide variety of applications into factories related to processes material cut. Cuts are done by machines which cut from one edge to other. A two stage algorithm is shown to place a finite set of items in a single plate with a known width and infinite length, using metaheuristics. All items are rectangular, mostly are distinct itself and each one can only be rotated 90 degrees at once. Main objective is minimization of length required by plate and waste accumulation were compacted as much as possible. First stage determines the order from items placed by using a tabu search algorithm. Second stage tries

---

<sup>▲</sup>VI Simposio Internacional de Ingeniería Industrial: Actualidad y nuevas tendencias 2013, Bogotá, Colombia. Artículo actualizado

<sup>1</sup>Departamento de Sistemas de Información, Centro de Ciencias Básicas, Universidad Autónoma de Aguascalientes, Av. Universidad #940, Cd. Universitaria, C.P. 20131, Aguascalientes, Ags., México.

<sup>2</sup>Departamento de Ingeniería, Facultad de Tecnologías de Información, UPAEP, 21 sur #1103 Barrio Santiago C.P. 72410, Puebla, Pue., México

<sup>♦</sup>Autor para correspondencia: lgruvalcaba@correo.uaa.mx

improving the order previous, through a simulated annealing algorithm. Experiments were performed over a set instance already known. Results showed solutions at least, equal or better than solutions known, due to rotation. Also, certain amount of waste grouped was achieved, which means a possible material recycling.

**Keywords:** Guillotine cut, packing, tabu search, simulated annealing

## INTRODUCCIÓN

Uno de los grandes objetivos de las industrias manufactureras es la optimización del uso de sus materias primas, debido a que representan un alto porcentaje del costo de su producto terminado. El corte de materiales efectuado en sus procesos es fundamental para poder ser competitivas. Esto hace que el problema de corte bidimensional (2CP, Two-dimensional Cutting Problem) tenga una amplia gama de aplicaciones en industrias dedicadas al corte de piezas de piel, tela, plástico, vidrio y madera, por mencionar algunos.

El 2CP consiste en cortar un conjunto de rectángulos con longitudes conocidas, denominados ítems, a partir de uno o varios rectángulos mayores llamados placas, con el requerimiento de que los ítems se ubiquen en la placa para ser cortados de manera tal que optimicen alguna función del área que ocupan sin sobreponerse y sin sobrepasar los límites de la placa. Una forma alternativa de interpretar el problema supone que se dispone de un conjunto de piezas rectangulares con longitudes conocidas que se desean distribuir en un objeto rectangular más grande. Este tipo de problema pertenece a la categoría de los llamados problemas de corte y empaquetado (C&P), los cuales son considerados problemas de optimización combinatoria NP-duros (Dowland & Dowland, 1992).

Los problemas C&P se remontan a 1939 con Kantorovich y 1949 con Brooks (Álvarez, 2010). Sin embargo Gilmore y Gomory (1961) presentaron la primera publicación relevante en esta área. Sweeney y Paternoster (1992) hacen una recopilación y clasificación de las publicaciones hechas entre 1940 y 1990 en base a las dimensiones, tipo del problema y tipo de aproximación empleada en la solución, mientras que Whitwell (2004) hace una recopilación de publicaciones de 1970 a 2003.

Aunque estos problemas son fáciles de definir intuitivamente, su modelación formal no es sencilla, además, la estrecha relación que guardan con las características de los materiales y su aplicación a procesos específicos ha dado lugar a múltiples variantes. La primera tipología para permitir un sistema de notaciones y definiciones unificado y consistente, fue propuesta por Dickhoff (1990) y fue mejorada posteriormente por Wäscher, Haussner y Schumann (2007). Esta última tipología establece que los nuevos criterios para la definición de estos problemas son: 1) dimensionalidad (uno/dos/tres); 2) tipo de asignación (maximización/minimización); 3) surtido de objetos pequeños (idénticos/débilmente heterogéneos/fuertemente heterogéneos); 4) surtido de objetos grandes (uno/múltiples) y 5) forma de los pequeños ítems (rectángulos/círculos/cajas/cilindros).

Lodi, Martello y Vigo (1999) definieron otras características no tan generales y que hacen referencia a la orientación y patrones de corte. Dependiendo de las características del material, los ítems pueden tener una orientación fija (O) o pueden ser rotados 90° (R), mientras que las características tecnológicas de las máquinas, pueden imponer (G) o no (F) un patrón de corte tipo guillotina. Un corte es de tipo guillotina si los ítems son obtenidos a partir de una secuencia de cortes de extremo a extremo de la placa, produciendo dos nuevos rectángulos por corte.

El-Bouriet *et al.* (2006) proponen una nueva heurística para resolver el problema de corte rectangular de tamaño múltiple que minimiza razonablemente el desperdicio y los tiempos computacionales. Toro y Granada (2007) utilizan un algoritmo genético modificado para encontrar un patrón de corte rectangular tipo guillotina y sugieren una estrategia de codificación del problema basada en cortes por secciones. Terashima-Marín *et al.* (2007) presentan dos modelos basados en computación evolutiva para producir hiper-heurísticas que resuelvan problemas de empaquetado bidimensional.

Toro, Rueda y Ruíz (2008) presentan un algoritmo de búsqueda tabú para el problema de corte bidimensional tipo guillotina usando diferentes soluciones constructivas de inicio para observar su efecto sobre la solución encontrada después de varias iteraciones. Yanasse y Morabito (2008) presentan modelos lineales y no-lineales para generar un grupo de patrones de corte bidimensionales tipo guillotina con y sin restricciones incluyendo casos exactos y no-exactos. Estos modelos son útiles para la investigación y desarrollo de métodos de solución efectivos, exploración de estructuras particulares, modelos de descomposición y relajación, entre otros. Clautiaux, Jouglet y Moukrim (2008) presentan una nueva clase de coloreado de grados orientados, llamados grafos guillotina, que modelan patrones guillotina y proponen un algoritmo de reconocimiento para esos grafos.

Yaodong *et al.* (2009) presenta un algoritmo para el problema de corte bidimensional restringido de objetos rectangulares que genera recursivamente un patrón óptimo de bloque uniforme y selecciona de manera óptima la primera tira de corte a partir de la sub-placa actual. Hifi, M'Hallah y Saadi (2009) proponen un algoritmo exacto de ramificación y acotamiento y un algoritmo aproximado de dos etapas que usa esquemas de cota inferior y superior eficientes. Amossen y Pisinger (2010) presentan una generalización de un algoritmo constructivo para el problema de empaquetado multidimensional, con y sin restricciones de guillotina, basado en programación de restricciones. Morabito y Pureza (2010) presentan un método heurístico para generar patrones restringidos de corte de guillotina de dos dimensiones, usando un espacio de relajación de una formulación de programación dinámica del problema y un procedimiento ascendente de optimización subgradiente. Hifi y Saadi (2010) estudian un problema de corte de orientación fija bidimensional de dos etapas e investigan el uso de un algoritmo de búsqueda en haz paralelo para resolver el problema de manera aproximada.

Recientemente se ha popularizado la utilización de métodos híbridos en la solución de problemas de corte de guillotina de dos dimensiones que consisten en la combinación de metaheurísticas como recocido simulado, búsqueda tabú, algoritmos evolutivos, métodos exactos, entre otros. Aryanezhad *et al.* (2012) proponen un nuevo método heurístico práctico y rápido para la solución de un problema de corte de dos dimensiones con elementos y placas rectangulares utilizando cortes de guillotina. En su implementación el usuario tiene acceso a información relacionada con la dimensión y número de cortes requeridos, la dimensión y número de piezas de desecho y el porcentaje de desperdicio producido. Park *et al.* (2013) presentan un problema de corte de guillotina de dos dimensiones mediante un algoritmo de dos etapas. Ellos usan un método de bandas de empaque para resolver un problema procedente de la industria del vidrio. El algoritmo propuesto genera niveles o bandas donde los elementos se seleccionan en base a la longitud de la tira considerando una rotación de  $90^\circ$ .

En este artículo se presenta un algoritmo de dos etapas para resolver un problema de corte de guillotinado bidimensional que busca acomodar en una sola placa de ancho conocido, un conjunto de ítems rectangulares fuertemente heterogéneos que pueden ser rotados  $90^\circ$ . El objetivo es minimizar la longitud requerida de la placa tratando de concentrar las áreas de desperdicio para su posterior reutilización. La primera etapa consiste en un algoritmo de búsqueda tabú (TSA –Tabu Search Algorithm) que ayudará a determinar el mejor orden en que los ítems deben considerarse para su posterior acomodo en la placa. La segunda etapa

consiste en un algoritmo de recocido simulado (SAA–Simulated Annealing Algorithm), mediante el cual se busca obtener la configuración del acomodo de los ítems en la placa (considerando las merma por corte) que minimice la cantidad de material requerido. Las configuraciones resultantes se muestran gráficamente para facilitar el proceso de corte.

## MATERIALES Y MÉTODOS

### Definición del problema

Se tiene un conjunto  $I$  de tipos de ítems rectangulares  $i=\{1,2,\dots,m\}$  que pueden ser rotados  $90^\circ$  en cualquier dirección y que deben ser acomodados y cortados a partir de una placa de ancho  $W$  y largo desconocido  $L$ . Cada tipo de ítem  $i$  tiene una demanda  $d_i=\{1,2,\dots,n\}$  y dimensiones heterogéneas, dadas por un ancho  $w_i$  y un largo  $l_i$  en unidades, tal que  $l_i$  y  $w_i$  no deben exceder  $W$  al mismo tiempo. La placa deberá dividirse en un conjunto  $S$  de subespacios  $s=\{1,2,\dots,z\}$  producidos a partir de cortes tipo guillotina con un ancho  $x_s$  y un largo  $y_s$ , de manera que se acomoden la totalidad de los ítems sin superponerse y se minimice el largo requerido  $L$  de placa. El problema propuesto puede ser modelado matemáticamente como sigue:

$$\min L \tag{1}$$

Sujeto a:

$$\frac{\sum_{s=1}^z \sum_{i=1}^m l_i w_i H_{si} V_{si}}{W} \geq L \tag{2}$$

$$\sum_{s=1}^z H_{si} + V_{si} \geq d_i, \forall i \in I \tag{3}$$

$$\sum_{i=1}^m w_i H_{si} + l_i V_{si} \leq x_s, \forall s \in S \tag{4}$$

$$\sum_{i=1}^m l_i H_{si} + w_i V_{si} \leq y_s, \forall s \in S \tag{5}$$

$$H_{si} \geq 0, \forall i \in I, \forall s \in S \tag{6}$$

$$V_{si} \geq 0, \forall i \in I, \forall s \in S \tag{7}$$

donde  $H_{si}$  y  $V_{si}$  son variables enteras que representan el número de ítems tipo  $i$  colocados horizontal y verticalmente en el subespacio  $s$ , respectivamente. La función objetivo (1) y la restricción (2) minimizan el largo requerido de la placa. La restricción (3) busca garantizar que el número de elementos de cada tipo colocados horizontal o verticalmente en los subespacios de la placa satisfaga la demanda. Las restricciones (4) y (5) aseguran que los ítems ubicados horizontal o verticalmente en cada subespacio son factibles, es decir que la suma de sus dimensiones no exceda el ancho ni el largo de cada subespacio. Finalmente, las restricciones (6) y (7) indican que el número de piezas colocadas horizontal y verticalmente, respectivamente, deben ser mayor o igual que 0.

## METODOLOGÍA

Bajo el supuesto de que el orden en que se acomodan los ítems determina la longitud requerida de la placa, así como en el número y orientación de los cortes tipo guillotina necesarios para obtenerlos, se propone un algoritmo de dos etapas para la solución de este problema (Figura 1). La primera etapa consiste en establecer el orden de acomodo de los ítems mediante un TSA. La segunda etapa usa un SAA para acomodar los ítems dentro de la placa. Las dos etapas son complementarias. TSA es utilizado para asegurar que el orden de acomodo de las piezas no se repite en el corto plazo y para lograr determinar en el mediano o largo plazo si existe un patrón de acomodo no favorable (tabú). Mientras que el SAA busca asegurar un buen acomodo de las piezas en la placa, para ello, acepta eventualmente soluciones de calidad pobre e implementa algunos otros mecanismos que le permiten detectar cuando ha incurrido en un mínimo local.

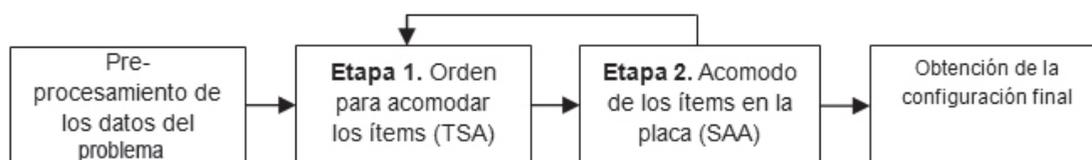


Figura 1. Metodología de solución propuesta

### Preprocesamiento de datos

El pre-procesamiento de datos consiste en la lectura del ancho  $W$  de la placa, la unidad de medida preferida para la instancia, el desperdicio de material producido por los cortes y los tipos de ítems que se desean obtener, sus medidas  $w_i$  y  $l_i$  y la cantidad  $d_i$  demanda de cada uno. Una vez obtenida esta información, los ítems se enumeran de manera consecutiva para simplificar su identificación y acomodo en las etapas posteriores.

### Orden en que se acomodan los ítems

TSA es un procedimiento metaheurístico de optimización combinatoria basado en conceptos de inteligencia artificial propuesto por Glover (1989). El método hace uso de una memoria adaptativa y de exploración sensible que lo forzan a explorar nuevas áreas en el espacio de búsqueda evitando caer en óptimos locales. Es decir, se pueden memorizar en el corto plazo algunas soluciones que han sido examinadas recientemente y que en el largo plazo pueden convertirse en puntos prohibidos para soluciones posteriores. Por ello, en esta propuesta el TSA ayudará a determinar el orden en que cada uno de los ítems debe ser acomodado dentro de la placa, que favorecerá la exploración de las combinaciones y permitirá detectar órdenes de inserción desfavorables (Figura 2).

```
Procedimiento OrdenarItems (Tvecindario, CriterioParo)
Inicia
  MejorSecuencia ← SecuencialInicial
  DisposicionItem(nVecinos, nExcepciones, MejorSecuencia, NuevaConfiguracion,
  NuevaEvaluacion)
  MejorSecuencia ← SecuencialInicial
  Inicializa MemoriaPrimaria
  Inicializa MemoriaSecundaria
  NoMejora ← 0
  Repite
    EvaluacionMin ← 1000000
    Para i=1 hasta Tvecindariohacer
      Tvecino[i] ← Intercambia (ObtenItem1, ObtenItem2)
      SiMemoriaSecundariaentonces
        EvaluaTvecino[i] ← EvaluacionAPriori(Tvecino[i])
      Sino
        Evaluacion ← falso
      Fin Si
    Fin Para
    Si not Evaluacion entonces
      MejorVecino ← Tvecino[aleatorio (tvecino)+1]
    Sino
      MejorVecino ← ObtenMejorVecino
    Fin Si
    DisposicionItem(nVecinos, nExcepciones, MejorSecuencia, NuevaConfiguracion,
    NuevaEvaluacion)
    EvaluacionPrevia← MejorEvaluacion
    SiMejorEvaluacion > NuevaEvaluacionentonces
      MejorEvaluacion ← NuevaEvaluacion
      MejorSecuencia ← MejorVecino
    Fin Si
    Actualiza MemoriaPrimaria
    Actualiza MemoriaSecundaria
    SiEvaluacionPrevia = MejorEvaluacionentonces
      NoMejora ← NoMejora + 1
    Sino
      No mejora ← 0
    Fin Si
  HastaNoMejora = CriterioParo
Fin Procedimiento
```

**Figura 2.** Pseudo-código del TSA para acomodo de los ítems

### **Acomodo de los ítems en la placa**

Una vez que se ha determinado el orden de los ítems, se propone el uso de un SAA para encontrar el acomodo y orientación de las piezas que minimice el largo de la placa. SAA hace una analogía con el equilibrio térmico de la mecánica estadística utilizado en el proceso de recocido de la industria de materiales resistentes o cristalinos. El proceso consiste de tres fases: 1) calentamiento a una temperatura determinada; 2) sostenimiento de la temperatura alta

que le permite a las moléculas acomodarse en estados de mínima energía; y 3) enfriamiento controlado para aumentar el tamaño de sus cristales, reducir sus defectos y alcanzar una máxima resistencia. Esta conexión con la mecánica estadística expone nueva información y proporciona una perspectiva familiar en problemas de optimización y métodos tradicionales (Kirkpatrick, Gelatt & Vecchi, 1983).

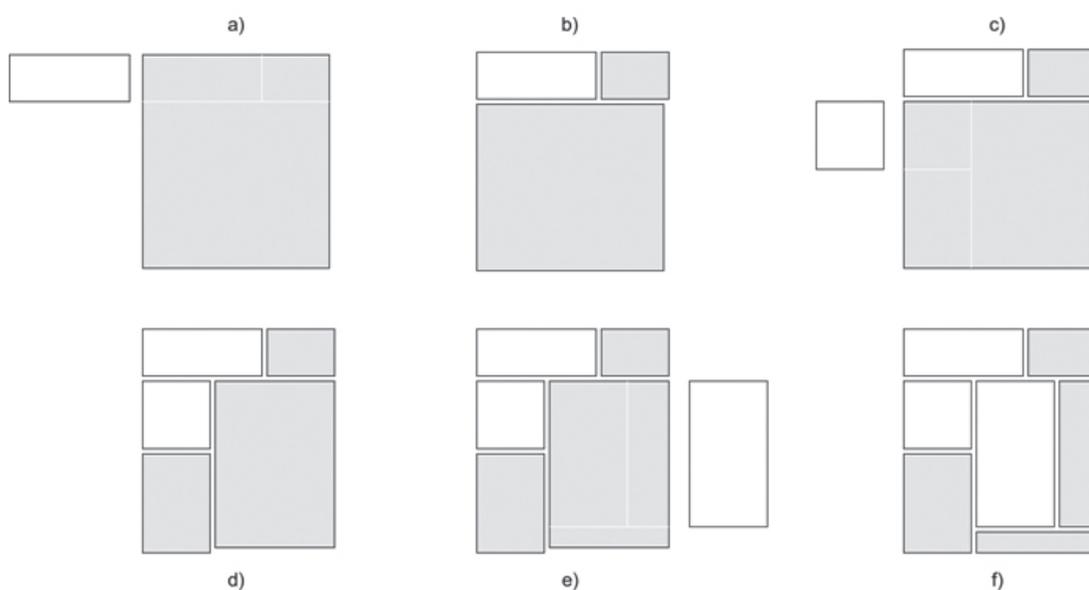
El procedimiento *DispocisionItem* (Figura 3) recibe como parámetros de entrada: 1) el tamaño de la vecindad ( $nVecinos$ ), 2) el número de soluciones iguales consecutivas para dar por terminado el algoritmo ( $nExcepciones$ ), 3) la temperatura inicial (*Templnicial*) que representa la probabilidad de aceptar una solución dominada, 4) la proporción de enfriamiento (*TempDec*) y 5) el orden de acomodo de los ítems (*Nueva Secuencia*), resultante de la etapa anterior. Como parámetros de salida el algoritmo proporciona: 1) la distribución de las piezas (*Nueva Configuración*) y 2) la longitud de la placa ( $L$ ). El algoritmo genera una configuración inicial de acomodo de los ítems en la placa y evalúa en términos de la función objetivo definida, en nuestro caso la minimización total de la longitud de la placa. Después se genera un procedimiento cíclico en el que se genera un vecindario de tamaño predefinido ( $nVecinos$ ). Si alguno de los vecinos mejora la solución anterior, ésta se actualiza, en caso contrario, se acepta una solución no dominada de acuerdo con una probabilidad variable. Si se acepta una nueva solución, el número de excepciones se inicializa en cero. En caso contrario, el contador de excepciones se incrementa en uno. Cuando el contador de excepciones llega a  $nExcepciones$ , se asume un óptimo local. El algoritmo termina cuando se detecta que la función objetivo no se mejora en un número predefinido de iteraciones continuas (dado por  $nExcepciones$ ).

El procedimiento *Organiza Items* funciona de la siguiente manera. Para insertar la primera pieza, se verifica inicialmente si el largo  $l_i$  o el ancho  $w_i$  son iguales al ancho  $W$  de la placa. Si lo anterior es verdadero, la orientación de la pieza y el primer corte se definen en función de ello, en caso contrario tanto la orientación de la pieza como la del corte se eligen de manera aleatoria.

```
Procedimiento DisposicionItem (nVecinos, nExcepciones, TempInicial, TempDec,
NuevaSecuencia, NuevaConfiguracion, NuevaEvaluacion)
Inicio
  LimpiaConfiguracion(NuevaConfiguracion)
  Organizaltems(NuevaConfiguracion, NuevaSecuencia)
  NuevaEvaluacion ← ObtenLargo(NuevaConfiguracion)
  Excepcion ← 0
  T ← TempInicial
  CEvaluacion ← 0
  Repite
    PreviaEvaluacion ← NuevaEvaluacion
    Repite
      MinLargo ← NuevaEvaluacion
      Para c=1 hastavecinos hacer
        LimpiaConfiguracion(Vecino[c])
        Organizaltems(Vecino[c], NuevaSecuencia)
        EvaluacionVecino[c] ← ObtenLargo(Vecino[c])
        Si MinLargo >= EvaluacionVecino[c] entonces
          MinLargo ← EvaluacionVecino[c]
          MejorVecino ← c
        Sino Si Aleatorio(0,1) < exp((NuevaEvaluacion-EvaluacionVecino[c])/T)
          entonces
            MinLargo ← EvaluacionVecino[c]
            MejorVecino ← c
        Fin Si
      Fin para
      Si MinLargo <> NuevaEvaluacion entonces
        NuevaConfiguracion ← Vecino[MejorVecino]
        NuevaEvaluacion ← MinLargo
        Excepcion ← 0
      Sino
        Excepcion ← Excepcion + 1
      Fin
    Si
  Hasta Excepcion = nExcepciones
  Si T > 0 entonces
    T ← T - TempDec
  Sino T ← 0
  Fin
  Si
  Si PreviaEvaluacion = NuevaEvaluacion entonces
    CEvaluacion ← CEvaluacion + 1
  Hasta CEvaluacion = nExcepcion
Fin procedimiento
```

**Figura 3.** Pseudo-código SAA para elacomodode ítems en la placa

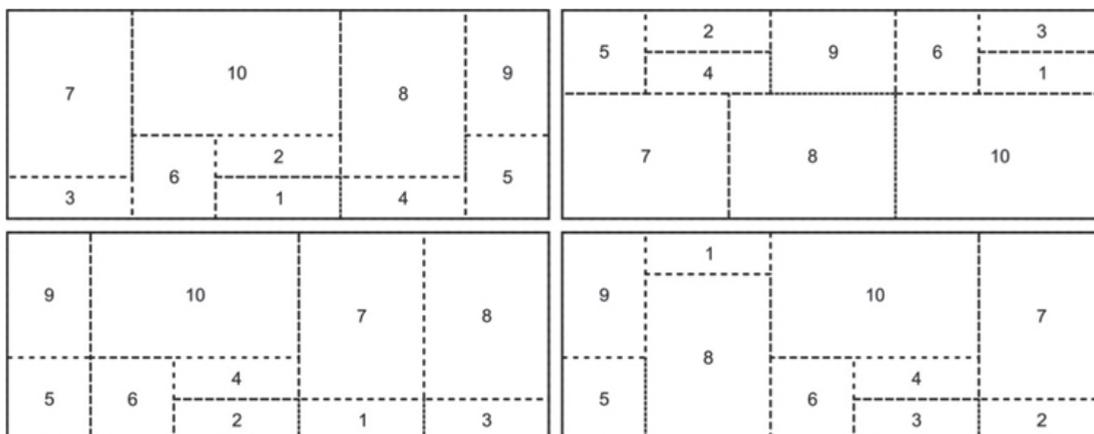
Para insertar el resto de las piezas se consideran los siguientes casos en orden de prioridad: 1) la pieza encaja completamente en vertical o en horizontal y aunque produce un corte, no genera subespacios adicionales; 2) la pieza encaja en vertical o en horizontal en algún subespacio predefinido y produce algún tipo de corte, en cuyo caso, se definirá el nuevo subespacio generado abajo y/o a la derecha del actual, dependiendo del tipo de corte que se produzca; y 3) la pieza encaja holgadamente en algún subespacio predefinido, eligiéndose de manera aleatoria la orientación del ítem y el tipo de corte, para luego definir los nuevos (2 o 3) subespacios resultantes (figura 4). El procedimiento *ObtenLargo* determina el largo requerido de la placa, la cantidad de desperdicio que produce y pondera la cercanía entre las áreas de desperdicio, de manera tal, que una solución es mejor que otra, si y sólo si, teniendo los mismos valores en largo y cantidad de desperdicio, sus áreas de desperdicio son contiguas. Esta última consideración, incrementa la posibilidad de reutilización del material en proyectos futuros.



**Figura 4.** Procedimiento de inserción de ítems en la placa

### Configuración final

La mejor configuración resultante se mostrará gráficamente con la intención de facilitar el proceso de corte, indicando el acomodo de las piezas en función de la numeración establecida en la fase de preprocesamiento de datos (Figura 5). Asimismo se indicará la longitud mínima de la placa; la cantidad de material desperdiciado por los cortes; y el número y medidas de las piezas desperdiciadas que pueden reutilizarse.



**Figura 5.** Ejemplo de configuraciones finales óptimas sin desperdicio

En la figura 6 se muestran tres configuraciones finales que sugieren la misma longitud de placa. Sin embargo, una configuración es considerada inferior si produce un mayor número de segmentos de desperdicio. Aunque a simple vista una configuración concentre más el desperdicio que otra, el uso de cortes tipo guillotina puede volverlas equivalentes, ya que inevitablemente producirá segmentos de desperdicio.



**Figura 6.** Ejemplo de configuraciones finales con desperdicio de material

La solución propuesta fue programada con el compilador Lazarus 9.26 (Baeseman, Miller y Hess, 1999). Las corridas se efectuaron en una computadora portátil con procesador AMD Athlon II Dual Core, con velocidad de 2 GHz y 3 GB de memoria RAM. Para evaluar su desempeño se emplearon 25+9 instancias de prueba para el problema de empaquetamiento rectangular bidimensional tipo guillotina (Guillotine Strip Cutting) que se encuentran en OR-Benchmark (Hifi, s.f.).

## RESULTADOS

Para verificar la estabilidad del algoritmo, se realizaron 50 experimentos por instancia, para las primeras 25 instancias, las cuales requirieron entre 15 y 120 segundos para su solución. La tabla 1 muestra las mejores soluciones obtenidas mediante el algoritmo propuesto para este bloque y las compara con las mejores respuestas reportadas por la literatura especializada. En el 64% de los casos las soluciones se igualaron, mientras que en el 36% restante se mejoraron. Las mejoras obtenidas oscilan entre el 1 y el 17 por ciento. Aunque las instancias de este bloque están ordenadas en función de su grado de complejidad, no se detectó algún patrón que indique en qué condiciones el algoritmo propuesto tiene un peor o mejor comportamiento, por lo que dichas variaciones podrían atribuirse a la complejidad implícita al problema en sí.

**Tabla 1.** Resultados computacionales obtenidos vs mejores soluciones conocidas

No. de instancia	ID del problema	Mejor largo conocido	Mejor largo obtenido	Tiempo mínimo (segundos)	Tiempo máximo (segundos)	Tasa de utilización del área	Error porcentual (%)
1	SCP1	13	13	15	16	100,00	0
2	SCP2	40	40	43	43	98,75	0
3	SCP3	14	14	48	49	100,00	0
4	SCP4	20	20	63	63	92,50	0
5	SCP5	10	10	39	40	87,50	0
6	SCP6	38	35	38	39	91,05	-9
7	SCP7	14	13	38	39	88,21	-8
8	SCP8	17	17	46	47	94,90	0
9	SCP9	68	68	68	69	100,00	0
10	SCP10	80	80	94	120	97,50	0
11	SCP11	48	47	28	85	97,87	-2
12	SCP12	34	34	76	103	100,00	0
13	SCP13	50	50	59	116	83,03	0
14	SCP14	69	68	90	110	87,81	-1
15	SCP15	34	34	98	117	100,00	0
16	SCP16	33	33	61	104	96,46	0
17	SCP17	39	38	27	49	89,47	-3
18	SCP18	101	96	53	93	92,16	-5
19	SCP19	26	25	81	109	96,00	-4
20	SCP20	21	21	92	107	89,84	0
21	SCP21	145	145	48	94	92,53	0
22	SCP22	34	34	104	116	98,33	0
23	SCP23	35	30	64	89	92,89	-17
24	SCP24	114	111	78	94	92,49	-3
25	SPC25	36	36	97	106	96,33	0

Para las 9 instancias restantes, consideradas grandes por la combinación de tamaños y densidades de los ítems y placas, se realizaron 70 experimentos para cada una, utilizando como condición de paro una temperatura mínima de 0,1 para el SAA. La tabla 2 muestra un resumen de los resultados obtenidos. En ella se observa que para el 44% de los casos, los resultados se igualan y el 66% se mejoran (entre el 1 y el 26%). Para estas instancias los tiempos de cómputo oscilaron entre 1 y 10 minutos.

**Tabla 2.** Resultados obtenidos en instancias grandes

No. de instancia	ID del problema	Mejor solución conocida	Mejor solución obtenida	Mejor largo obtenido	Tiempo mínimo (minutos)	Tiempo máximo (minutos)	Error porcentual (%)
1	SCPL1	24225	20825	49	1	7	-16
2	SCPL2	20955	20701	163	3	8	-1
3	SCPL3	40725	40725	181	4	6	0
4	SCPL4	24455	24455	67	2	7	0
5	SCPL5	24255	23595	143	5	10	-3
6	SCPL6	15768	14454	22	1	8	-9
7	SCPL7	44982	35700	100	4	7	-26
8	SCPL8	36575	36575	77	1	5	0
9	SCPL9	17850	17850	102	3	10	0

## DISCUSIONES

Un requerimiento establecido para esta implementación, es que el largo y ancho de los ítems no debe exceder al mismo tiempo el ancho conocido de la placa. En la primera etapa, el orden de acomodo inicial de las piezas, se define de manera aleatoria y se va refinando conforme el intercambio de ítems produce información para la memoria de largo plazo. Para asegurar una mejor exploración del espacio de búsqueda, los intercambios en el orden de acomodo son moderados a través de una memoria de corto plazo que impide la repetición continua de una combinación y diversifica las soluciones. La evaluación del acomodo final de los ítems en la placa (resultante de la segunda etapa), construye una memoria de largo plazo que determina si una secuencia de introducción de los ítems es desfavorable para mejorar la función objetivo.

En la segunda etapa, la rotación de las piezas y los tipos de corte se determinan en función de la disposición de espacio o de manera aleatoria, según corresponda. Para acomodar un ítem primero se verifica si éste encaja completamente en horizontal o vertical en alguno de los subespacios definidos sin producir cortes, si encaja vertical y horizontalmente produciendo algún tipo de corte o si para acomodar el ítem es necesario definir un nuevo subespacio. En los casos en que se producen cortes, primero se verifica si existe algún tipo de corte que se ajuste mejor al caso, pero si no se detecta el tipo corte se define aleatoriamente.

En los cortes de guillotina es difícil que los desperdicios se concentren, sin embargo, podemos considerar que una solución es mejor que otra, si el número de piezas de desperdicio que producen es menor. En este tipo de cortes el que los desperdicios sean visualmente contiguos no garantiza que podamos obtenerlos de esa forma, sin embargo, a veces el concentrarlos en un mismo subespacio puede ayudarnos a pensar en alternativas para reducir la cantidad de cortes entre ellos. La intención de concentrar el desperdicio de material obedece al hecho de poder utilizarlo en futuros proyectos. Aunque no es viable para todos los materiales, en la

experimentación pudo observarse que la rotación de piezas es una condición deseable, debido a que favorece la reducción y concentración de desperdicios.

Los resultados obtenidos en la experimentación son comparados con los reportados en Hifi (1998) y Hifi *et al.* (2009). Aunque en Hifi *et al.* (2009) se sugiere que las mejores soluciones obtenidas para las instancias de la tabla 1 son soluciones óptimas, por un lado, no se define el criterio de optimización, y por otro, en Hifi (1998) estas soluciones son presentadas como mejores soluciones encontradas hasta el momento. Por ello, si consideramos que el objetivo fue minimizar la longitud de la placa permitiendo una rotación de piezas, no nos extraña mejorar entre el 1 y el 17% los resultados presentados en el 46% de estas instancias, con tiempos de cómputo entre 15 y 120 segundos.

En lo que respecta a las instancias de la tabla 2, los resultados presentados nunca son referidos como óptimos en Hifi (1998) y Hifi *et al.* (2009). Un 44% de las mejores soluciones conocidas para estas instancias son igualadas y un 66% mejoradas. Los tiempos de cómputo requeridos oscilaron entre 1 y 10 minutos.

## CONCLUSIONES

Dado que las soluciones obtenidas igualan al menos las reportadas previamente en la literatura, podemos concluir que mediante la heurística propuesta es posible minimizar la longitud requerida de la placa. Sin embargo, aunque aparentemente la concentración de desperdicios es posible, dado el tipo de corte, no hay garantía de obtenerlos en una sola pieza, lo que puede o no favorecer su reutilización.

El orden en que se introducen las piezas influye en la calidad de las soluciones obtenidas, en este sentido, el TSA asociado a la primera etapa de la heurística, ayuda a identificar a través de la memorias de corto y largo plazo aquellas combinaciones que contribuyen a la obtención de resultados más eficientes en la segunda etapa.

La priorización de casos en el acomodo de las piezas considerado en el SAA contribuye a la disminución del número total de cortes en la pieza, el aprovechamiento de los subespacios previamente generados y por ende a la concentración de los desperdicios.

## RECOMENDACIONES

Como trabajo futuro se considera la utilización de estructuras de datos dinámicas que permitan un mejor aprovechamiento de la memoria de la computadora para la resolución de instancias grandes y la minimización del número de cortes requeridos para la obtención de los ítems. Dado que la consideración de mermas por concepto de corte, facilitó el uso de la herramienta en aplicaciones relacionadas con la distribución en planta, se buscará facilitar la programación manual de la memoria de largo plazo en la fase de búsqueda tabú, de manera que se preestablezcan restricciones de contigüidad entre ítems.

Adicionalmente, se considera el análisis e implementación de otras metaheurísticas que favorezcan la exploración del espacio de soluciones y que permitan tomar en cuenta las características de presentación y defectos de los diversos tipos de materiales (e.g., los nudos en la madera). Por último, se buscará extender el algoritmo propuesto a una dimensión adicional con la finalidad de observar su desempeño y usabilidad en la búsqueda de soluciones para problemas de empaquetamiento tridimensional.

## REFERENCIAS

- ÁLVAREZ, M. D. Solución del problema de empaquetamiento óptimo bidimensional en una sola placa, en placas y rollos infinitos con o sin rotación de piezas usando técnicas metaheurísticas de optimización. Tesis de Maestría. Pereira, Colombia. Universidad Tecnológica de Pereira. 2010.
- ARYANEZHAD, M.B.; HASHEMI, N.H.; MAKUI, A.; and JAVANSHIR, H.A Simple approach to the two-dimensional guillotine cutting stock problem. *Journal of Industrial Engineering International*, 2012, vol. 8, no. 21. p. 2-10.
- BAESEMANN, C.; MILLER, S.; and HESS, M. A. Free Pascal Lazarus, versión 9.26.1999. [consultado 01/06/2010]. [en línea] <http://sourceforge.net/projects/lazarus/>
- CLAUTIAUX, F.; JOUGLET, A.; and MOUKRIM, A. A new graph-theoretical model for k-dimensional guillotine-cutting problem. *Experimental Algorithms, Lecture Notes in Computer Science*, 2008.p. 43-54, DOI: 10.1007/978-3-540-68552-4\_4.
- DICKHOFF, H. A typology of cutting and packing problems. *European Journal of Operational Research*, 1990, vol. 44. p. 145–159.
- DOWSLAND, K.A.; and DOWSLAND, W.B. Packing Problems. *European Journal of Operational Research*, 1992, vol. 56, no. 1. p. 2–14.
- EL-BOURI, A.; RAO, J.; POPPLEWELL, N.; and BALAKRISHNAN, S. An improved heuristic for the two-dimensional cutting stock problem with multiple sized stock sheets. *International Journal of Industrial Engineering*, 2006, vol. 13, no. 2. p. 198-206.
- GILMORE, P.; and GOMORY, R. A linear programming approach to the cutting stock problem. *Operation Research*, 1961, vol. 9. p. 849-859.
- GLOVER, F. Tabu Search –Part 1. *Journal of computing*, 1989, vol. 1, no. 3. p. 190-206.
- HIFI, M. Exact Algorithms for the Guillotine Strip Cutting/Packing Problem. *Computers & Operations Research*, 1998, vol. 25, no. 11. p. 925-940.
- HIFI, M. Strip-cutting, OR-Benchmark. s.f. [consultado 20/12/2009]. [en línea] <ftp://cermse.univ-paris1.fr/pub/CERMSEM/hifi/Strip-cutting/>
- HIFI, M.; M'HALLAH, R.; and SAADI, T. Approximate and exact algorithms for the double-constrained two-dimensional guillotine cutting stock problem. *Computational Optimization and Applications*, 2009, vol. 42, no. 2. p. 303-326. DOI 10.1007/s10589-007-9081-5.
- HIFI, M.; and SAADI, T. A parallel algorithm for two-staged two-dimensional fixed-orientation cutting problems. *Computational Optimization and Application*, 2010. DOI 10.1007/s10589-010-9351-5.
- KIRKPATRICK, S.; GELATT, C.D.; and VECCHI, M.P. Optimization by simulated annealing. *Science, New Series*, 1983, vol. 220, no. 4598. p. 671-680.
- LODI, A.; MARTELLO, S.; and VIGO, D. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 1999, vol. 11. p. 345-357.

MORABITO, R.; and PUREZA, V. A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem. *Annals of Operations Research*, vol. 179, 2010.p. 297-315. DOI 10.1007/s10479-008-0457-4.

PARK, K.T.; RYU, J.H.; LEE, H.K.; and LEE, I.B. Developing a heuristics for glass cutting process optimization: A case of two-dimensional two-stage guillotine cutting with multiple stock sizes. *Korean Journal of Chemistry Engineering*, 2013, vol. 30, no. 2. p. 278-285.

SWEENEY, P.; and PATERNOSTER, R. Cutting and packing problems: A categorized, application-orientated research bibliography. *Journal of the Operational Research Society*, 1992, vol. 43, no. 7. p. 691-706.

TERASHIMA-MARÍN, H.; FARÍAS-ZÁRATE, C. J.; ROSS, P.; and VALENZUELA-RENDÓN, M. Comparing Two Models to Generate Hyper-heuristics for the 2D-Regular Bin-Packing Problem. *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO* (London, UK, 2007).p. 2182-2189.

TORO, E.M.; RUEDA, A.C.; and RUÍZ, H.A. Effect of the initial configuration in the solution of the two-dimensional cut problem using the taboo search algorithm. *Revista Colombiana de Tecnologías de Avanzada*, 2008, vol. 1, no. 11.p. 107-113.

TORO, E. M.; and GRANADA, M. Two dimensional guillotine cutting packing problem using genetic algorithm. *Scientia et Technical*, 2007. año XIII, vol. 35. p. 321-326.

WÄSCHER, G.; HAUSSNER, H.; and SCHUMANN, H. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 2007, vol. 183. p. 1109-1130. DOI:10.1016/j.ejor.2005. 12.047.

WHITWELL, G. Novel heuristic and metaheuristic approaches to cutting and packing. Thesis for Phd degree. United Kingdom. University of Nottingham.2004.

YANASSE, H.; and MORABITO, R. A note on linear models for two-group and three-group two-dimensional guillotine cutting problems. *International Journal of Production Research*,2008, vol. 46, no. 21.p. 6189-6206.

YAODONG, C.; XINFANG, Z.; YING, Y.; and PENG, Y. Uniform block patterns for constrained guillotine cutting of rectangular items. *International Journal of Information Management Sciences*, 2009. p. 89-101.