

# Un método de optimización proximal al problema de anidamiento de piezas irregulares utilizando arquitecturas en paralelo

Juan P. D'Amato<sup>a,b,\*</sup>, Matias Mercado<sup>a</sup>, Alejandro Heiling<sup>a</sup>, Virginia Cifuentes<sup>a,c</sup>

<sup>a</sup> Universidad Nacional del Centro de la Provincia de Buenos Aires, Instituto PLADEMA, Tandil, Pinto 399, Argentina.

<sup>b</sup> Consejo Nacional de Investigaciones Científicas y Técnicas, Ciudad Autónoma de Bs. As, Rivadavia 1917, Argentina.

<sup>c</sup> Comisión de Investigaciones Científicas de la Prov. De Buenos Aires, La Plata, calle 26, e/10 y 11, Argentina

## Resumen

Se presenta un modelo discreto que resuelve el problema bidimensional de corte y ubicación, generalmente llamado *nesting* (anidamiento), de gran interés en las industrias textiles. El problema consiste en minimizar el remanente o desperdicio de un material a través de la ordenación de moldes geoméricamente irregulares. Como solución se propone un algoritmo heurístico polinomial, flexible porque permite evaluar distintas condiciones y restricciones del problema, y paralelizable en arquitecturas de múltiples núcleos de bajo costo. La metodología propuesta se evaluó con casos de estudio de la literatura del área y se comparan los tiempos de cómputo con una herramienta comercial del sector, obteniéndose muy buenos resultados. Además, se logra una aceleración del procesamiento de hasta 4X con respecto a la versión secuencial.

## Palabras Clave:

Optimización, corte, industria textil, heurística, paralelización.

## 1. Introducción

El problema de corte (también llamado problema de posicionamiento, anidamiento o *nesting*, *Trim-Loss* o *cutting stock problem* (Dyckho 1992; Gilmore and Gomory 1965), ha sido ampliamente estudiado en la literatura. En general, el problema inicia con un material del cual se desean extraer un conjunto de piezas. Tanto el material como las piezas generalmente presentan formas irregulares, cuyas geometrías equivalen a molduras. Este modelado se sucede en distintos tipos de industria, tal como la aeronáutica, la naviera, del cartón, del vidrio, de la madera, del metal, entre otras. También existen estudios particulares cuando las piezas son regulares, siendo un subtipo del problema planteado.

Aunque los procesos se han automatizado, y hasta cuentan con gran cantidad de herramientas digitales para el seguimiento del proceso, en muchas industrias la etapa de ubicar las piezas sigue siendo manual. El nivel de desperdicio luego del corte depende de la habilidad del experto. Ciertamente, cuando el personal no tiene tanta experiencia, la etapa se vuelve crítica, incurriendo en mayores tiempos de corte y un mayor costo por sub-utilización del material. Tal como se conoce, este es un problema de difícil resolución del tipo *NP-Hard* debido a la gran cantidad de variables a ser considerados; por lo que se han propuesto estra-

tegias, algoritmos heurísticos y aproximados que ofrecen soluciones aceptables. A pesar de la gran cantidad de trabajos relacionados, en la práctica se pretenden soluciones rápidas y robustas, antes que de mayor precisión (Ross 2002).



Figura 1: Ejemplo de máquina de corte textil.

Es por esto, que creemos importante poder aprovechar las capacidades de cálculo en paralelo de las PCs de escritorio, con un costo de adquisición relativamente bajo para la implementación de algoritmos estructurados (Alba and Tomassini 2002). En la actualidad, son pocas las propuestas que resuelven el problema en paralelo, entre los que se encuentran el trabajo de (Sławomir et al 2010), pero únicamente para el problema de cortes regulares.

\* Autor en correspondencia.

Correos electrónicos: [juan.damato@gmail.com](mailto:juan.damato@gmail.com) (Juan P. D'Amato),

URL: [www.pladema.net](http://www.pladema.net) (Juan P. D'Amato)

En este trabajo se propone un algoritmo heurístico general que busca resolver el problema de ubicar las piezas en la placa de material, en forma completa, sin solapamientos con la posibilidad de girarlas para obtener un mayor beneficio. Creemos que lo más novedoso de la propuesta es una métrica que mide el beneficio local de incluir un elemento candidato en la solución parcial tendiente a lograr un buen aprovechamiento de todo el material. La solución es general y adaptable, ya que permite variar tanto los parámetros como los métodos para resolver cada etapa. En la **Figura 1** se muestra una máquina utilizada comúnmente para el corte.

Además se plantea una solución pensada para arquitecturas en paralelo que reducen significativamente los tiempos de cálculo. Se ha desarrollado una herramienta de software que facilita el modelado y evaluación de distintos casos y la visualización de los resultados. Dicha herramienta genera una salida en el formato estándar de descripción de gráficos HPGL (*Hewlett & Packard Graphics Language*), por lo cual puede ser leído desde distintos medios de corte. Por último, se comparan los resultados obtenidos con los mismos casos resueltos por una herramienta comercial del sector.

## 2. Estado del Arte

El problema de corte en dos dimensiones (2D), donde las piezas requeridas se seccionan a partir de un planchón o lámina de materia prima, ha sido categorizado, de acuerdo a la forma de dichas piezas, en regulares e irregulares. Los cortes regulares son enfocados, generalmente, a obtener cortes de piezas rectangulares, como secciones de vidrio; mientras que los cortes irregulares manipulan piezas no rectangulares, tal como en la industria del calzado. La cantidad de estrategias estudiadas para resolver el problema de cortes irregulares es aún mayor que para el caso de cortes regulares, debido a la gran dificultad adicional que acarrea las piezas irregulares y sus bordes/contornos impredecibles. Los trabajos relacionados a este tema se encuentran generalmente aplicados a industrias textiles, mobiliarios e industria aeroespacial entre otros (Albano 1977) (Cheng et al 1994). En la **Figura 2** se esquematiza los pasos del proceso de nesting.

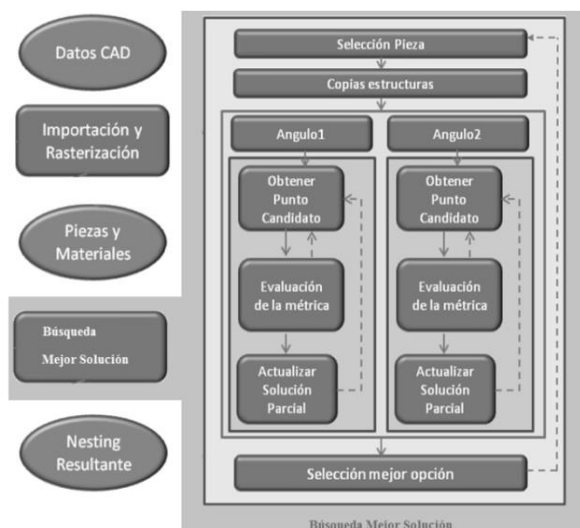


Figura 2: Etapas del proceso de nesting propuesto.

Quizás uno de los primeros trabajos y de los más relevantes es el propuesto por Gilmore y otro (Gilmore and Gomory 1965) utilizando técnicas de programación lineal. La heurística *Bottom-Left* (BL) (Baker et al. 1980) (Downslad et al 2002) y sus variantes, *Bottom-Left-Fill* (BLF) (Chazelle 1983) (Hopper and Turton 1999) aún vigente (Weng and Kuo 2011). Otra heurística basada en la energía potencial total, minimizando la distancia entre el borde la plancha y el centroide de las piezas del *nesting* (Savio et al. 2012).

Recientemente se ha explorado la búsqueda estocástica en combinación con heurísticas existentes o novedosas, planteando métodos híbridos basados en técnicas de inteligencia artificial (Hopper and Turton 2001) (Tay et al. 2002) (Cheng and Rao 2000) (Lee et al. 2008) y en heurísticas de BLF (Siasos and Vosniakos 2014).

Las estrategias para resolver el problema de corte pueden catalogarse de acuerdo al enfoque general propuesto. En la **Tabla 1** se resumen algunos trabajos de acuerdo a la forma del método, evaluando ventajas y desventajas generales.

Tabla 1: Clasificación de los métodos

Método	Ventajas	Desventajas
Programación Lineal (Parada et al 1995) (Abbasi, Sahir 2010)	Muy eficiente cuando el tamaño del problema es moderado.	Tiempo computacional elevado en casos grandes. Dificultad para definir reglas no lineales.
<i>Branch and bound</i> (Cui et al 2006) (Rodrigo et al 2012)	Produce soluciones de alta calidad. Se puede paralelizar.	Converge más lento. Puede llegar a evaluar más posibilidades que las necesarias.
GRASP (Greedy) (Dagli 1988) (Hifi 2001)	Muy rápido y produce soluciones fiables.	Soluciones <i>ad-hoc</i> . Produce menos soluciones exactas que las restantes.
Algoritmo Genético y <i>Simulated Annealing</i> (Hopper y Turton 1999) (Hopper and Turton 2001) (Taiy et al, 2002) (Lai, Chan 1997) (Burke et al 2009) (Junior et al 2013) (Elkeran, 2013)	De implementación relativamente simple. Produce soluciones de mejor calidad en comparación al GRASP.	Lento en convergencia.

## 3. Metodología

El problema de *nesting* alcanza a todo proceso de industrialización que pretenda optimizar la materia prima del producto: sea textil, en diseño de circuitos electrónicos, de productos laminados, o de papel. En la fabricación tradicional del calzado, los obreros calificados realizan manualmente el acomodo

“óptimo” (o *nesting*) de las piezas dentro de la piel, previo al corte. Los operadores van realizando los cortes en uno de los extremos, y corren hacia ellos el material restante hasta que es usado en su totalidad.

La automatización del proceso de acomodo “óptimo” busca desperdiciar la menor cantidad de material aunque subordina los resultados a la geometría propia de las piezas. Especialmente, en la industria del calzado tanto las piezas como el contenedor son espacialmente irregulares, lo que acrecienta la dificultad del problema.

A continuación se formula el algoritmo heurístico propuesto que resuelve esta clase compleja de problemas de corte asociados al *nesting* aplicable a la industria textil y del calzado. En tiempos relativamente cortos, se reduce el remanente en placas de materia prima cuya asimetría, además, restringe la disposición de las piezas.

#### Algoritmo propuesto

El algoritmo combina una heurística constructiva con una búsqueda local para resolver el problema de *nesting* de piezas de geometría irregular. El proceso es incremental. La heurística toma como entrada una secuencia ordenada de las piezas a cortar. En cada paso, selecciona una de ellas y prueba acomodarla sobre el material evaluando el conjunto acotado de ubicaciones factibles, en todas las orientaciones posibles. Entre las disposiciones físicas de la pieza, elige la disposición que introduce menor desperdicio de material entre las mediciones recientemente evaluadas. El proceso incremental finaliza cuando el conjunto de elementos candidatos está vacío.

Para determinar de forma eficiente si dos o más piezas se solapan al momento de colocarse, se utiliza una estructura regular de tipo grilla de celdas, donde cada celda representa una sección cuadrada del material. Cuando se coloca una pieza, la misma se marca como ocupada, a fin de evitar que se superponga con otra. En general, los datos provenientes de los diseños son polilíneas, por lo que es necesario convertir estos datos al formato utilizado por el algoritmo, proceso llamado de *rasterización*.

La solución particular se determinará optimizando una métrica de utilidad del material. El algoritmo comienza tomando una pieza de acuerdo a un criterio de selección e itera probando ubicaciones sobre el material y rotaciones admisibles. A cada resultado encontrado se lo compara con la mejor solución obtenida hasta el momento, de modo que al finalizar encontrará la mejor solución particular para esa pieza en ese material. Incrementalmente se irán sumando a la solución parcial del material, cada una a una las diferentes soluciones particulares. Como esta es una estrategia de optimización localizada, para ampliar el espacio de búsqueda de la solución, se proponen distintos criterios de tomar los elementos de entrada antes de iniciar el proceso. Estos criterios los denominamos *ordenamientos*, y consideran distintos atributos de las piezas (por tamaño de la pieza, por un atributo de calidad o por prioridad en el corte).

El Seudocódigo 1 a continuación es una abstracción del algoritmo secuencial de *nesting*. La entrada la constituye un conjunto de piezas  $P$ , un material  $Q$ , una función *Métrica*  $M$ . Para determinar la mejor ubicación de cada pieza se evalúa cada punto candidato  $C$ , dentro un intervalo de ángulos proporcionados  $[\theta_{\min}, \theta_{\max}]$  con un incremento angular  $\Delta\theta$  correspondiente.

Cada solución parcial se la almacena en un vector  $S_p$ , indexado por posición y ángulo. En las distintas secciones analizamos cada sección del algoritmo aquí propuesto.

```

1.  NESTING ( $P, Q, \theta_{\min}, \theta_{\max}, \Delta\theta$ )
2.   $S \leftarrow$  ordenar  $P$ 
3.  for each  $x \in P$ 
4.     $C \leftarrow$  PuntosCandidatos( $S$ )
5.    for  $\theta_{\min}$  to  $\theta_{\max}$  by  $\Delta\theta$ 
6.      for each  $c \in C$ 
7.         $S_p[\theta, c] = M(x, c, \theta)$ 
8.       $(\theta, c) \leftarrow$  elegir mejor solución ( $S_p$ )
9.       $P \leftarrow P \setminus \{x\}$ 
10.      $S \leftarrow S \cup \{x, c, \theta\}$ 
11.  return  $S$ 

```

Seudocódigo 1 -- para una hoja de material  $Q$ .

El algoritmo ideado entendemos es una estrategia de tipo *Greedy*, el cual en cada paso elige una solución localmente óptima y considera sólo un conjunto de posibles ubicaciones y orientaciones de las piezas sobre el material. El algoritmo intentará en todo momento maximizar la ocupación de una pieza del material utilizando la métrica propuesta. Se supone que todos los materiales son iguales y que cualquier parte puede ser ubicada en cualquier ubicación.

#### 3.1. Discretización del dominio

La rasterización es el proceso mediante el cual las geometrías poligonales de entrada son convertidas a un formato discreto de una grilla regular de celdas equi-espaciadas. Entre las técnicas más conocidas para rasterizar se utiliza el algoritmo DDA (*Digital Differential Analyzer*) (Mollá et al 1992), el cual detecta las celdas del contorno de dicha geometría. Seguidamente se identifican las celdas interiores a cada contorno y se aplica un algoritmo de inundación basado en el *Seed Fill*. El algoritmo selecciona un punto del interior del polígono utilizando una estrategia de *Point-In-Polygon* (PIP) y a partir de esa posición comienza a inundar el resto de la figura. En la **Figura 3** se muestra el resultado del algoritmo.

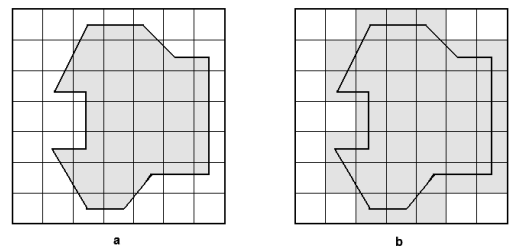


Figura 3 : (izq.) Polígono a analizar. (der.) Celdas ocupadas por dicho polígono.

A su vez, la precisión de la representación y por ende la cantidad y tamaño de celdas en que se divide la geometría de las piezas, llevan a una mayor aproximación en los bordes y mejores “encastrés” entre las piezas. Lógicamente, si bien esto beneficia

al aprovechamiento del material, perjudica el tiempo de respuesta del algoritmo, ya que deberá evaluar una mayor cantidad de posiciones. En la **Figura 4** se observa como a la misma pieza se la aproxima con más celdas.

### 3.2. Métodos de Solución para Nesting

La heurística propuesta genera un acomodamiento de piezas sobre el material que, al ser incremental, permite evaluar la calidad de la solución paso a paso. El conjunto de elementos candidatos está formado por aquellos elementos que se pueden agregar a la solución parcial en construcción sin alterar su factibilidad. Para valorar la calidad de las soluciones parciales, es necesario concertar una métrica que cuantifique el material sobrante a partir de la ubicación, la orientación y la calidad de las distintas piezas en la solución en un momento dado.

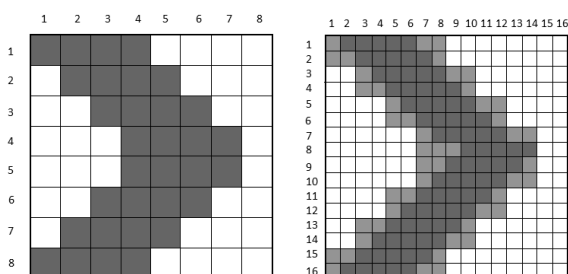


Figura 4: Dos resoluciones de celdas distintas para la misma pieza.

El método de resolución del problema de *nesting* se propone a partir de heurísticas muy documentadas, como la *Bottom-Left* (BL). La heurística BL coloca sucesivamente las piezas rectangulares en la posición de la plancha más profunda posible y luego los desplaza totalmente hacia la izquierda. Por tanto las soluciones están determinadas por el orden en que se introducen las piezas y su posible rotación. El método BLF introduce una mejora que permite el llenado de los huecos producidos en el patrón de empaquetado.

Otras estrategias BL, que requieren de un tiempo cuadrático para resolverse, posicionan cada pieza inicialmente en la esquina superior derecha de la estructura. A continuación se desplaza hasta la posición más profunda posible y desde allí es movida hacia la izquierda tanto como sea posible, repitiéndose esta rutina hasta que la pieza alcance una posición inamovible. La principal ventaja es la creación de áreas totalmente vacías en el material. Una variante de esta estrategia, prioriza los desplazamientos hacia abajo hasta cuando ya no fueran posibles y luego continúa con los movimientos a la izquierda (Liu y Teng 1999).

La heurística *Best-Fit* (BF) (Burke et al. 2009) aplica un ordenamiento dinámico de las piezas en contraposición a las permutación de los elementos propuestas en las heurísticas anteriores. El algoritmo analiza el conjunto de espacios disponibles comenzando por la posición más a la izquierda e inferior posible, posición a la que denominan (LAG), y selecciona por cada lugar la pieza que mejor se adapte, si es que existe alguna. Esto permite al algoritmo tomar decisiones informadas acerca de cuál será el siguiente elemento y dónde se debe asignar. Una implementación natural de esta estrategia se ejecuta en tiempo cuadrático también.

Hopper y Turton (2001) presentan BLD la cual es una estrategia mejorada de BL, donde los objetos son ordenados de

acuerdo a algunos criterios (alto, ancho, perímetro y área) y el algoritmo selecciona aquella pieza que obtenga el mejor resultado. Una mejora es la heurística BLD\* (Lesh et al. 2005) (Lesh and Mitzenmacher 2006), donde se construye una lista de objetos de acuerdo a cierto criterio (alto, ancho u otro) en forma decreciente. La estrategia BLD\* repite distintas ubicaciones con ordenamientos aleatorios específicos hasta que se alcanza un tiempo límite. Su algoritmo mejora otros algoritmos heurísticos y metaheurísticos basados en la estrategia BL reportados en la literatura.

Nuestra propuesta se diferencia de estos y otros trabajos que consideran la ocupación total (Abbasi, Shair 2010); y propone estimar en cuánto se alejan los puntos de la geometría de una pieza con respecto a una posición de referencia. Esta posición de referencia es el punto más abajo y más izquierdo del material (LAG), y mantiene analogía con el esfuerzo que hace el operario al arrastrar el material hacia abajo para realizar el corte. Estos cálculos son muy eficientes, lo que permite realizar múltiples evaluaciones en poco tiempo.

En la ecuación 1, la métrica  $M$  se computa como la suma del producto entre los índices  $i, j$  de las celdas no usadas. Cuanto mayor es el valor de  $M$ , mejor es la ubicación de las piezas. Por lo tanto, el método intentará maximizar esta métrica.

$$M = \sum_{i,j} ijF(i,j) \text{ donde } F(i,j) = \begin{cases} 0 & \text{si cell}_{ij} \text{ fué asignada} \\ 1 & \text{en otro caso} \end{cases} \quad (1)$$

En la **Figura 5** se muestran los resultados de aplicar la métrica en dos configuraciones distintas.

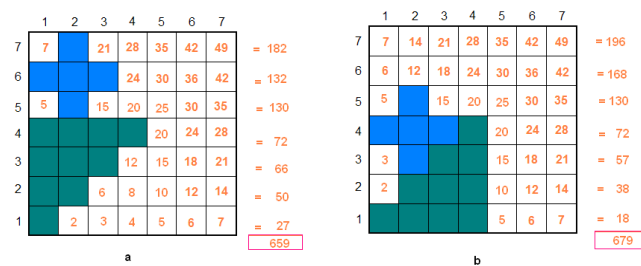


Figura 5: Comparación de la métrica en dos casos posibles de ubicación.

### 3.3. Selección de los puntos candidatos

En cada etapa del *nesting*, es necesario definir la posición sobre la cual se ubicará la pieza. Para ello, se seleccionan las celdas o puntos candidatos dentro del material discretizado, denominado C. De acuerdo a las restricciones de precisión y de tiempo, se puede aplicar uno u otro el criterio de selección.

Se evaluaron distintas estrategias, también conocidas en la literatura, las cuales se detallan a continuación:

- *Bounding Box*: se agregan a C los vértices superior izquierdo e inferior derecho del rectángulo que contiene a la figura de la pieza recientemente ubicada.
- *Raster*: el conjunto P se inicializa con todos los vértices de la grilla. A medida que se encajan las nuevas piezas se eliminan aquellos puntos candidatos ocupados por ellas. A medida que se van colocando piezas disminuye la cantidad de puntos candidatos y el algoritmo resulta ser más eficiente.

- *Raster Bounding*: los puntos candidatos se corresponden con el mínimo rectángulo que encierra las piezas ya ubicadas.

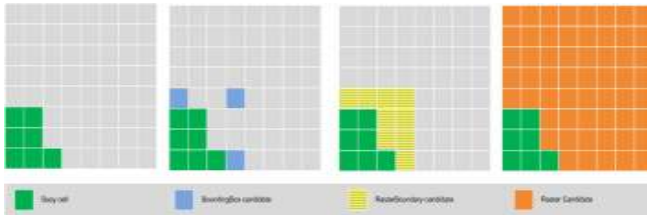


Figura 6: Esquema de selección de los puntos candidatos

En la **Figura 6** se ejemplifican visualmente como se eligen los posibles puntos candidatos a partir de una pieza ya colocada. En verde las celdas ocupadas y las candidatas coloreadas según la estrategia usada: *Bounding Box* (en azul), *Raster Boundary* (en amarillo) y *Raster* (en naranja).

#### 4. Paralelización del algoritmo

La estrategia propuesta tiene complejidad temporal de orden polinomial, pero es posible que encontrar una solución aceptable pueda llegar a demorar varios minutos; típicamente cuando se requiere obtener una precisión del orden de los milímetros y evaluar múltiples orientaciones. En este caso, cobra sentido buscar una forma de acelerar el proceso, es decir, reducir el tiempo de espera proponiendo una extensión que utilice técnicas paralelas. Utilizar algoritmos paralelos es una forma de aliviar los problemas vinculados a tiempos intensivos de ejecución e importantes requerimientos de memoria para resolver instancias complejas de interés.

Para esto, se propone utilizar los múltiples procesadores que se cuentan en las PCs comerciales y convertir el problema secuencial en un problema en paralelo. En esta cuestión, no se han encontrado muchos trabajos relacionados. Por último, se evalúa el rendimiento del algoritmo paralelo comparando el tiempo de ejecución secuencial contra el tiempo correspondiente para el caso paralelo a la hora de resolver un problema particular, resultando en la aceleración real o *speedup*.

##### 4.1. Análisis de las alternativas de implementación

En forma general, para que cualquier tarea pueda resolverse en simultáneo con múltiples hilos de búsqueda o ejecución (*threads*) se deben asegurar ciertas condiciones de independencia tal que:

- cada *hilo* procesa el mismo estado de evaluación,
- la modificación impuesta por un *hilo*, no afecte a los restantes para evitar problemas de sincronismo.

Obtener esta implementación en paralelo no es un proceso directo, ya que es necesario plantear distintas maneras que aseguren una solución equivalente.

Para poder implementar este problema se realizó un análisis detallado de las posibles variantes. En primer lugar, se observa que existe una dependencia temporal que se produce al ubicar una pieza, ya que esta depende de otras ubicadas anteriormente. En este sentido, no es factible su paralelización. Por otro lado, sí es factible definir múltiples hilos de búsqueda que exploran en forma concurrente el espacio de búsqueda donde cada hilo evalúe

en simultáneo las posibles ubicaciones y rotaciones que puede tomar una pieza.

Por lo tanto, para cumplir con las condiciones de independencia, se propuso implementar una adaptación del tipo *map-reduce* (Chue et al. 2007), en el cual se distribuye el problema en  $N$  procesadores y los resultados se analizan en forma secuencial. Para que no existan problemas de acceso a memoria simultáneo, cada *thread* cuenta con una copia del último estado válido.

##### 4.2. Implementación en paralelo

El algoritmo se inicia de manera incremental. Por cada pieza a ubicar, en forma simultánea, se analizan todos los ángulos de giro  $\theta$  del intervalo y cada punto candidato  $C$  del material  $Q$ . Cada combinación se evalúa utilizando la métrica propuesta en una estructura  $M_{Th}$  indexada por identificador de proceso. Una vez que se termina de realizar la evaluación en paralelo de todos los candidatos, se escoge en forma secuencial aquel que ha maximizado la métrica  $M$ . Se actualiza el estado  $S$  y se continúa con la siguiente pieza.

El Seudocódigo 2 presenta la versión modificada del algoritmo. La evaluación de cada ángulo en simultáneo se describe como un *parallelfor*, sintaxis que soportan la mayoría de las APIs de paralelismo como OpenMP o Intel ITBB.

1. NESTING ( $P, Q, \theta_{\min}, \theta_{\max}, \Delta\theta$ )
2.  $S \leftarrow$  ordenar  $P$
3. **for each**  $x \in P$
4.      $M_{Th} \leftarrow$  Copiar estructuras ( $M$ )
5.      $C \leftarrow$  PuntosCandidatos( $S$ )
6.     **parallelfor**  $\theta_{\min}$  **to**  $\theta_{\max}$  **by**  $\Delta\theta$
7.         **for each**  $c \in C$
8.              $M_{Th}[\theta, c, Th] = \text{Métrica}(x, c, \theta)$
9.          $(\theta, c) \leftarrow$  elegir mejor solución ( $M_{Th}$ )
10.          $P \leftarrow P \setminus \{x\}$
11.          $S \leftarrow S \cup \{x, c, \theta\}$
12. **return**  $S$

Seudocódigo 2 – implementación en paralelo.

Con esta estructura, el proceso es distribuido rápidamente, reduciendo efectivamente el tiempo total de cómputo, a costo de requerir una mayor cantidad de memoria. El aporte interesante es que el problema escala rápidamente si se disponen de más cantidad de procesadores, e incluso anda muy bien cuando se requiere mayor precisión ( $\Delta\theta$  pequeño).

#### 5. Resultados

Para validar la estrategia propuesta se emplearon piezas irregulares de distinta forma (o patrón), añadiendo una distancia de seguridad o separación que protege al material de posibles errores provocados por las herramientas de corte. Estos casos se cargaban y parametrizaban en una herramienta desarrollada para el análisis de cortes antes descripto. La misma herramienta cuenta con funcionalidad para visualizar, almacenar y comparar los resultados.

### Caso de estudio #1

El primer caso analizado se extrajo de un sitio web público (ASNS) y plantea la ubicación de 76 piezas irregulares, que fueron generadas a partir de diferentes patrones, sobre un material de 50x50 cm<sup>2</sup>. En la **Figura 7** se muestran los patrones seleccionados para el experimento.

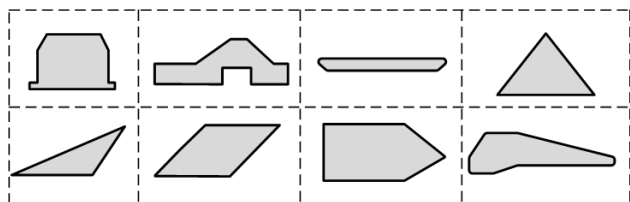


Figura 7: Conjunto de patrones utilizados en las pruebas


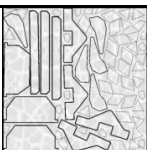
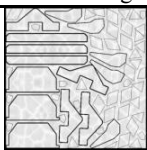
Se estudiaron diferentes aspectos del algoritmo tal como la forma de seleccionar los candidatos y la precisión del corte. Las piezas se ordenaron por área, intentando ubicar siempre las piezas mayores. Para comparar los resultados, se evaluó el aprovechamiento de la materia prima estimada según la tasa de ocupación resultante del material al finalizar la asignación, es decir, la relación entre la superficie del material ocupado respecto a la superficie original. Para las pruebas se utilizó una PC AMD de 6 núcleos a 2.0 GHZ con 4 GB de RAM y Windows 7 OS.

A continuación se detallan distintos experimentos y se compara la estrategia de evaluación propuesta con una de las herramientas más difundidas y utilizadas en el ámbito fabril. Para reducir aún más los tiempos de asignación de nuestro algoritmo, se realiza una etapa de pre-procesamiento que discretiza las piezas en cada posible orientación.

#### 5.1. Estudio de selección de los puntos candidatos

Primeramente, como se muestra en la **Tabla 2**, se evaluó la influencia de los métodos de selección de puntos candidatos (*Bounding Box*, *Raster* y *Raster Boundaring*) en los tiempos de respuesta y asignación de piezas. Se instanció a la variable de precisión de ubicación de la pieza en 1cm y a la distancia de seguridad en 0,5 cm.

Tabla 2: Estudio de ubicación variando el método de selección de puntos

	Bounding Box	Raster	Raster Boundaring
Resultado			
Piezas ubicadas	21	60	62
Tiempo ejecución	13s	161s	66s


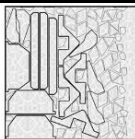
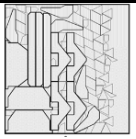
Al comparar los métodos, en ningún caso se pudo ubicar la totalidad de las piezas. Aunque con el método de *Raster Boundaring* se logró no sólo la mejor alocación en cuanto a cantidad de piezas sino también el menor tiempo de respuesta.

#### 5.2. Estudio de variación de la precisión

Para el estudio de variación de la precisión se continuó evaluando el algoritmo sobre el mismo caso de estudio. Ahora adoptando el método *Raster Boundaring* para la selección de puntos candidatos, respondiendo así a los resultados obtenidos en 5.1.

La precisión se probó con valores de tamaño de celda equivalentes a 1 cm, 0,5 cm y 1 mm. Esta exactitud en los valores de precisión de la celda es suficiente no sólo al tipo de material utilizado sino también al tipo de cuchilla metálica pensada. Por otra parte, la distancia de seguridad se corresponde con la mitad de la precisión seleccionada. La **Tabla 3** registra la comparación de ejecuciones del algoritmo de posicionamiento en las distintas precisiones.

Tabla 3: Estudio de ubicación variando la precisión de la discretización

	1 cm	0,5 cm	0,1mm
Resultado			
Piezas ubicadas	62	76	76
Porcentaje utilizado	89%	80%	72%
Tiempo ejecución (s)	35s	127s	708s

Tal como se esperaba, el tiempo de ejecución aumenta cuadráticamente con respecto a la precisión aunque también, a mayor precisión mayor cantidad de piezas ubicadas. En esta configuración, se obtiene una solución conveniente cuando la precisión es de 0,5 cm.

#### 5.3. Estudio de rendimiento

La última evaluación numérica se realizó analizando cómo se comporta el algoritmo cuando se ejecuta en una PC con múltiples núcleos. Se evaluó el caso anterior (caso #1) junto con otro que cuenta con una mayor cantidad de piezas (caso #2). Se tomaron los tiempos totales de procesamiento variando la cantidad de *threads* en que se divide el problema. Se utilizó el algoritmo de *Raster Boundaring*; con una precisión de 1(cm) y una variación de 5° en las rotaciones en ambos casos, corriendo en el equipo antes detallado. Los resultados se muestran en la **Figura 8**.

Los tiempos de respuesta fueron satisfactorios y los resultados se condicen con los obtenidos anteriormente. El algoritmo escala cuasi-linealmente con la cantidad de procesadores, recordando que la actualización de las estructuras y selección de la mejor solución es secuencial. Se obtuvo una aceleración cercana a los 4x para una PC con 6 núcleos.

#### 5.4. Comparación de rendimiento con herramienta comercial

Se compararon rendimientos de la versión del algoritmo heurístico paralelo con la aplicación comercial *Optimizer For Anyshape* (OFA), herramienta que utiliza una solución propietaria

basada en un algoritmo híbrido. Se analizaron tres nuevos casos de estudio, ampliamente estudiados en la literatura, nombrados caso #3, caso #4 y caso #5) corriendo en la PC propuesta.

La comparación de rendimientos se enfocó principalmente en el aprovechamiento del material y el tiempo total de procesamiento. Además se determina el aprovechamiento considerando también el tamaño de las fracciones sobrantes del material. Es de suponer que un retazo de material es mayormente más reutilizable si lo constituye un fragmento mayor que varios pequeños. La configuración de los demás parámetros se ajusta con los mismos valores en la prueba de los dos algoritmos: precisión de la solución en 1 cm y variación en el ángulo de rotación de cada pieza.

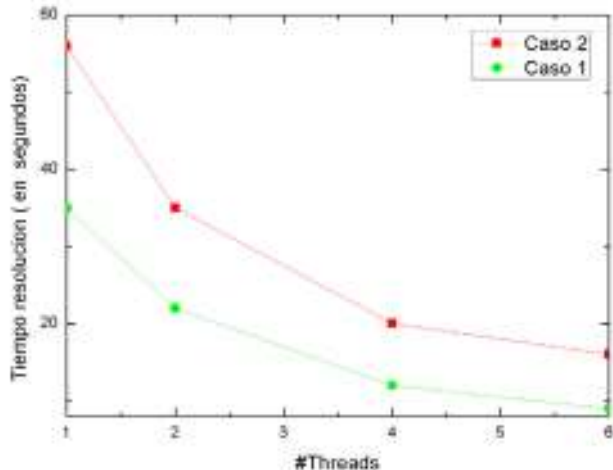


Figura 8: Tiempos de ejecución de su ejecución en paralelo

En la **Tabla 4** se muestran los resultados obtenidos para los tres casos de estudio propuestos. En el caso #3 se acomodan 20 piezas irregulares sobre un material de dimensión 70 x 80 cm. Nuestro algoritmo construye la solución mejorando no sólo los tiempos de respuesta de OFA en 27.6% menos sino también la ocupación del material en casi el 13.2%. Además, con la posibilidad de reutilización del sobrante ubicado sobre la derecha de la plancha. En el caso #4 se acomodan 11 piezas regulares sobre un material de 35 x 70 cm. Nuestra heurística también mejora el aprovechamiento de OFA en 5.14%, mientras que los tiempos de ejecución son destacadamente menores y se reducen a un 35.43% menos que lo que demora en dar una respuesta FOA. Por último, en el caso #5 se acomodan 60 piezas en un material de 20 x 80 cm. Nuestra heurística mejora el aprovechamiento del material en 47%. En la gráfica aparece un sector despejado sobre la parte inferior de la plancha de materia prima. Asimismo, nuestra solución reduce los tiempos de ejecución en un 74.25% que la respuesta de OFA.

## 6. Conclusiones

En este trabajo se ha descrito un algoritmo utilizado en el problema de cortes de materiales en la industria textil y de curtiembre. Este algoritmo fue incluido en una herramienta que le permite a un operario en pocos pasos, realizar una gestión cuasi-óptima del corte. Seguramente, no reemplaza la tarea del experto humano, pero propone una solución viable en poco tiempo. La implementación utiliza datos de entrada y salida de acuerdo a los estándares, lo cual le permite integrarse fácilmente a una cadena de producción.

Los resultados de ubicación han sido satisfactorios, y quizás lo más importante es el bajo costo computacional del algoritmo. A su vez, la implementación en paralelo del algoritmo, ha probado ser escalable y robusta, pudiendo llegar a ser implementado en GPUs para reducir aún más los tiempos de proceso, incluso aumentando la precisión del algoritmo.

Al mismo tiempo, gracias al desarrollo de la herramienta, los operarios pudieron validar visualmente los resultados y realizar pruebas con casos reales, con máquinas comerciales que soportaban el estándar HPGL.

Basándonos en la comparación de algunos casos con la herramienta comercial evaluada, puede afirmarse que la heurística propuesta resulta una alternativa conveniente para la fabricación de calzado y textil. Reduce significativamente los tiempos de ejecución y beneficia el aprovechamiento de material, apuntando a la conservación de extensos sobrantes muy aprovechables en la industria específica hacia lo que está orientada. Como trabajo futuro, se estima la incorporación de una medida de ubicación en base a secciones de calidad en el material.

## English Summary

### A proximal optimization method to the problem of nesting irregular pieces using parallel architectures

#### Abstract

In this paper, a discrete model that solves the two-dimensional cutting problem, usually called nesting, of great interest in the textile industries is presented. The problem consists in finding the best position and orientation of irregularly shaped molds on a material without overlapping, in order to minimize the residual or waste. We propose an adaptive heuristic that evaluates various conditions and constraints of the problem, with a polynomial computational complexity that can be accelerated using multi-core architectures. The proposed methodology is evaluated using known cases of the literature of the area and the resolution times are compared with a commercial tool sector, obtaining very good results. Furthermore, it achieves acceleration up to 4X processing respect to its sequential version.

**Keywords:** Optimization, nesting, textile industry, heuristics, parallelization.


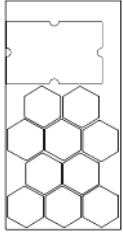




#### Agradecimientos

Este trabajo ha sido parcialmente financiado por el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina.

## Referencias

- Abbasi, J., Sahir, M. Development of Optimal Cutting Plan using Linear Programming Tools and MATLAB Algorithm – Int. J. of Innovation, Management and Technology, Vol. 1, No. 5, pp.483–492, 2010.
- Alba E. and Tomassini M.. Parallelism and evolutionary algorithms. IEEE Transaction on Evolutionary Computation, 6(5):443–462, 2002.
- Albano, A. A Method to Improve Two-Dimensional Layout. Computer Aided Design, 9, 1977.
- ASNS - Nesting Software For Optimum Use Of Materials, LLC Technos, 2013.

Tabla 4: Estudio comparativo de tres casos con una herramienta comercial

	<b>Caso #3</b>		<b>Caso #4</b>		<b>Caso #5</b>	
Tamaño :	70 x 80 cm		35 x 70 cm		20 x 80 cm	
Cantidad	20 piezas		11 piezas		60 piezas	
	OFA	Nuestro método	OFA	Nuestro método	OFA	Nuestro método
Precisión de la solución en cm	1 cm	1 cm	1 cm	1 cm	1 cm	1 cm
Distribución obtenida						
						
Variación del ángulo (°)	1°	1°	5°	5°	5°	5°
Material usado (%)	99.3%	86.2%	83.7%	79.4%	74.5%	40,2%
Tiempo ejecuc. (s)	23,15s	16,76s	12,42s	8,02s	6,6s	1,7s

- Bak S., Błażewicz, J., Pawlak G., Plaza, M., Burke, E., and Kendall G., A Parallel Branch-and-Bound Approach to the Rectangular Guillotine Strip Cutting Problem. *Journal on Computing Winter*, 23(1), 2011.
- Baker B., Coffman E. Jr., and Rivest R. Orthogonal packing in two dimensions. *SIAM Journal on Computing*, 9(4):846–855, 1980.
- Burke, E., Kendall K., and Whitwell G, A Simulated Annealing Enhancement of the Best-Fit Heuristic for the Orthogonal Stock-Cutting Problem, *Journal on Computing*, 21(1), pp.505-516, 2009.
- Chazelle B. The bottom-left bin-packing heuristic: An efficient implementation. *IEEE Transactions on Computers*, 32(8):697–707, 1983.
- Cheng C.H., Feiring, B.R., Cheng, T.C.E. The Cutting Stock Problem – A Survey, *Int. J. of Production Economics*, 36/3: 291–305, 1994.
- Cheng, S.K., Rao, K.P. Large-scale Nesting of Irregular Patterns Using Compact Neighborhood Algorithm, *Journal of Materials Processing Technology*, 103:135–140, 2000.
- Chu, C., Kim, S., Lin Y., Yu, Y., Bradski, G., Ng, A. and Olukotun, A. Map-Reduce for machine learning on multicore. *Neural Information Processing Systems*, 2007.
- Cui, Y., D. He, and X. Song, Generating Optimal Two-Sectional Cutting Patterns for Rectangular Blanks. *Journal of Computers & Operations Research*, 2006. 33(1), pp. 1505-1520, 2006.
- Dagli, C. Cutting Stock Problem: Combined Use of Heuristics and Optimization Methods, *Recent Developments in Production Research Amsterdam*, pp. 500-506, 1988. Dowsland, K.A., Vaid, S., Dowsland, W.B. An Algorithm for Polygon Placement Using a Bottom-left Strategy, *European Journal of Operational Research*, 141:371–381, 2002.
- Dyckho H. and Finke, U. *Cutting and Parking in Production and Distribution*. Physica-Verlag, 1992.
- Elkeran, E. A New Approach for Sheet Nesting Problem Using Guided Cuckoo Search and Pairwise Clustering, *European Journal of Operational Research*, 231:757–769, 2013.
- Gilmore G. and Gomory, R. Multistage cutting stock problems of two and more dimensions. *European J. of Op. Research*, 14(1): 94-120, 1965.
- Hifi M. Exact algorithms for large-scale unconstrained two and three staged cutting problems. *Computational Optimization and Applications*, 18(1):63–88, 2001.
- Hopper E., Turton, B. A genetic algorithm for a 2D industrial packing problem, *Computers & Industrial Engineering* 37(1), pp. 375–378, 1999.
- Hopper E., Turton B. An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. *European Journal of Operational Research*, 128(1):34–57, 2001.
- Jakobs S. On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, 88:165–181, 1996.
- Junior, B.A., Pinheiro, P.R., Saraiva, R.D. A Hybrid Methodology for Nesting Irregular Shape: Case Study on a Textile Industry, 6th IFAC Conference on M. and Control of Production and Logistics (Brazil), pp.15–20, 2013.
- Lai, K., Chan, J. Developing a simulated annealing algorithm for the cutting stock problem, *Journal of Computers ind. Engng*, 32(1), pp. 115-127, 1997.
- Lee, W.-C., Ma, H., Cheng, B.-W. A Heuristic for Nesting Problems of Irregular Shapes, *Computer-Aided Design*, 40/5: 625–633, 2008.
- Lesh N., Marks H., McMahon A., and Mitzenmacher M.. New heuristic and interactive approaches to 2D rectangular strip packing. *ACM Journal of Experimental Algorithmics*, 10:1–18, 2005.
- Lesh N. and Mitzenmacher M.. Bubble search: a simple heuristic for improving priority-based greedy algorithms. *Information Processing Letters*, 97:161–169, 2006.
- Liton, C. A frequency approach to the one dimensional cutting problem for carpet rolls. *Operation Research Quarterly*, 28(1), pp. 927-938, 1997.
- Liu D. and Teng H.. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *European Journal of Operational Research*, 112:413–420, 1999.
- Mollá R., Quirós R., Vivó R. Fixed Point Digital Differential Analyzer Proceedings of Compugraphics '92, pp. 1-5. 14-17, 1992.
- OFA - Optimizer For Anyshape - Samtec Solutions - Website <http://www.samtecsolutions.com/>
- Parada, V., Gómez, A. and De Diego, J.. Exact solutions for constrained two-dimensional cutting stock problems. *European Journal of Operation Research*, 84(1), pp. 633-644, 1995.
- Rodrigo, W., Daundasekera, W. and Perera A. Pattern Generation for Two Dimensional Cutting Stock Problem - *International Journal of Mathematics Trends and Technology*, 3(2), pp.54-62, 2012.
- Ross, P., Schulenburg, S., Marín-Blázquez, J. G., & Hart, E. Hyper-heuristics: learning to combine simple heuristics in bin-packing problems. In *LNCS. Conference on genetic and evolutionary computation*, pp. 942–948, 2002.
- Rui Liu, Xianlong Hong, Sheqin Dong, Yici Cai, Jun Gu, and Chung-Kuan Cheng. Module placement with boundary constraints using o-tree representation. *IEEE Int. Symposium on Circuits and Systems (ISCAS2002)*, 2:871–874, 2002.
- Savio, G., Meneghello, R., Conceri, G. A Heuristic Approach for Nesting of 2D Shapes, in: *Proceedings of the 37th Int. MATADOR Conference (Springer)*, pp.49–53, 2012.
- Siasos A., Vosniakos G. Optimal directional nesting of planar profiles on fabric bands for composites manufacturing. *CIRP Journal of Manufacturing Science and Technology*, vol 7, pp. 283-297, 2014.
- Tay, F., Chong, F. and Lee, F.. Pattern nesting on irregular-shaped stock using Genetic Algorithms. *Engineering Applications of Artificial Intelligence*, 15(1), pp. 551-558, 2002.
- Weng, W.-C., Kuo, H.-C. Irregular Stock Cutting System Based on AutoCAD, *Advances in Engineering Software*, 42/9: 634–643, 2011.