

## Capítulo 4

# ALGORITMO MATRICIAL

### 4.1. INTRODUCCIÓN

En este capítulo se desarrollan los principios del algoritmo matricial para el problema de corte bidimensional que se puede clasificar como un algoritmo heurístico constructivo.

El capítulo comienza explicando la representación matricial de las piezas y las hojas y expone un planteamiento para el formato de dichas matrices con intención de reducir parámetros.

En una segunda parte, se explica el algoritmo de disposición, incluyendo su función de eficiencia y su mecanismo de colocación, y se expone un ejemplo para ver cómo trabaja el algoritmo.

Al final del capítulo se dan algunas de las relaciones matriciales que serán de utilidad para el programa.

## 4.2. CONSIDERACIONES

El presente algoritmo está pensado para series relativamente grandes de piezas irregulares con relativamente muchas piezas de cada forma. En una distribución en la que las figuras se repiten continuamente es habitual que las piezas de la misma forma terminen adoptando orientaciones parecidas; por este motivo se genera un algoritmo que sólo permite giros de  $90^\circ$  en las formas (es decir, la orientación de la pieza se calcula antes y a partir de ella se permiten giros de  $90^\circ$ ) pero que a cambio permite considerar simetrías y huecos de una forma natural.

Para establecer el orden de colocación de las piezas se utiliza una metodología heurística constructiva que, como se veía en el capítulo anterior, divide el problema en una sucesión de problemas en los que los datos de entrada de uno de ellos son los datos de salida del anterior (en cada paso se trata de colocar la pieza con la configuración más eficiente sin tener en cuenta las piezas colocadas anteriormente), el orden de colocación de las piezas no es aleatorio ni depende de una aproximación genética, sino que ensaya todas las formas en todas las configuraciones posibles y, en base a una determinada función de eficiencia, selecciona la forma y la configuración que produce la menor pérdida de material. Como se vio en el Capítulo 3 este tipo de aproximación limita mucho el rango de soluciones accesibles, pero da mayor seguridad a la hora de obtener soluciones eficientes.

El algoritmo parte de la idea inicial de intentar representar la superficie de la hoja mediante una matriz, de forma que si una región de la hoja estuviera ya ocupada por una pieza el valor de la celda correspondiente sería 1, mientras que si la sección de la chapa se encuentra vacía el valor de dicha celda será 0. Esta idea es bastante intuitiva cuando se trabaja con hojas rectangulares, como en las empresas de chapistería en las que, normalmente, la materia prima se sirve en grandes planchas metálicas, aunque se puede adaptar fácilmente al caso de hojas irregulares.

### 4.3. REPRESENTACIÓN DE LAS SUPERFICIES

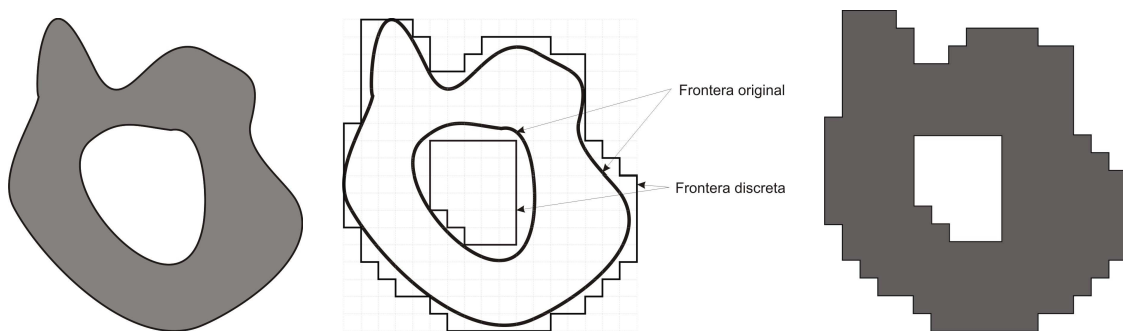
En esta sección se detalla el proceso mediante el cual se generan los datos de las formas que el algoritmo de distribución necesitará para generar las pautas. Dicho proceso se divide en dos apartados:

- discretización de las superficies (generación de las matrices) y,
- reducción de parámetros y formateado.

#### 4.3.1. Discretización de las Superficies

En este apartado se explica de forma general como se obtiene la matriz que representa a cada superficie. El camino concreto para obtener dicha matriz a partir de la información proporcionada por CATIA se explicará más adelante cuando se genere el programa para CATIA.

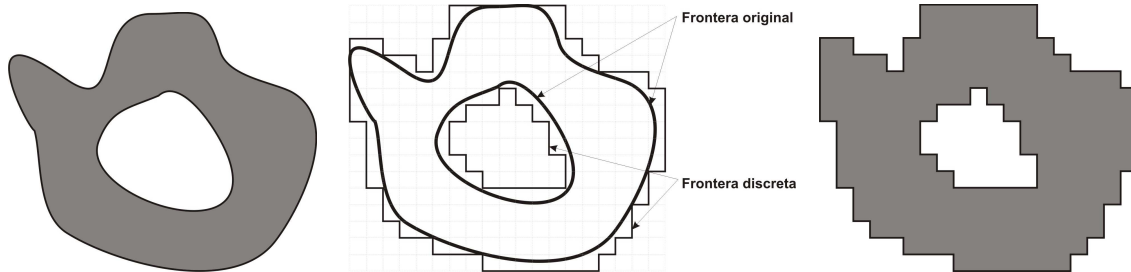
Dada una pieza como la de la Figura 4.1 (a) este proceso pretende obtener una matriz como la de la Figura 4.1 (c) que represente de la manera más fiel posible a la figura original. La manera de obtener dicha matriz varía en función del formato de los datos de entrada, por lo aquí no se entra en detalles.



**Figura 4.1.** (a) Forma original. (b) Discretización mediante mallado.  
(c) Forma discreta.

Como se observa en las Figuras 4.1 y 4.2 la forma discreta final y todo el proceso dependen de la orientación relativa de la pieza. En la Figura 4.1, con una pendiente de  $0^\circ$ , se logra un área discreta de  $203 \text{ uds}^2$  dentro de una matriz de  $306 \text{ uds}^2$

(18 filas y 17 columnas), mientras que en la Figura 4.2, con una inclinación de 30°, se obtiene un área discreta de 201 uds<sup>2</sup> dentro de una matriz de 304 uds<sup>2</sup> (16 filas y 19 columnas).



**Figura 4.2.** Discretización de la forma de la Figura 4.1 con una inclinación de 30°.

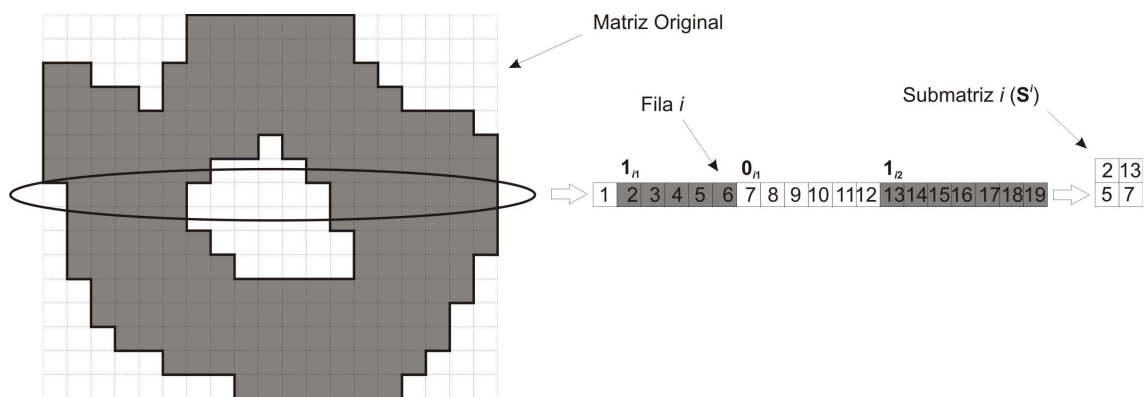
Conviene hacer ver que las piezas aquí representadas corresponden a una discretización muy basta ya que el algoritmo está pensado para trabajar con unidades del orden de milímetros de manera que la pérdida por discretización sea la menor posible. En este ejemplo está claro que la orientación que ofrece menor pérdida es la segunda (30°), aunque podría ocurrir que una orientación produjese menor área discreta y mayor área matricial, o viceversa, así que se establece como norma que la orientación seleccionada sea la que proporcione la matriz de menor área, ya que, como se verá más adelante, es la más conveniente para el algoritmo de distribución, y la pérdida por discretización se ha considerado suficientemente pequeña como para ser irrelevante (las formas de chapistería habituales no son tan irregulares como para generar excesivas pérdidas cuando las unidades de discretización son del orden de milímetros); podemos aproximar la pérdida por discretización mediante  $perd.discret. \approx \frac{perímetro \times 1mm}{área}$ ; así para una superficie circular tendríamos una pérdida de  $2/r$  (siendo  $r$  el radio del círculo, si  $r = 100mm$  la pérdida es del 2%), para un cuadrado se tiene una pérdida de  $4/a$  (siendo  $a$  el lado del cuadrado, si  $a = 100mm$  la pérdida es del 4%).

Este algoritmo se ha diseñado para ser utilizado en hojas en las que las propiedades no varían en el plano, es decir, se admite cualquier inclinación de la pieza respecto a la hoja, aunque se puede adaptar fácilmente a otras aplicaciones (como telas estampadas) ya que la inclinación de la pieza con respecto a la hoja se establece antes de ejecutar el algoritmo, tan sólo habría que limitar las simetrías y los giros.

### 4.3.2. Reducción de Parámetros y Formateado

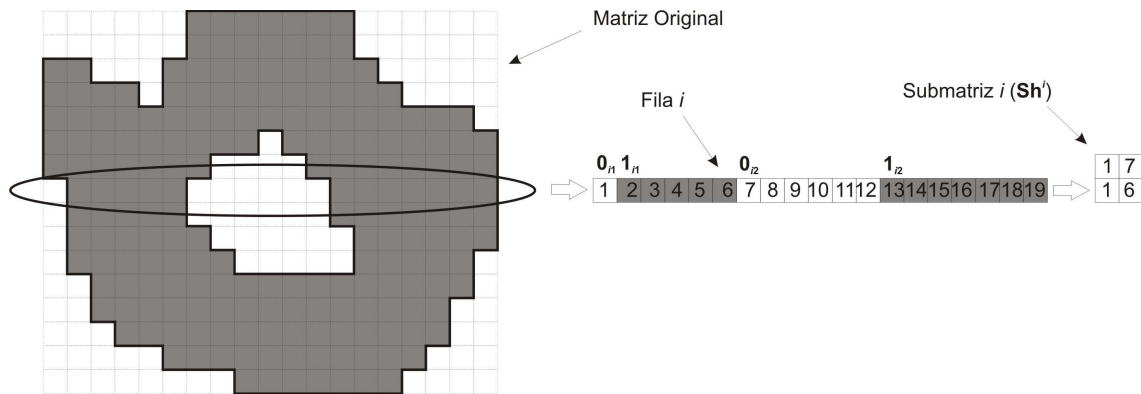
El siguiente paso es depurar la información matricial obtenida y transformarla para que sea útil al algoritmo de disposición. El algoritmo podría funcionar con la información que ya se tiene pero conviene “simplificar” dicha información para reducir el tiempo de operación, sobre todo si se trabaja con unidades de órdenes muy bajos (las matrices obtenidas son cada vez mayores).

Para simplificar esta información, reducimos el contenido de las filas de la matriz de pieza en unas pocas celdas como se observa en el ejemplo de la Figura 4.3. El proceso es el siguiente; para cada fila  $i$  se busca la primera casilla con valor igual a 1 ( $1_{i1}$ ) y se escribe su posición en la primera casilla de la primera fila de la submatriz  $i$  ( $S^i$ ), después se busca la primera celda con valor 0 ( $0_{i1}$ ) situada tras la anterior casilla, se calcula la distancia desde su posición a la de  $1_{i1}$  y se escribe en la primera celda de la segunda fila de la submatriz  $i$ , se repite el proceso anterior empezando a buscar desde la posición del último 0 encontrado hasta que no se halle el siguiente 1 ó el siguiente 0; si el último valor hallado es un 1 ( $1_{ik}$ ), tras anotar en la celda  $S^i_{(1,k)}$  su posición, se anota en la celda  $S^i_{(2,k)}$  la distancia entre la longitud total de la fila (que es la misma para todas las filas de la matriz) más uno y la posición del valor  $1_{ik}$  y se inicia el proceso en la siguiente fila, si el último valor hallado es un 0 ( $0_{ik}$ ) se anota en la celda  $S^i_{(2,k)}$  la distancia entre la posición del valor  $0_{ik}$  y la posición del valor  $1_{ik}$  y se inicia el proceso en la siguiente fila. Al final obtenemos una matriz mucho menor que la original pero con muchas menos columnas.



**Figura 4.3.** Condensación y formato de la matriz de pieza.

Para la hoja, en cambio, el proceso es análogo pero lo que se trata de identificar son los “huecos” en la matriz, así que donde se detectaban 1’s ahora se detectan 0’s y viceversa. Siguiendo con el ejemplo, si la Figura 4.2 representase la hoja se obtendría el resultado que se muestra en la Figura 4.4.



**Figura 4.4.** Condensación y formato de una matriz de hoja de 16 filas y 19 columnas.

Este proceso se realiza con cada una de las ocho orientaciones posibles de cada pieza (cuatro en el plano por dos de la orientación en el eje  $z$ ) y se almacena la información en la matriz de formas. Si la forma ofrece simetrías, se elimina la información duplicada para disminuir el tiempo de ejecución. De la misma manera, se almacenarán en otras dos matrices el número de piezas requeridas de cada forma y el número de configuraciones distintas para cada forma.

#### 4.4. EL ALGORITMO DE DISPOSICIÓN

Ya tenemos la información de las formas, la cantidad y las orientaciones de las piezas, nos falta la información de la hoja. Dado que este algoritmo está pensado para hojas rectangulares sólo nos hará falta saber las dimensiones de las hojas y la matriz de hoja será una matriz de ceros con las dimensiones de la hoja. Si quisiésemos introducir una hoja no rectangular, dicha hoja tendría que seguir un proceso similar al de las piezas para después introducir la matriz de hoja en el algoritmo. Una vez dispongamos de la información de la hoja tenemos todos los datos para iniciar el algoritmo.

El algoritmo para la hoja rectangular se compone de dos etapas, para hojas irregulares no se tiene en cuenta la primera etapa, estas etapas son:

- designación de la primera pieza, y
- disposición recursiva de piezas.

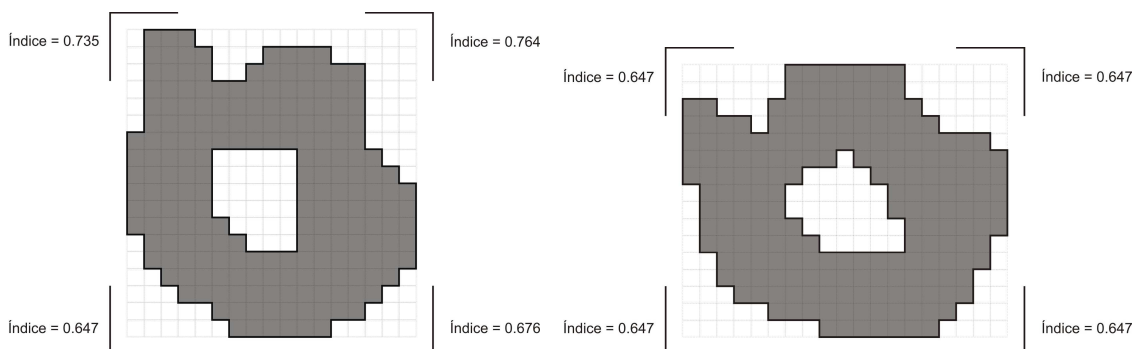
#### 4.4.1. Disposición de la Primera Pieza

En placas rectangulares, cuando se va a colocar la primera pieza, la matriz de hoja está vacía, por lo que no se puede aplicar la función de eficiencia (que se define más adelante), así que para determinar cual es la primera pieza en ser colocada establecemos una función de eficiencia alternativa. Para las hojas irregulares la matriz de hoja no se encuentra vacía y es posible aplicar la función de eficiencia general.

##### 4.4.1.1. Función de eficiencia alternativa

Como la función de eficiencia principal no es de aplicación (ya que la hoja esta vacía) y dado que este algoritmo de disposición llena la hoja empezando desde la esquina superior izquierda, es razonable pensar que la forma que menor espacio vacío deje en el borde de la hoja será la que mejor se adapte a la esquina, por esto se establece como función de eficiencia alternativa una función de densidad de huecos que determina para cada matriz de forma la proporción de celdas que lindan con la hoja cuyo valor es igual a 0, como se puede ver en la siguiente función y en la Figura 4.5:

*índice de huecos = número de celdas fronterizas vacías / número de celdas fronterizas*



**Figura 4.5.** Valor de la función de densidad de huecos en las esquinas de las formas. (a) Inclinación 0°. (b) Inclinación 30°.

Como se observa en la Figura 4.5 la forma con una inclinación de  $30^\circ$  en cualquier esquina y la forma con inclinación de  $0^\circ$  en su esquina inferior izquierda poseen valores más bajos del índice de densidad de huecos; por lo tanto, tras aplicar la función de densidad de huecos a todas las formas, se selecciona la forma cuya esquina posea el menor índice de huecos y se sitúa en la hoja haciendo coincidir dicha esquina con la esquina superior izquierda de la hoja.

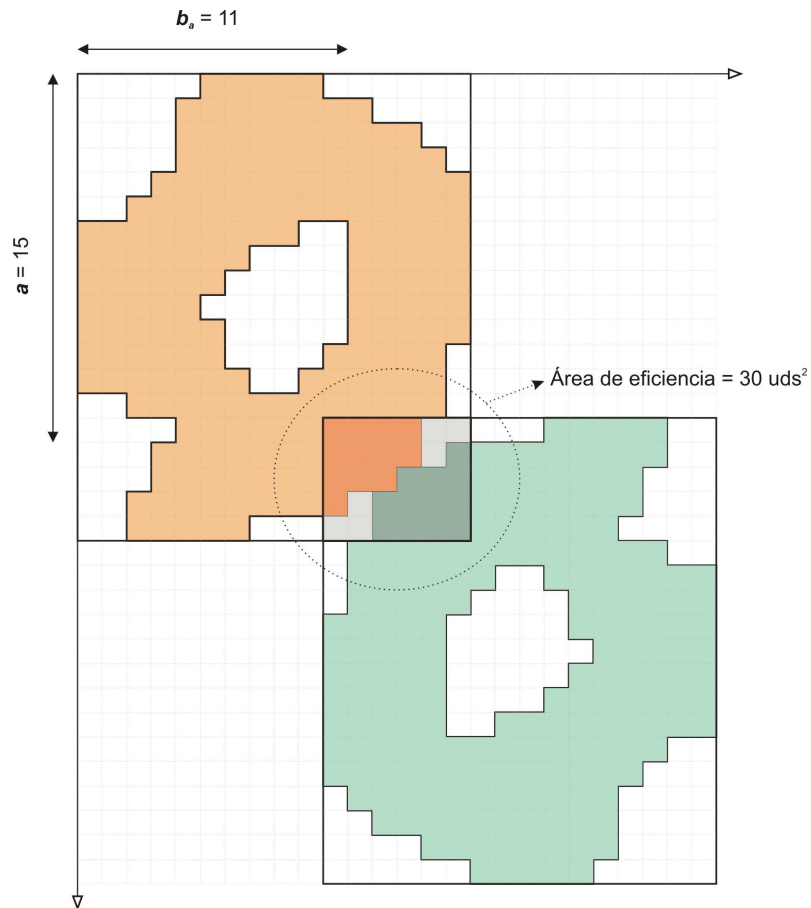
#### **4.4.2. Disposición Recursiva de Piezas**

Una vez situada la primera pieza, ya podemos aplicar la función de eficiencia, así que empezaremos definiendo dicha función.

##### 4.4.2.1. Función de eficiencia

Esta es la parte más delicada del algoritmo, ya que es la función de eficiencia la encargada de seleccionar la forma que colocaremos y la posición y orientación en las que la colocaremos. La función de eficiencia asigna un valor a cada forma en cada posición y orientación (siempre que no se superponga a las piezas colocadas con anterioridad) en función del área matricial superpuesta, para verlo mejor nos ayudamos del siguiente ejemplo; supongamos que tenemos una matriz de hoja rectangular de dimensiones suficientemente grandes para que no se alcancen ninguno de los límites inferior ni derecho, supongamos que ya tenemos situada la pieza del ejemplo de la Figura 4.5 (b) en la matriz de hoja, supongamos también, por comodidad, que la siguiente pieza que queremos situar en la hoja es de la misma forma que la anterior, como se ve en la Figura 4.6.





**Figura 4.6.** Área solapada.

El algoritmo calcula, para todas las piezas, en todas las orientaciones posibles de la pieza y para cada valor de  $a$ , el valor mínimo de  $b_a$  que hace que las piezas no se superpongan (los valores  $a$  y  $b_a$  indican dónde se sumará la matriz de pieza dentro de la matriz de hoja) y determina el valor del área matricial solapada (área de eficiencia en la Figura 4.6); después se seleccionan la pieza y la configuración que poseen el valor más alto de área matricial solapada, se anotan en la matriz de disposición, se resta la pieza colocada de la matriz de cantidades y se empieza de nuevo con el algoritmo hasta que no quede ninguna pieza en la matriz de cantidades o hasta que la hoja esté llena.

El área de eficiencia nunca puede ser mayor que el área de cualquiera de las matrices involucradas, por lo tanto, antes de probar una pieza se comprueba primero si su área es mayor que el área de eficiencia actual, si no es así no hay necesidad de probarla ya que su área de eficiencia será siempre menor que la actual.

#### 4.4.2.2. Superposiciones

Para evitar las superposiciones, el algoritmo se ayuda de las matrices de pieza y hoja debidamente formateadas ( $S$  y  $Sh$ ), el siguiente proceso se realiza para cada valor de  $a$ :

- Dado un valor de  $a$ , a cada fila  $i$  de la matriz de pieza le corresponde una fila  $a + i - 1$  de la matriz de hoja; es decir, a cada submatriz  $i$  de la matriz  $S$  ( $S^i$ ) le corresponde la submatriz  $a + i - 1$  de la matriz  $Sh$  ( $Sh^{a + i - 1}$ ), cada columna de la submatriz  $S^i$  representa un espacio que va a ser ocupado por la pieza que se trata de colocar, así que el algoritmo busca dentro de la submatriz  $Sh^{a + i - 1}$  (en la que cada columna representa una región vacía de la hoja) la columna de menor valor que pueda albergar el espacio representado por la columna de  $S^i$ , para esto debe localizar una columna que cumpla la siguiente ecuación:  $Sh^{a + i - 1}_{(1,j)} + Sh^{a + i - 1}_{(2,j)} \geq S^i_{(1,1)} + S^i_{(2,1)} + b$ . Además se debe cumplir una de las siguientes relaciones: o bien  $Sh^{a + i - 1}_{(1,j)} \leq S^i_{(1,1)}$  en cuyo caso el mínimo valor de  $b$  es 0, o bien  $Sh^{a + i - 1}_{(1,j)} \leq S^i_{(1,1)} + b$  en cuyo caso el mínimo valor de  $b$  es  $b = Sh^{a + i - 1}_{(1,j)} - S^i_{(1,1)}$ ; de entre todas las columnas de  $Sh^{a + i - 1}$  que cumplan con lo anterior se selecciona la de índice más bajo y se calcula el valor de  $b$  para esa columna.

- Ahora hay que intentar colocar la siguiente columna de  $S^i$ ; primero se comprueba si existe una columna de  $Sh^{a + i - 1}$  que cumpla lo anterior con el valor de  $b$  ya calculado, si es así, se continúa con la siguiente columna de  $S^i$ , si no, se calcula un nuevo valor de  $b$  teniendo en cuenta lo anterior y que ahora el valor de  $b$  no puede ser inferior al anterior, se repite el proceso hasta que se hayan explorado todas las columnas de  $S^i$  y, si al final del proceso el valor de  $b$  ha variado con respecto al inicial, se repite todo el proceso pero tomando el último valor de  $b$  como valor mínimo, si el valor de  $b$  no ha variado, se anota dicho valor como  $b_a^i$ .

- Una vez se ha realizado el proceso anterior para todo valor de  $i$ , el siguiente paso toma el mayor de los valores  $b_a^i$  y comprueba en cada fila si todos los segmentos de pieza están bien colocados, si es así, se toma dicho valor como  $b_a$ , si no, se repiten los pasos anteriores pero tomando dicho valor como mínimo valor de  $b$  en cada submatriz.

Este es el proceso más costoso (en cuanto a tiempo de operación) de todo el algoritmo, ya que se ha de realizar para cada pieza en cada configuración. Una vez determinado  $b_a$  se puede calcular el área matricial solapada, al final se seleccionan la pieza y la configuración que aportan la mayor área matricial solapada y se “añade” dicha pieza en dicha configuración a la matriz de hoja.

#### 4.5. EJEMPLO

Siguiendo con el ejemplo de la Figura 4.6, admitiendo un ancho de hoja de 50 unidades, se muestran en la Tabla 4.1 los valores de  $S^i$ ,  $Sh^{a+i-1}$  y  $b_a^i$ , teniendo en cuenta que para  $a + i - 1 > 19$  ( $i > 5$ ) cualquier valor de  $b_a^i$  mayor que 0 es admisible.

$i$	$Sh^{a+i-1}$		$S^i$	$b_a^i$
1	1	15	10	6
	4	36	5	
2	1	15	6	10
	3	36	9	
3	1	14	4	11
	3	37	10	
4	1	12	3	10
	2	39	11	
5	1	8	3	6
	2	43	10	

**Tabla 4.1.** Valores de  $S^i$ ,  $Sh^{a+i-1}$  y  $b_a^i$  para el ejemplo de la Figura 4.6.

Se toma el valor más alto de  $b_a^i$  y se comprueba si con el nuevo valor de  $b$  se cumplen las condiciones de la sección anterior, en este ejemplo se comprueba que se cumplen por lo que se toma el valor  $b_a = 11$  uds. como el mínimo sin superposición, se calcula con dicho valor el área solapada (30 uds<sup>2</sup>.) y, tras realizar la misma operación con todos los valores de  $a$  y todas las piezas y orientaciones posibles, se selecciona aquella disposición que genera la mayor área solapada. Una vez colocada la pieza se

modifica la matriz de hoja y, para colocar la siguiente pieza, se resuelve otra vez el algoritmo pero tomando la nueva matriz de hoja como dato inicial del problema.

#### 4.6. ORIENTACIONES

En esta sección vamos a exponer algunas relaciones matriciales utilizadas para obtener las ocho orientaciones posibles de cada pieza, cuatro en plano por dos del eje Z.

##### 4.6.1. Rotaciones en el Plano XY

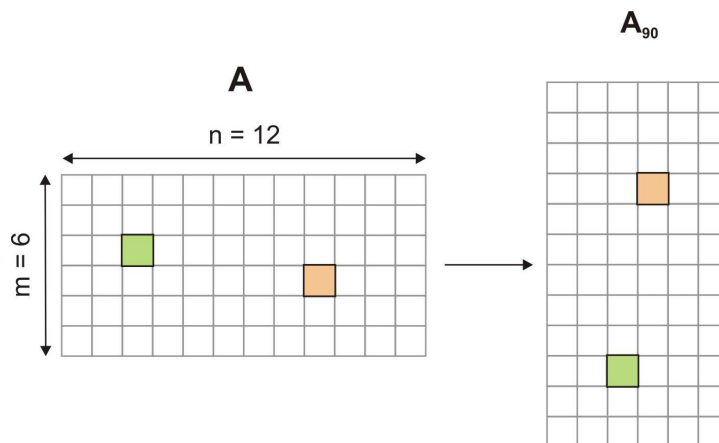
###### 4.6.1.1. Rotar 90°

La rotación se produce en el sentido de las agujas de reloj, como muestra la Figura 4.7, las relaciones matemáticas que ligan sus celdas son:

$$A_{90}(i, j) = A(j, n + 1 - i)$$

$$i \in [1, n]$$

$$j \in [1, m]$$



**Figura 4.7.** Rotación de 90 grados.

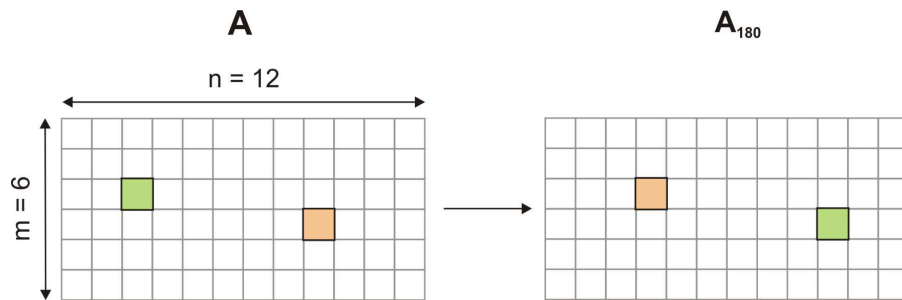
###### 4.6.1.2. Rotar 180°

La rotación se produce como muestra la Figura 4.8, en esta ocasión, las relaciones matemáticas que ligan sus celdas son:

$$A_{180}(i, j) = A(m + 1 - i, n + 1 - j)$$

$$i \in [1, m]$$

$$j \in [1, n]$$



**Figura 4.8.** Rotación de 180 grados.

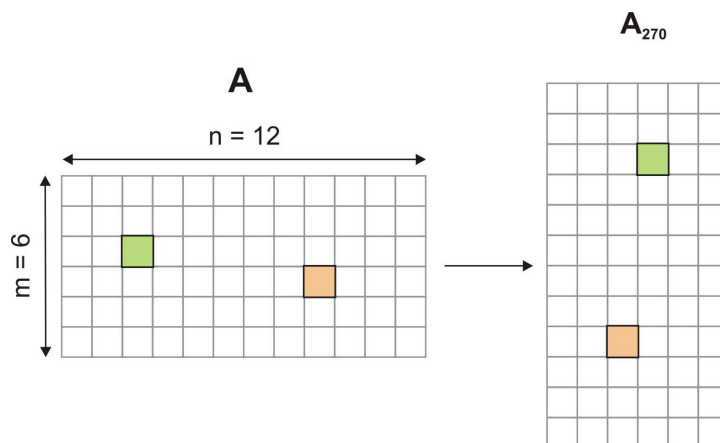
#### 4.6.1.3. Rotar 270°

De igual manera que en los apartados anteriores, la rotación se produce como muestra la Figura 4.9, ahora, las relaciones matemáticas que ligan sus celdas son:

$$A_{270}(i, j) = A(m + 1 - j, i)$$

$$i \in [1, n]$$

$$j \in [1, m]$$



**Figura 4.9.** Rotación de 270 grados.

### 4.6.2. Orientación en el Eje Z

Para acceder a las orientaciones del eje Z negativo de una forma eficiente y no perder información, la orientación según el eje Z se realiza siempre desde la forma original, sin rotaciones, y establece la siguiente secuencia de orientación:

- primero se orienta la forma según el eje Z,
- después se orienta en el plano XY.

#### 4.6.2.1. Orientar según el eje Z negativo

Para orientar una pieza según el eje Z negativo generamos una nueva matriz rotando la forma original 180 grados alrededor del eje X, como se observa en la Figura 4.10. Las relaciones que fijan las celdas de la nueva matriz son:

$$A^{-z}(i, j) = A(m + 1 - i, j)$$

$$i \in [1, m]$$

$$j \in [1, n]$$

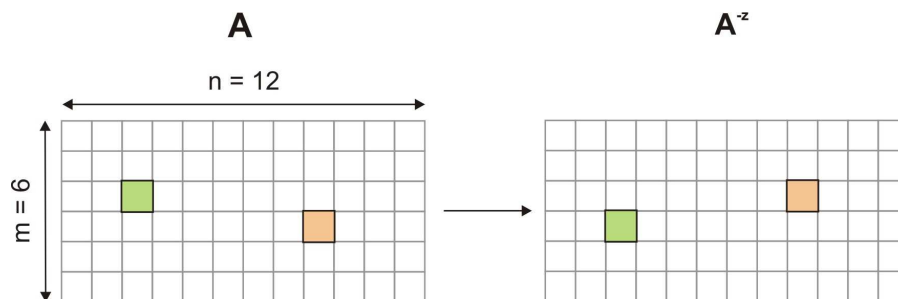


Figura 4.10. Orientación Z negativa.

### 4.6.3. Optimización

En realidad, para mejorar el tiempo de ejecución del algoritmo, éste no calcula cada vez la orientación de cada matriz, sino que, para una forma dada, el algoritmo calcula primero las matrices de cada orientación posible y las guarda como si fuesen formas distintas. Para poder hacer referencia a una orientación concreta, el algoritmo asigna un número a cada una de las orientaciones posibles según la Tabla 4.2.

		Orientación en el plano			
		0°	90°	180°	270°
Orientación en el eje Z	Z positiva	1	2	3	4
	Z negativa	5	6	7	8

**Tabla 4.2.** Correspondencia de orientaciones.

Ahora, para disminuir el número de operaciones que realiza el algoritmo, se comprueba si algunas de las matrices de distinta orientación son idénticas, esto puede ocurrir debido a simetrías o patrones de las formas.

#### 4.6.3.1. $A = A_{90}$

Si se produce la relación  $A = A_{90}$ , entonces  $A = A_{90} = A_{180} = A_{270}$  y  $A^{-Z} = A^{-Z}_{90} = A^{-Z}_{180} = A^{-Z}_{270}$ . La correspondencia queda ahora:

		Orientación en el plano			
		0°	90°	180°	270°
Orientación en el eje Z	Z positiva	1			
	Z negativa	2			

**Tabla 4.3.** Correspondencia de orientaciones;  $A = A_{90}$ .

#### 4.6.3.1.1. $A = A_{90}; A = A^{-Z}$

Si además  $A = A^{-Z}$ , entonces la correspondencia queda:

		Orientación en el plano			
		0°	90°	180°	270°
Orientación en el eje Z	Z positiva	1			
	Z negativa				

**Tabla 4.4.** Correspondencia de orientaciones;  $A = A_{90}; A = A^{-Z}$ .

4.6.3.2.  $A \neq A_{90}; A = A_{180}$

Si  $A \neq A_{90}$  y  $A = A_{180}$ , entonces  $A_{90} = A_{270}$ ,  $A^{-Z} = A^{-Z}_{180}$  y  $A^{-Z}_{90} = A^{-Z}_{270}$ . La correspondencia queda:

		Orientación en el plano			
		0°	90°	180°	270°
Orientación en el eje Z	Z positiva	1	2		
	Z negativa	3	4		

**Tabla 4.5.** Correspondencia de orientaciones;  $A \neq A_{90}; A = A_{180}$ .

4.6.3.2.1.  $A \neq A_{90}; A = A_{180}; A = A^{-Z}$

En este caso,  $A_{90} = A^{-Z}_{90}$  y la correspondencia queda:

		Orientación en el plano			
		0°	90°	180°	270°
Orientación en el eje Z	Z positiva	1	2		
	Z negativa				

**Tabla 4.6.** Correspondencia de orientaciones;  $A \neq A_{90}; A = A_{180}; A = A^{-Z}$ .

4.6.3.3.  $A \neq A_{90}; A \neq A_{180}$

En este caso la correspondencia es la de la Tabla 4.7.

		Orientación en el plano			
		0°	90°	180°	270°
Orientación en el eje Z	Z positiva	1	2	3	4
	Z negativa	5	6	7	8

**Tabla 4.7.** Correspondencia de orientaciones;  $A \neq A_{90}; A \neq A_{180}$ .



4.6.3.3.1.  $A \neq A_{90}; A \neq A_{180}; A = A^{-Z}_{\theta}$

Si  $A = A^{-Z}_{\theta}$ , entonces  $A_{\varphi} = A^{-Z}_{\varphi+\theta}$ . Si  $\theta$  es múltiplo de 90 entonces:

		Orientación en el plano			
		0°	90°	180°	270°
Orientación en el eje Z	Z positiva	1	2	3	4
	Z negativa				

**Tabla 4.8.** Correspondencia de orientaciones;  $A \neq A_{90}; A \neq A_{180}; A = A^{-Z}_{\theta}$ .

#### 4.6.3.4. Resumen

En la siguiente tabla se resumen todos los casos vistos.

Igualdades	Orientación							
	Z Positiva				Z Negativa			
	0°	90°	180°	270°	0°	90°	180°	270°
<b>Ninguna</b>	1	2	3	4	5	6	7	8
$A = A^{-Z}_{\theta}$	1	2	3	4				
$A = A_{180}$	1	2			3	4		
$A = A_{180}; A = A^{-Z}$	1	2						
$A = A_{90}$	1				2			
$A = A_{90}; A = A^{-Z}$	1							

**Tabla 4.9.** Resumen de correspondencias.

## **4.7. RESUMEN**

En este capítulo se ha visto cómo trabaja el algoritmo matricial, que se ha clasificado como un algoritmo heurístico constructivo.

El capítulo comienza describiendo el modo de representación de las piezas y estableciendo el formato que tendrán las matrices de las piezas y la hoja.

Después se ha dividido el algoritmo en dos etapas, colocación de la primera pieza y disposición recursiva de piezas, y se han descrito los parámetros fundamentales del algoritmo en cada etapa, cómo la función de eficiencia y el procedimiento para resolver las superposiciones. En la Sección 4.5 se plantea un ejemplo para ver mejor cómo se resuelven las superposiciones.

El capítulo termina dando algunas relaciones matriciales de utilidad a la hora de orientar las formas, y definiendo las orientaciones posibles de cada forma en función de sus simetrías o patrones.