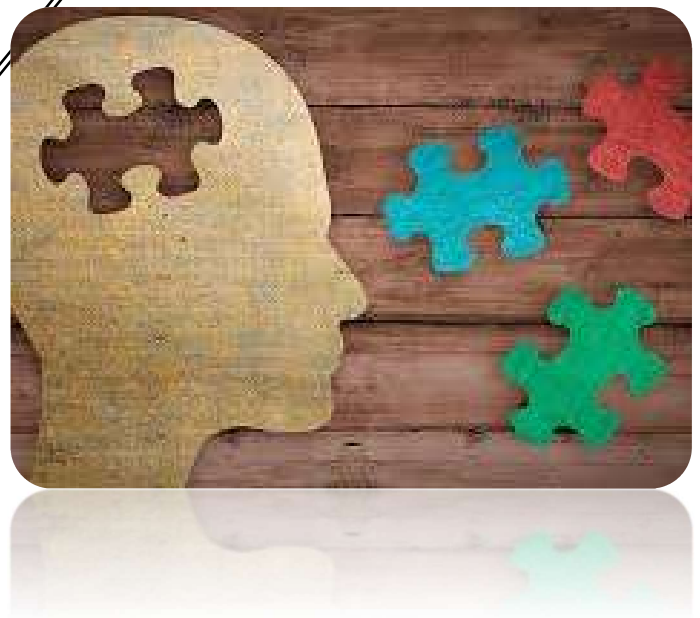


# PROBLEMA DE CORTE DE PIEZAS RECTANGULARES

Taller de Simulación de Sistemas



Carrera: Licenciatura en Ingeniería de Sistemas  
Modelos de Simulación de Sistemas

# Problema de Corte de Piezas Rectangulares

## Actividad Numero Dos

### 1. Introducción

El problema de corte de piezas rectangulares, también conocido como el problema de empaquetamiento rectangular 2D o el problema de corte de stock rectangular, consiste en disponer un conjunto dado de piezas rectangulares más pequeñas dentro de una superficie rectangular mayor (el contenedor o la plancha de material) de manera eficiente, buscando optimizar algún criterio específico. Los objetivos comunes incluyen minimizar el desperdicio de material, maximizar el número de piezas colocadas o minimizar el número de contenedores utilizados.

Desde una perspectiva computacional, este problema se enmarca dentro del campo de la optimización combinatoria. El desafío radica en encontrar la mejor configuración entre una vasta cantidad de posibles ubicaciones y orientaciones para las piezas, lo que propicia una solución computacional. Esto significa que encontrar una solución óptima para instancias grandes puede ser costoso en términos de tiempo de cálculo. Por lo tanto, la investigación y la aplicación se centran en el desarrollo y uso de algoritmos capaces de encontrar soluciones óptimas de buena calidad en un tiempo razonable.

### 2. Elementos Clave del Problema

- Superficie Rectangular Mayor (Contenedor/Plancha): Definida por su ancho ( $W$ ) y alto ( $H$ ).
- Conjunto de Piezas Rectangulares Menores: Una lista de ' $n$ ' piezas, donde cada pieza ' $i$ ' tiene un ancho ( $w_i$ ) y un alto ( $h_i$ ). Pueden existir restricciones sobre la cantidad de cada tipo de pieza.
- Restricciones:
  - No Superposición: Las piezas no deben solaparse.
  - Dentro de los Límites: Cada pieza debe estar completamente dentro de la superficie mayor.
  - Orientación: Puede estar fija o permitir rotación de 90 grados.
- Función Objetivo: El criterio a optimizar (e.g., minimizar desperdicio, maximizar piezas).

### 3. Ejemplos Prácticos y Aplicaciones

El problema de corte rectangular tiene aplicaciones significativas en diversas industrias:

- Industria Textil y de la Confección: Optimización del corte de patrones de tela de rollos para minimizar el desperdicio.
  - Ejemplo: Disposición de patrones de diferentes tallas de ropa en un rollo de tela para un pedido específico.

- Industria del Vidrio: Optimización del corte de piezas de vidrio de láminas grandes para pedidos de ventanas, espejos, etc.
  - Ejemplo: Corte de diferentes dimensiones de vidrio de una lámina estándar para cumplir con los pedidos de los clientes.
- Industria del Papel y la Impresión: Maximización del número de hojas de tamaño deseado que se pueden cortar de pliegos de papel estándar.
  - Ejemplo: Imposición y corte de páginas de un folleto en un pliego de impresión para minimizar el desperdicio de papel.
- Industria de la Madera y el Mueble: Optimización del corte de piezas de madera de tabloncillos para la fabricación de muebles.
  - Ejemplo: Obtención de tableros, patas y travesaños para un lote de mesas a partir de tabloncillos de madera disponibles.
- Logística y Almacenamiento: Empaquetamiento eficiente de cajas rectangulares en paletas o contenedores (simplificado a 2D en algunos casos).
  - Ejemplo: Disposición de cajas de diferentes tamaños en la base de una paleta para maximizar la utilización del espacio.

Usted puede escoger el ámbito de aplicación a manera de caso de estudio en su actividad.

#### 4. Algoritmo Básico: Primer Ajuste Decreciente (FFD)

El algoritmo Primer Ajuste Decreciente es una solución para intentar colocar las piezas dentro de la superficie mayor:

Entrada:

- Superficie rectangular mayor: Ancho (W), Alto (H)
- Lista de piezas rectangulares:  $L = [(w_1, h_1), (w_2, h_2), \dots, (w_n, h_n)]$

Salida:

- Una posible disposición de las piezas o indicación de fallo.

Pasos:

1. Ordenar: Ordenar la lista 'L' de piezas de forma no creciente según su área ( $w_i * h_i$ ).
2. Inicializar Espacio Libre: Considerar la superficie mayor como el espacio libre inicial (posición (0,0), ancho W, alto H). Se puede mantener una lista de espacios libres.
3. Iterar Piezas: Para cada pieza ( $w_i, h_i$ ) en la lista ordenada:
  - a. Buscar Ajuste: Recorrer la lista de espacios libres. Para el primer espacio libre con ancho  $\geq w_i$  y alto  $\geq h_i$ .
    - i. Colocar Pieza: Colocar la pieza en la esquina inferior izquierda del espacio libre.

- ii. Actualizar Espacios Libres: Dividir el espacio libre donde se colocó la pieza en uno o dos nuevos espacios libres restantes.
- iii. Marcar Pieza como Colocada: Registrar la posición de la pieza.
- iv. Romper Búsqueda: Pasar a la siguiente pieza.

b. No Encontrado: Si no se encuentra ningún espacio libre adecuado para la pieza, no se puede colocar con esta estrategia.

4. Resultado: La lista de piezas colocadas y sus posiciones.

Este algoritmo es una propuesta inicial y podría aportar a una solución óptima. Sin embargo, la gestión precisa de los espacios libres es considerable para su implementación y su discernimiento debería ayudar en esa tarea. El enfoque depende de los requisitos de la aplicación en términos de calidad de la solución y tiempo de cómputo permitido.

## **6. Conclusión**

Se recomienda una adecuada comprensión de los elementos del problema, los objetivos de optimización y que realice una propuesta algorítmica implementable ajusta al caso de estudio que escogió en el punto 3.

## **7. De la Presentación**

La presentación de esta actividad consiste en:

- Documentación técnica y de usuario (formato digital e impreso).
- Diapositivas del trabajo (formato digital).
- Código fuente(formato digital).
- Librerías necesarias(formato digital).

La fecha de presentación de esta actividad es hasta el día 7 de abril de los corrientes, según condición de plazos especificados en classroom.

En la fecha ya especificada habrá defensa del trabajo presentado en esta actividad.

Se subirán al classroom de la clase hasta la fecha indicada lo especificado en los puntos previamente descritos. Así mismo, se deberá presentar al ingreso de la clase el formato físico en impreso acompañado del formato electrónico del contenido ya enunciado. También deberán acompañarse de su equipo a fin de validar la realización del trabajo.

## **8. Resultado Esperado de la Actividad Dos**

La actividad tendrá como resultado:

- Propuesta respecto del modelo de Simulación, es decir algoritmo con fundamentación matemática pertinente.
- El modelo de Simulación del proyecto deberá reproducirse en Excel, así como en un programa desarrollado en Java, Python y/o lenguaje R. A fin de validar resultados entre las aplicaciones desarrolladas.