Reporte de Modelo de Optimización II

Alberto Helguera Fleitas C-412
José Gabriel Navarro Comabella C-412
Juan David Menéndez del Cueto C-412
Karl Lewis Sosa Justiz C-412

November, 2020

Presentación del problema

Se plantea el siguiente problema: Dado un conjunto de objetos geométricos de distintas formas y que pueden tener distintos colores o no, ¿cuál es la mejor forma, visualmente hablando, de ubicarlos en una región rectangular?

Este problema tiene 2 puntos claves a la hora de resolverlo computacionalmente: el primero son los objetos geométricos, esto es debido a la gran variedad de figuras que existen y a la dificultad de representarlas de una manera que sea conveniente a la hora de resolver el problema; y el segundo es definir cuál es la mejor forma de ubicarlos.

El primer punto es solucionado de la siguiente manera: cada objeto es sustituido por el rectángulo, con los lados paralelos a lo ejes de coordenadas, de menor área que lo contiene. Mientras que la mejor forma de ubicarlos es definida como la forma en la cual se maximice la utilización del área y los objetos sean empaquetados ortogonalmente y sin solapamientos.

Entonces el problema original queda reducido al siguiente: Dado un conjunto de rectángulos, encontrar la forma de empaquetarlos ortogonalmente y sin solapamientos en una región rectangular tal que se maximice la utilización del área.

Modelo matemático

Se tiene un conjunto de n rectángulos r_i con dimensiones $w_i \times h_i,\ i = 1, ..., n$ y un rectángulo más grande con dimensiones $W \times H$ donde ubicarlos de la manera planteada en la sección anterior.

Para saber la posición en que fue ubicado cada rectángulo, se define

$$p_i = \begin{cases} 1 & \text{si} \quad r_i \textit{ fue ubicado horizontalmente} \\ 0 & \text{si} \quad r_i \textit{ fue ubicado verticalmente} \end{cases}$$

Sean:

el conjunto $\{(x_i, y_i),\ i = 1, ..., n\}$, donde (x_i, y_i) representa la localización de la esquina inferior izquierda del rectángulo i, una ubicación de los n rectángulos en el rectángulo más grande;

U, el conjunto formado por los rectángulos que tienen un rectángulo ubicado por encima, y para cada elemento de U se define U_i como el recángulo ubicado inmediatamente encima;

R, el conjunto formado por los rectángulos que tienen un rectángulo ubicado a su derecha y para cada uno sus elementos se define R_i como el rectángulo ubicado inmediatamente a su derecha;

$w_{max} = \max\ x_i + w_i * p_i + h_i * (1 - p_i),\ i = 1, ..., n;$

$h_{max} = \max\ y_i + h_i * p_i + w_i * (1 - p_i),\ i = 1, ..., n;$

Entonces podemos modelar el problema como encontrar la ubicación $\{(x_i, y_i),\ i = 1, ..., n\}$ tal que:

$$\max\ f = W * H - w_{max} * h_{max}$$

$$\text{s.a.}$$

$$0 \leq x_i \leq W - (w_i * p_i + h_i * (1 - p_i))\ \ i \in \{1, ..., n\}$$

$$0 \leq y_i \leq H - (h_i * p_i + w_i * (1 - p_i))\ \ i \in \{1, ..., n\}$$

$$x_i + w_i * p_i + h_i * (1 - p_i) \leq x_{R_i}\ \ i \in R$$

$$y_i + h_i * p_i + w_i * (1 - p_i) \leq y_{U_i}\ \ i \in U$$

$$x_i \in \mathbb{R}$$

$$y_i \in \mathbb{R}$$

$$p_i \in \{0, 1\}$$

Las dos primeras restricciones del modelo aseguran que todas los rectángulos estén contenidos en el rectángulo más grande, y la tercera y la cuarta garantizan que no exista ningún solapamiento entre los rectángulos a ubicar.

Notemos además que la función objetivo del modelo es equivalente a

$$\min\ g = w_{max} * h_{max}$$

dado que $W * H$ es un valor fijo una vez planteado el problema.

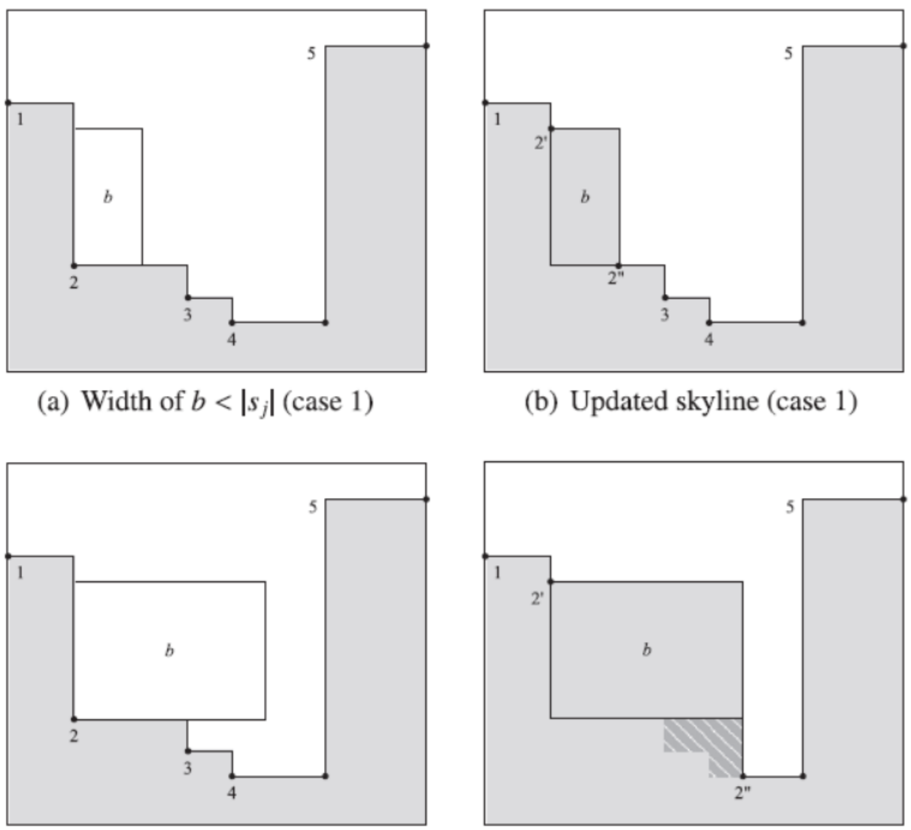
Algoritmo

El algoritmo utilizado en la resolución de este problema es una heurística *greedy*. Esta heurística toma como entrada una secuencia de rectángulos, las dimensiones W x H del rectángulo donde se desean ubicar y el parámetro de control max_spread.

Representamos un patrón de empaquetamiento actual mediante un horizonte rectilíneo, y este puede expresarse como una secuencia de k segmentos de línea horizontal $(s_1, s_2, ..., s_k)$ que satisfacen las siguientes propiedades: (1) la coordenada y de s_j es diferente de la coordenada y de s_{j+1}, j = 1, ..., k-1; y (2) la coordenada x del extremo derecho de s_j es la misma que la coordenada x del punto final izquierdo de s_{j+1}, j = 1, ..., k-1. El patrón de empaque vacío inicial está representado por un segmento de una sola línea correspondiente a la parte inferior de la hoja.

Nuestra heurística coloca los rectángulos uno por uno. Cada rectángulo se coloca con su esquina inferior izquierda tocando un punto final izquierdo o su esquina inferior derecha tocando un punto final derecho de un segmento de línea s_j en el horizonte. El extremo izquierdo de un segmento s_j es una posición candidata si y solo si s_{j-1} es mayor que s_j; de manera similar, el extremo derecho de un segmento s_j es una posición candidata si y solo si s_{j+1} es mayor que s_j. Tenga en cuenta que el extremo izquierdo de s_1 y el extremo derecho de s_k siempre son candidatos.

Cuando se coloca un rectángulo b en un segmento s_j, el horizonte se actualiza. Esto se hace en dos pasos. El primer paso instancia un nuevo segmento de línea correspondiente al borde superior de b y actualiza los segmentos existentes afectados. Hay dos casos dependiendo de si el ancho de b es mayor que $s+j$; la figura 1 muestra un ejemplo de los dos casos en los que se coloca un rectángulo b en el extremo izquierdo de s_j, junto con cómo se actualiza el horizonte. Tenga en cuenta que el área sombreada en la Fig. 1 (d) se considera espacio desperdiciado porque la heurística nunca considerará colocar rectángulos en ella.



(a) Width of $b < |s_j|$ (case 1) (b) Updated skyline (case 1)

(c) Width of $b > |s_j|$ (case 2) (d) Updated skyline (case2)

Figure 1: Actualización del horizonte, paso 1

En el segundo paso, comprobamos cada segmento de línea que es menor que sus dos segmentos adyacentes, lo que llamamos un segmento menor localmente (para el primer y último segmentos, sólo los comparamos con su un segmento adyacente). Si no hay rectángulos sin colocar que se puedan colocar en ese segmento, lo elevamos a la altura de su segmento adyacente inferior y los fusionamos. Esto se repite hasta que se consideran todos los segmentos de línea.

Nuestra heurística considera todas las ubicaciones posibles, elige la mejor y coloca el rectángulo en esa posición. Esto se repite hasta que no se puedan colocar más rectángulos en ninguna posición o se hayan colocado todos los rectángulos. Primero establecemos nuestra función de evaluación y luego explicamos las motivaciones detrás de cada una de sus componentes.

Considere una ubicación factible (p, b), donde p es una posición candidata en el segmento de línea s_j. Al evaluar una ubicación, el algoritmo examina el horizonte resultante después del primer paso de nuestro procedimiento de actualización del horizonte. La función de evaluación para una ubicación factible dada (p, b) viene dada por el siguiente conjunto de reglas de prioridad:

1. (restricción de extensión) La extensión de un horizonte es la diferencia entre las coordenadas y de sus segmentos de línea más altos y más bajos. Si la extensión del horizonte resultante después de colocar el rectángulo en esa posición es mayor que el parámetro de entrada max_spread, entonces la ubicación se considera inviable y se rechaza inmediatamente.

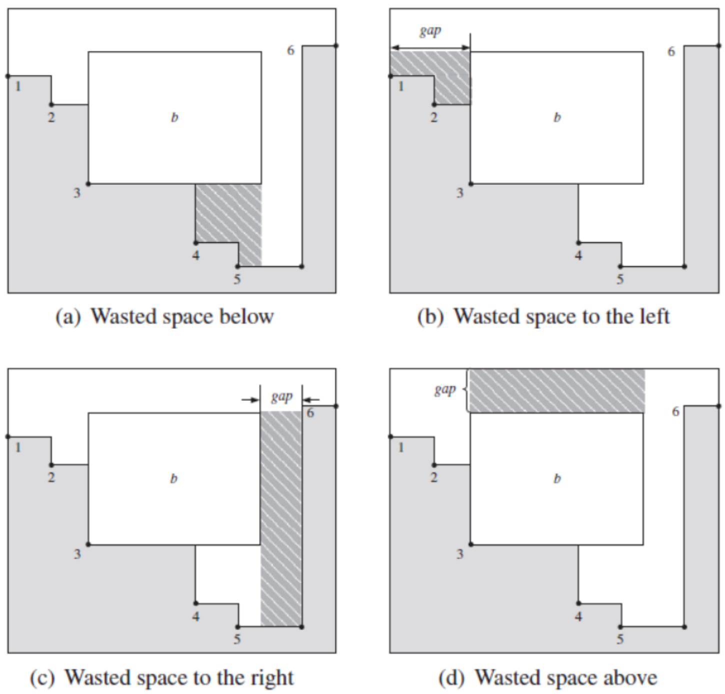
2. (único encaje) Si el rectángulo b es el único rectángulo que queda que puede encajar en p, entonces (p, b) tiene la prioridad más alta.

3. (desperdicio local mínimo) Si no existen tales casos o más de uno, seleccionamos la ubicación que minimiza la cantidad de espacio local desperdiciado. Esto se calcula como el volumen total de espacio desperdiciado de los cuatro tipos dados en la Fig. 2. Sea w_min y h_min el ancho y alto mínimos, respectivamente, de los rectángulos restantes excluyendo b. El área sombreada en el caso (a) siempre se considera espacio desperdiciado. Las áreas sombreadas en los casos (b) y (c) se consideran desperdiciadas si la longitud del espacio es menor que w_min. Finalmente, el área sombreada en el caso (d) se desperdicia si el espacio es menor que h_min.

4. (número máximo de aptitud) Si hay un empate, se prefiere la ubicación con el número de aptitud más alto. El número de aptitud de una ubicación es el número de lados del rectángulo que coincide exactamente con el segmento que toca en el horizonte. El lado inferior de un rectÃąngulo es una coincidencia exacta si su ancho es igual a la longitud de s_j. El lado izquierdo (resp. Derecho) de un rectángulo es una coincidencia exacta si su altura es igual a la coordenada y de s_{j-1} (resp. s_{j+1}) menos la coordenada y de s_j. Cualquier lado de un rectÃąngulo que toque el lado izquierdo, derecho o inferior del rectángulo donde se desean ubicar no se considera una coincidencia exacta a menos que llene todo el espacio restante en ese lado. Sin embargo, cuando el borde superior de un rectángulo colocado toca la parte superior del rectángulo donde se desean ubicar, se considera una coincidencia exacta. El número de aptitud puede ser 0, 1, 2, 3 o 4.

5. (el primero en la secuencia) Si hay un empate, se prefiere la ubicación que involucre el primer rectángulo en la secuencia de entrada. Si el rectángulo se puede colocar en varias ubicaciones, se prefiere la ubicación con la coordenada y más pequeña, luego con la coordenada x más pequeña.

Probamos ambas orientaciones para cada rectángulo, y la mejor orientaciÃşn determina la evaluación para la ubicación. Si ambas orientaciones tienen evaluaciones idénticas basadas en las cinco reglas anteriores, preferimos la orientación correspondiente a la entrada original.



(a) Wasted space below (b) Wasted space to the left

(c) Wasted space to the right (d) Wasted space above

Figure 2: Espacio desperdiciado

Hay dos motivaciones principales para implementar una restricción de extensión. En primer lugar, sin esta restricción, nuestra heurística *greedy* tendería a producir una torre alta de rectángulos a lo largo del lado izquierdo hasta que se alcanza la altura del rectángulo donde se desean ubicar. Este es un fenómeno posiblemente indeseable que se puede controlar con la restricción de extensión. En segundo lugar, diferentes valores de max_spread hacen que nuestra heurística se comporte de manera diferente: un valor de max_spread alto permite torres altas, mientras que un valor de max_spread bajo hace que la heurística coloque rectángulos capa por capa. Al probar un rango de valores max_spread, podemos generar empaques más diversos, explorando así el espacio de búsqueda más a fondo. Los valores de max_spread probados son $\{mh, mh + (H - mh) * \frac{1}{3}, mh + (H - mh) * \frac{2}{3}, H\}$, donde mh es la altura del rectángulo más alto.

Recuerde que solo realizamos el primer paso de la rutina de actualización del horizonte cuando realizamos nuestra evaluación. Esto detecta espacio desperdiciado en segmentos adyacentes de los tipos mostrados en el ejemplo de la Fig. 2 pero no tiene en cuenta otros tipos de espacio desperdiciado. La restricción del único encaje ayuda a abordar esta omisión. Se basa en la observación de que si el único rectángulo b que se puede colocar en un segmento de línea s_j se coloca en otro lugar, entonces se desperdicia el área justo encima de s_j, que podría ser grande. Por tanto, es probable que colocar b en el segmento s_j reduzca la cantidad total de espacio desperdiciado.

El componente clave de nuestra función de evaluación es la regla de desperdicio local mínimo. Está motivada por la suposición natural de que, si la cantidad de espacio desperdiciado se minimiza en cada etapa del proceso, es más probable que los rectángulos restantes no colocados se puedan colocar. Tenga en cuenta que nuestra medida de residuos local es solo una medida aproximada de la cantidad real de espacio desperdiciado.

La regla del número máximo de aptitud favorece las ubicaciones que dan como resultado un horizonte "más regular" que contiene menos segmentos de línea, lo que es más probable que permita la ubicación de rectángulos más grandes sin producir espacio desperdiciado.

Dadas varias ubicaciones con el menor desperdicio y el mayor número de aptitud, el primer criterio de secuencia introduce una regla de desempate. Esto nos permite considerar diferentes formas de priorizar la ubicación de los rectángulos utilizando diferentes secuencias de entrada.