Floor planning como un Geometric program

Diego F. Fonseca. V. *

Resumen. Optimizar el área que puede tener el menor rectángulo que alberga un numero de bloques dados a priori con áreas dadas de antemano y sujeto a restricciones en el largo y disposición de los bloques en el plano, constituye un problema de optimización el cual formularemos matemáticamente, evidenciaremos que en esa primera formulación no es convexo pero constituye un *Geometric program* lo que permitirá darle una formulación convexa. Ademas, se formulara un análogo de esta situación pero con bloques en el espacio (3D).

Palabras clave: Optimización conveza, grafos, Geometric program.

1 Introducción

La frase "floor planning" puede traducirse como "planificación de piso", para los ingenieros y arquitectos esta es una situación importante ya que influye en los costos de los proyectos de construcción, supongamos que el arquitecto esta trabajando en la construcción de una torre de apartamentos, este diseña una misma distribución para todos los pisos, es decir, determina que las habitaciones, baños, sala y cocina deben estar bajo una misma distribución y que no se puede alterar, ademas establece que cada uno de estos lugares en el piso debe tener un área determinada que es inmodificable, pero permite que el largo de cada uno de estos lugares pueda variar entre un largo mínimo un largo máximo, estas cotas en el largo pueden ser diferentes por cada lugar del piso pero estas no se pueden variar, esto para no alterar la forma de los lugares. Una de las primeras inquietudes y que abordaremos en este trabajo se puede formular así: *sí encerramos el piso en un rectángulo de tal manera que dicho rectángulo debe ser el menor rectángulo en el que se puede encerrar el piso, ¿cuál es el área mínima que puede tener ese rectángulo?*. Saber el área de este rectángulo permite a los constructores aprovechar al máximo el terreno de la construcción sin alterar los diseños del arquitecto.
Seguido, lo natural ya no es preguntarse por el área sino por el volumen, en ese sentido, suponiendo que se han impuesto restricciones en las alturas de cada lugar del piso y que los rectángulos que formas la base de los lugares son similares al caso anterior, la inquietud que abordaremos es: *sí encerramos la construcción en un cubo de tal manera que dicho cubo debe ser el menor cubo en el que se puede encerrar todos los bloques que forman la construcción, ¿cuál es el volumen mínimo que puede tener ese cubo?*.

El ámbito de la construcción civil es tan solo uno de los muchos campos en los que se puede presentar esta situación, precisamente, es fundamental formular esta situación sin importar el contexto y dar una solución, es en este punto en donde la formulación matemática de la situación es importante y que es precisamente lo que aborda este trabajo, para tal fin, se mostrara que dicho problema tiene una formulación de *gemetric program* (GP) que en principio no es un problema convexo, pero gracias a que es un Geometric program entonces se puede formular como un problema de optimización convexa de acuerdo a lo expuesto en [citar S. Boyd], en ese sentido, abordaremos en principio el caso del Floor planning en 2D, es decir, la primera pregunta que nos acabamos de formular, entender esta situación permite entender el Floor planning en 3D de manera concisa.

* Universidad de los Andes,Bogotá, Colombia.

2 Formulación matemática en 2D

El contexto geométrico en el que se desarrollan los conceptos de la primera parte de articulo se limita al plano, la primera parte termina cuando empieza el análisis de la situación en 3D, lo cual ocurre en la Sección 5, de modo que cuando hagamos referencia a un bloque en esta primera parte estamos haciendo alusión a un rectángulo, en ese sentido, consideremos n bloques de áreas A_1, $A_2, \ldots, A_n$, denotamos por (x_i, y_i) las coordenadas de la esquina inferior izquierda del bloque i, adicionalmente, w_i representa el largo del bloque i, y L_i y U_i son el largo mínimo y máximo del bloque i respectivamente, es decir, $L_i \leq w_i \leq U_i$ para cada $i = 1, \ldots, n$. Note que la altura de cada bloque es determinada por $h_i = A_i/w_i$.

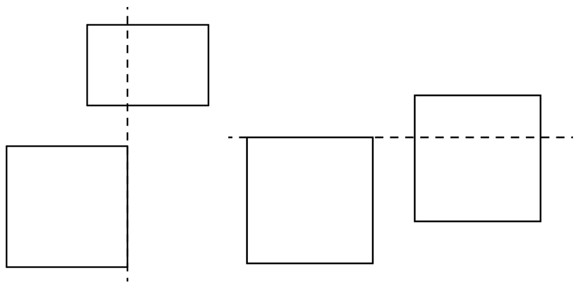
2.1 Disposición de los bloques en términos de grafos

En aras de formalizar la situación es fundamental caracterizar la disposición espacial de los bloques en términos accesibles en el sentido de la formulación que se pretende dar en este articulo, en ese sentido, los grafos son la herramienta adecuada.
Consideramos los grafos $\mathcal{H}$ y $\mathcal{V}$ definidos de la siguiente forma:

$\mathcal{H}$: Los nodos son los bloques i; si el bloque j esta a la derecha del bloque i sin solaparce entonces se establece una flecha de i a j.
Decimos que $(i, j) \in \mathcal{H}$ si existe una flecha de i a j en dicho grafo.

$\mathcal{V}$: Los nodos son los bloques i; si el bloque j esta arriba del bloque i sin solaparce entonces se establece una flecha de i a j.
Decimos que $(i, j) \in \mathcal{H}$ si existe una flecha de i a j en dicho grafo.

Note que las relaciones que determinan las flechas en ambos grafos son transitivas, esto permite omitir algunas flechas. Adicionalmete a cada uno de estos grafos insertamos el nodo $n + 1$, este solo tiene flechas entrantes que provienen de los nodos que son sumidero, es decir, nodos que no tienen flechas salientes. Gráficamente el nodo $n + 1$ no representa ningún bloque, en realidad, lo podemos representar con un punto (x_{n+1}, y_{n+1}) que se ubica en el extremos superior derecho del menor rectángulo que contiene a todos los bloques, precisamente, es el área de dicho rectángulo que se pretende minimizar.



(a) (b)

Fig. 2.1. (a) Bloques que solapan horizontalmente. (b) Bloques que se solapan verticalmente.

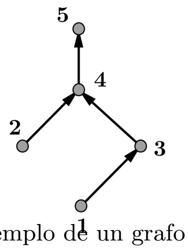
Además, denotamos por $\mathcal{F}_{\mathcal{H}}$ y $\mathcal{S}_{\mathcal{H}}$ el conjunto de nodos fuente y sumidero respectivamente del grafo $\mathcal{H}$, de manera análoga, $\mathcal{F}_{\mathcal{V}}$ y $\mathcal{S}_{\mathcal{V}}$ el conjunto de nodos fuente y sumidero respectivamente del grafo $\mathcal{V}$. Con todo lo anterior en mente, $\mathcal{H}$ y $\mathcal{V}$ son los grafos que representan la disposición horizontal y vertical de los bloques respectivamente, de modo que los llamamos grafo horizontal y vertical respectivamente.

2.2 Representación matricial de los grafos

Aunque no entraremos en los detalles de la programación del método que emplearemos para resolver el problema (1), para efectos de programación es importante codificar los grafos mencionados por medio de una matriz, en ese sentido, definimos las matrices H y V de de tamaño $n+1 \times n+1$ dadas por

$$h_{ij} = \begin{cases} 1 & \text{sí } (i,j) \in \mathcal{H} \\ 0 & \text{en otro caso} \end{cases}, \quad v_{ij} = \begin{cases} 1 & \text{sí } (i,j) \in \mathcal{V} \\ 0 & \text{en otro caso} \end{cases}.$$

A estas matrices las llamamos matriz horizontal y vertical respectivamente.



$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 2.2. Ejemplo de un grafo y su matriz asociada.

2.3 Formulación como problema de optimización

Dada la disposición de los n bloques, la cuál es determinada por los grafos $\mathcal{H}$ y $\mathcal{V}$, se da de antemano el área y el largo máximo y mínimo de cada bloque, entonces el problema consiste en minimizar el área del menor rectángulo que contiene los n bloques sujeto a dichas restricciones. Sin perdida de generalidad podemos asumir que si $i \in \mathcal{F}_H$, entonces $x_i = 0$, adicionalemente, si $i \in \mathcal{F}_V$, entonces $y_i = 0$, esta convención significa recostar los n bloques en el eje x e y. En ese sentido, teniendo en cuenta las convenciones anteriores el problema se puede formular de la siguiente manera:

$$\begin{array}{lll} P_{\text{Área}}: \text{Minimizar } x_{n+1}y_{n+1} & & \\ \quad\text{Sujeto a} & w_i \leq x_j & \forall (i,j) \in \mathcal{H} \text{ con } i \in \mathcal{F}_{\mathcal{H}} \\ & \frac{A_i}{w_i} \leq y_i & \forall (i,j) \in \mathcal{V} \text{ con } i \in \mathcal{F}_{\mathcal{V}} \\ & x_i + w_i \leq x_j & \forall (i,j) \in \mathcal{H} \text{ con } i \notin \mathcal{F}_{\mathcal{H}} \\ & y_i + \frac{A_i}{w_i} \leq y_j & \forall (i,j) \in \mathcal{V} \text{ con } i \notin \mathcal{F}_{\mathcal{V}} \\ & L_i \leq w_i & \forall i = 1, 2, \ldots, n. \\ & w_i \leq U_i & \forall i = 1, 2, \ldots, n. \end{array} \tag{1}$$

Las variables de este problema son todas las x_i tales que $i \notin \mathcal{F}_{\mathcal{H}}$, todas la y_i tales que $i \notin \mathcal{F}_{\mathcal{V}}$ y $w = w_1, w_2, \ldots, w_n$.

Este problema no es convexo dado que la función objetivo ni las funciones restricción son convexas, no obstante, este problema se enmarca dentro de una clase de problemas conocidos como *Geometric program* para los cuales se les puede dar un trato de problema de optimizan convexa mediante una adecuada transformación en las variables, en la siguiente sección haremos una revisión de este tipo de problemas.

3 Revisión de *Geometric programming* y su convexificación

Una función $f : \mathbb{R}^n \to \mathbb{R}$ con dominio $\text{dom} f = \mathbb{R}^n_{++}$ donde $\mathbb{R}^n_{++}$ es el conjunto de vectores en $\mathbb{R}^n$ cuyas componentes son todas estrictamente positivas, definida por

$$f(x) = c x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$$

donde $c > 0$ y $\alpha_i \in \mathbb{R}$, es llamada un *monomio*, note que en este contexto α_i puede ser cualquier numero racional, irracional e incluso negativo, esa es la diferencia con el termino "monomio" usado en álgebra.
Una suma de monomios, es decir, una función de la forma

$$f(x) = \sum_{k=1}^{K} c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \cdots x_n^{\alpha_{nk}}$$

donde $c_k > 0$ y $a_{ik} \in \mathbb{R}$, es llamada un *posinomio*, en este caso, de K términos.

Con esto ultimo en mente, un problema de optimización de la forma

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeto a} & g_i(x) \leq 1 \text{ para } i = 1, \ldots, m \\ & h_i(x) = 1 \text{ para } i = 1, \ldots, p \end{array} \tag{2}$$

donde $f, g_1, \ldots, g_m$ son posinomios y $h_1, h_2, \ldots, h_p$ son monomios, es llamado un *geometric program* (GP), el dominio de este problema es $\mathbb{R}^n_{++}$.

En general, un geometric program no es un problema de optimización convexa, esto se debe a que los posinomios en general no son funciones convexas, no obstante, se puede convexificar el problema mediante un cambio de variables y una transformación de las restricciones del problema, para ilustrar este procedimiento suponga que

$$f(x) = \textstyle\sum_{k=1}^{K} c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \cdots x_n^{\alpha_{nk}}, \quad g_i(x) = \sum_{k=1}^{K_i} b_k^i x_1^{\beta_{1k}^i} x_2^{\beta_{2k}^i} \cdots x_n^{\beta_{nk}^i} \text{ y}$$
$$h_i(x) = a^i x_1^{\delta_1^i} x_2^{\delta_2^i} \cdots x_n^{\delta_n^i}.$$

El cambio de variable que se usa es $X_i = \log x_i$, en ese sentido, $x_i = e^{X_i}$, así, considerando $X = (X_1, \ldots, X_n)$, $\alpha_k = (\alpha_{1k}, \ldots, \alpha_{nk})$, $\beta_k^i = \left(\beta_{1k}^i, \ldots, \beta_{nk}^i\right)$, $\delta^i = \left(\delta_1^i, \ldots, \delta_n^i\right)$, $C_k = \log c_k$, $B_k^i = \log b_k^i$ y $A^i = \log a^i$, el problema (2) puede ser expresado en términos de las nuevas variables como sigue:

$$\begin{array}{ll} \text{minimizar} & \sum_{k=1}^{K} e^{\alpha_k^T X + C_k} \\ \text{sujeto a} & \sum_{k=1}^{K_i} e^{\beta_k^{iT} X + B_k^i} \leq 1 \text{ para } i = 1, \ldots, m \\ & e^{\delta^{iT} X + A^i} = 1 \qquad \text{para } i = 1, \ldots, p \end{array}$$

donde T significa transposición. Ahora, transformamos la función objetivo y las restricciones tomando logaritmos, así, resulta el siguiente problema

$$\begin{array}{ll} \text{minimizar} & \tilde{f}(x) = \log\left(\sum_{k=1}^{K} e^{\alpha_k^T X + C_k}\right) \\ \text{sujeto a} & \tilde{g}_i(x) = \log\left(\sum_{k=1}^{K_i} e^{\beta_k^{iT} X + B_k^i}\right) \leq 0 \text{ para } i = 1, \ldots, m \\ & \tilde{h}_i(x) = \delta^{iT} X + A^i = 0 \qquad \text{para } i = 1, \ldots, p. \end{array} \tag{3}$$

Ya que las funciones $\tilde{f}$, $\tilde{g}_1$,$\tilde{g}_2$,..., $\tilde{g}_m$ son convexas y las funciones $\tilde{h}_i$ son afines, entonces, este ultimo es un problema de optimización convexa.

4 Floor planning 2D como un geometric program

El problema (1) es equivalente al siguiente problema:

$$
\begin{array}{llll}
P_{\text{Área}}: & \text{Minimizar } x_{n+1}y_{n+1} & & \\
 & \text{Sujeto a} & \frac{w_i}{x_j} \leq 1 & \forall (i,j) \in \mathcal{H} \text{ con } i \in \mathcal{F}_{\mathcal{H}} \\
 & & \frac{A_i}{w_i y_i} \leq 1 & \forall (i,j) \in \mathcal{V} \text{ con } i \in \mathcal{F}_{\mathcal{V}} \\
 & & \frac{x_i}{x_j} + \frac{w_i}{x_j} \leq 1 & \forall (i,j) \in \mathcal{H} \text{ con } i \notin \mathcal{F}_{\mathcal{H}} \\
 & & \frac{y_i}{y_j} + \frac{A_i}{w_i y_j} \leq 1 & \forall (i,j) \in \mathcal{V} \text{ con } i \notin \mathcal{F}_{\mathcal{V}} \\
 & & \frac{L_i}{w_i} \leq 1 & \forall i = 1,2,\ldots,n. \\
 & & \frac{w_i}{U_i} \leq 1 & \forall i = 1,2,\ldots,n.
\end{array}
\tag{4}
$$

El cual es un geometric program donde, de nuevo, las variables de este problema son todas las x_i tales que $i \notin \mathcal{F}_{\mathcal{H}}$, todas la y_i tales que $i \notin \mathcal{F}_{\mathcal{V}}$ y $w = w_1, w_2, \ldots, w_n$.

Así pues, de acuerdo a la sección anterior, la convexificación de este problema da lugar a la siguiente formulación

$$
\begin{array}{llll}
GP_{\text{Área}}: & \text{Minimizar } X_{n+1} + Y_{n+1} & & \\
 & \text{Sujeto a} & W_i - X_j \leq 0 & \forall (i,j) \in \mathcal{H} \text{ con } i \in \mathcal{F}_{\mathcal{H}} \\
 & & W_i - Y_j + \log A_i \leq 0 & \forall (i,j) \in \mathcal{V} \text{ con } i \in \mathcal{F}_{\mathcal{V}} \\
 & & \log\left(e^{X_i - X_j} + e^{W_i - X_j}\right) \leq 0 & \forall (i,j) \in \mathcal{H} \text{ con } i \notin \mathcal{F}_{\mathcal{H}} \\
 & & \log\left(e^{Y_i - Y_j} + A_i e^{-W_i - X_j}\right) \leq 0 & \forall (i,j) \in \mathcal{V} \text{ con } i \notin \mathcal{F}_{\mathcal{V}} \\
 & & -W_i + \log L_i \leq 0 & \forall i = 1,2,\ldots,n. \\
 & & W_i - \log U_i \leq 0 & \forall i = 1,2,\ldots,n.
\end{array}
\tag{5}
$$

Las variables de este problema son todas las $X_i = \log x_i$ tales que $i \notin \mathcal{F}_{\mathcal{H}}$, todas las $Y_i = \log y_i$ tales que $i \notin \mathcal{F}_{\mathcal{V}}$ y $W_i = \log w_i$ para cada $i = 1, \ldots, n$. Por lo tanto, resolver el problema (4) es equivalente a solucionar el problema (5) ya que de los óptimos de este ultimo se obtiene los óptimos de (4) vía la exponencial de dichos valores.

4.1 Ejemplo de Floor planning 2D por medio de CVX

Lo ideal es contar con una herramienta que no se restringa a casos particulares, precisamente CVX, que es un programa que esta ambientado en `Matlab`, permite solucionar este tipo de problemas de optimización convexa, el inconveniente se encuentra en que cada situación de Floor planning representa una serie de restricciones que varían de acuerdo a la configuración de los bloques, de modo que si el problema cuenta con un numero considerable de bloques, entonces su ingreso a CVX es engorroso ya que el numero de restricciones sera considerablemente grande, para tal fin es pertinente idear una forma para solventar esta situación, precisamente, caracterizar los grafo $\mathcal{H}$ y $\mathcal{V}$ como matriz las matrices H y V respectivamente (ver Subsección 2.2) constituyen un avance significativo en ese sentido. Por lo tanto, se ha diseñado una rutina en `Matlab` usando CVX a modo de programa llamada `Floorplanning.m`, para entender su funcionamiento veamos el siguiente ejemplo:

Consideremos la configuración de bloques dada en la Figura 4.1 y supongamos que cada bloque esta sujeto a las restricciones dadas en la Tabla 1, entonces, la Figura 4.2 se presentan los respectivos grafos y las matrices asociadas a ellos.

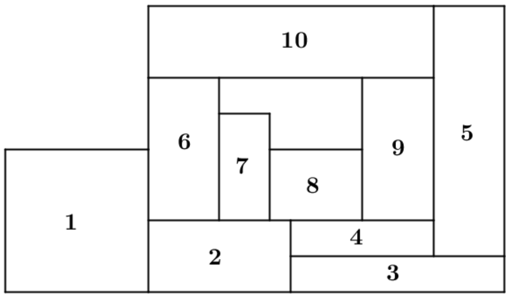
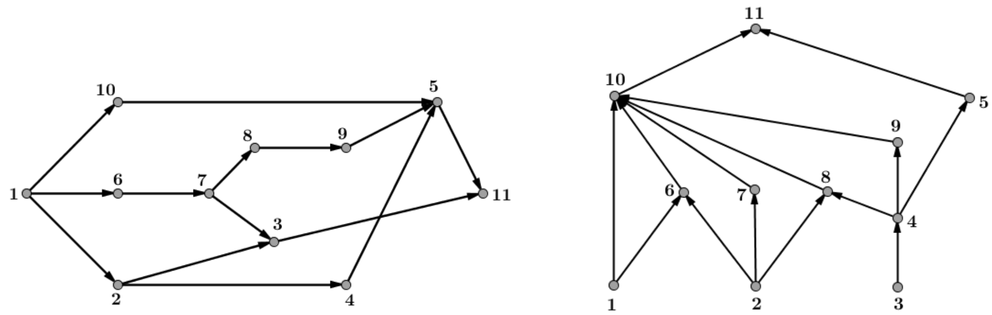


Fig. 4.1. Ejemplo de propuesta inicial de configuración de bloques.

Table 1. Tabla de restricciones

Bloque	Área (A_i)	Largo mínimo (L_i)	Largo máximo (U_i)
1	400	15	20
2	200	15	20
3	150	25	30
4	100	15	20
5	350	5	10
6	200	5	10
7	75	3	8
8	150	13	18
9	200	5	10
10	400	30	45



(a) (b)

$$H = \begin{bmatrix} 0&1&0&0&0&1&0&0&0&1&0 \\ 0&0&1&1&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&1&1&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

(c)

$$V = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&1&1&1&1&0&0 \\ 0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

(d)

Fig. 4.2. Grafos y matrices asociados a la disposición de bloques de la Figura 4.1. (a) Grafo horizontal. (b) Grafo vertical. (c) Matriz del grafo horizontal. (d) Matriz del grafo vertical.

`Floorplanning` es una función que toma las matrices H, V y los vectores A, U y L (los últimos dos deben estar dispuestos verticalmente en `matlab`) y devuelve dos objetos, el primero es una matriz con el numero de filas igual al numero de bloques del problema mas el nodo adicional, es decir, si el problema consiste de n bloques, entonces el largo de dicha matriz es $n+1$, cada fila representa uno de los bloque, respecto a las columnas, esta matriz cuenta con tres, la primera, segunda y tercera columna representan las coordenadas x, y y el largo w respectivamente de cada bloque, los valores que están en dicha matriz son los óptimos que permiten minimizar el área del menor rectangular que contiene todos los bloques. El segundo objeto que devuelve `Floorplanning` es precisamente el valor de esa área mínima, de modo que un ingreso adecuado es el siguiente:

```
>> [Optimos,AreaMin]=Floorplanning(H,V,A,L,U)
```

Para nuestro ejemplo, en `Floorplanning` obtenemos que la menor área posible del rectángulo que encierra los diez recuadros es `AreaMin` $= 25.608$ y los valores óptimos, es decir, `Optimos` son los consignados en la Tabla 2, adicionalmente, `Floorplanning` proporciona una imagen con una de las configuraciones optimas (ver Figura 4.3).

Table 2. Tabla de resultados óptimos

Bloque	coordenada x_i	coordenada y_i	Largo optimo w_i
1	0	0	15.0000
2	15.0000	0	18.4544
3	33.4544	0	27.9241
4	33.4544	5.3717	18.2956
5	51.7500	5.3717	9.6286
6	15.0000	10.8375	10.0000
7	25.0000	10.8375	3.7500
8	28.7500	14.7006	13.0000
9	41.7500	10.8375	10.0000
10	15.0000	30.8375	36.7500

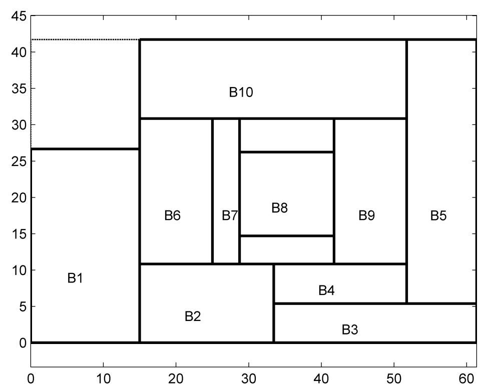


Fig. 4.3. Disposición optima de los bloques.

5 Floor planning en 3D

Como es natural cada problema suele tener versiones en otros contextos, en algunos casos son abstracciones de una situación que a la postre pasa a ser un caso particular, en este caso, lo que sigue es pensar en la situación en la cual los bloques ya no son rectángulos sobre el plano xy sino cubos en el espacio xyz, esto es, dados n bloques con áreas de sus bases fijas pero con restricciones en su ancho y su alto, y con su disposición en el espacio fijada, el interés es saber el menor volumen que puede tener el menor cubo que contiene a todos los bloques, y más aun, saber cual es la forma de los bloques que permite alcanzar dicho mínimo.

La formulación de este problema es la siguiente, consideremos n bloques cuyas bases tienen áreas A_1, $A_2, \ldots, A_n$, denotamos por (x_i, y_i, z_i) las coordenadas del vértice inferior izquierdo del bloque i como aparece en la Figura 5.1, adicionalmente, w_i representa el ancho del bloque i, y L_i^H y U_i^H son el ancho mínimo y máximo del bloque i respectivamente, es decir, $L_i^H \leq w_i \leq U_i^H$ para cada $i = 1, \ldots, n$, por otro lado, h_i representa la altura del bloque i, y L_i^T y U_i^T son la altura mínima y máxima del bloque i respectivamente, es decir, $L_i^T \leq h_i \leq U_i^T$ para cada $i = 1, \ldots, n$. Note que la profundidad de cada bloque es determinada por $p_i = A_i/w_i$.

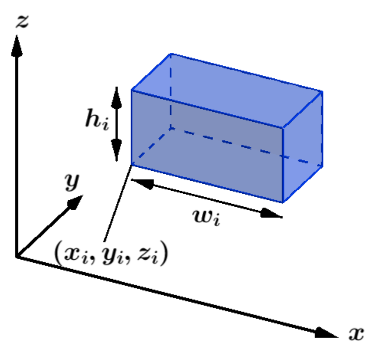


Fig. 5.1. Identificación de los bloques.

5.1 Disposición de los bloques en términos de grafos

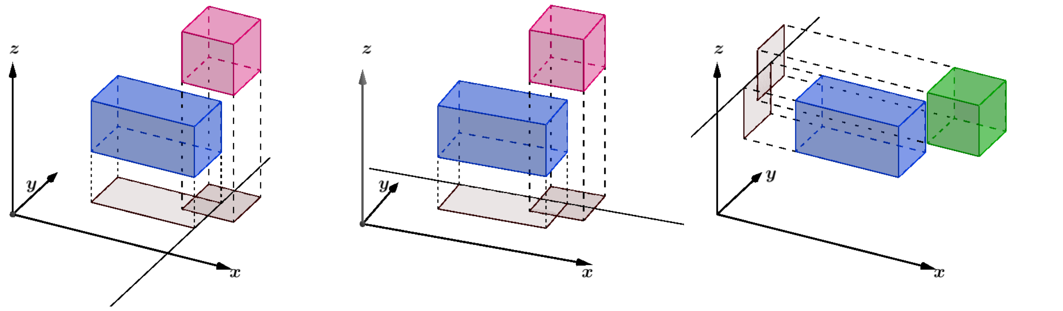
De manera similar al caso 2D, y teniendo en cuenta el sentido de los ejes xyz dado en la Figura 5, en este caso consideramos tres grafos que son grafos $\mathcal{H}$, $\mathcal{V}$ y $\mathcal{T}$ definidos de la siguiente forma:

$\mathcal{H}$: Los nodos son los bloques i; si el bloque j esta a la derecha del bloque i sin solaparce entonces se establece una flecha de i a j.
Decimos que $(i, j) \in \mathcal{H}$ si existe una flecha de i a j en dicho grafo.

$\mathcal{V}$: Los nodos son los bloques i; si el bloque j esta atras del bloque i sin solaparce entonces se establece una flecha de i a j.
Decimos que $(i, j) \in \mathcal{H}$ si existe una flecha de i a j en dicho grafo.

$\mathcal{T}$: Los nodos son los bloques i; si el bloque j esta arriba del bloque i sin solaparce entonces se establece una flecha de i a j.
Decimos que $(i, j) \in \mathcal{H}$ si existe una flecha de i a j en dicho grafo.

Sabemos las relaciones que determinan las flechas en estos grafos son transitivas, esto permite omitir algunas flechas. Adicionalmete a cada uno de estos grafos insertamos el nodo $n + 1$, este solo tiene flechas entrantes que provienen de los nodos que son sumidero, es decir, nodos que no tienen flechas salientes. De nuevo, gráficamente el nodo $n+1$ no representa ningún bloque, en realidad, lo podemos representar con un punto $(x_{n+1}, y_{n+1}, z_{n+1})$ que se ubica en el extremos superior derecho posterior del menor cubo que contiene a todos los bloques, precisamente, es el volumen de dicho cubo que se pretende minimizar.



(a) (b) (c)

Fig. 5.2. El significado de solaparse para cada grafo. (a) Para $\mathcal{H}$. (a) Para $\mathcal{V}$. (a) Para $\mathcal{T}$.

Además, denotamos por $\mathcal{F}_\mathcal{H}$ y $\mathcal{S}_\mathcal{H}$ el conjunto de nodos fuente y sumidero respectivamente del grafo $\mathcal{H}$, $\mathcal{F}_\mathcal{V}$ y $\mathcal{S}_\mathcal{V}$ el conjunto de nodos fuente y sumidero respectivamente del grafo $\mathcal{V}$, y de manera análoga, $\mathcal{F}_\mathcal{T}$ y $\mathcal{S}_\mathcal{T}$ el conjunto de nodos fuente y sumidero respectivamente del grafo $\mathcal{T}$. Con todo lo anterior en mente, $\mathcal{H}$, $\mathcal{V}$ y $\mathcal{T}$ son los grafos que representan la disposición a lo largo, a lo profundo y a lo alto de los bloques respectivamente.

5.2 Formulación como problema de optimización

Dada la disposición de los n bloques, la cuál es determinada por los grafos $\mathcal{H}$, $\mathcal{V}$ y $\mathcal{T}$, se da de antemano el área de la base, el ancho máximo y mínimo, y la altura máxima y mínima de cada bloque, entonces el problema consiste en minimizar el volumen del menor cubo que contiene los n bloques sujeto a dichas restricciones. De manera similar al caso 2D, sin perdida de generalidad podemos asumir que si $i \in \mathcal{F}_H$, entonces $x_i = 0$, si $i \in \mathcal{F}_V$, entonces $y_i = 0$, adicionalemente, si $i \in \mathcal{F}_T$, entonces $z_i = 0$, esta convención significa recostar los n bloques en los eje x, y y z. En ese sentido, teniendo en cuenta las convenciones anteriores el problema se puede formular de la siguiente manera:

$$
P_{\text{Área}} : \begin{array}{lll}
\text{Minimizar} & x_{n+1}y_{n+1}z_{n+1} & \\
\text{Sujeto a} & \frac{h_i}{z_j} \leq 1 & \forall (i,j) \in \mathcal{T} \text{ con } i \in \mathcal{F}_\mathcal{T} \\
& \frac{z_i}{z_j} + \frac{h_i}{z_j} \leq 1 & \forall (i,j) \in \mathcal{T} \text{ con } i \notin \mathcal{F}_\mathcal{T} \\
& \frac{L_i^T}{h_i} \leq 1 & \forall i = 1, 2, \ldots, n. \\
& \frac{h_i}{U_i^T} \leq 1 & \forall i = 1, 2, \ldots, n. \\
& \frac{w_i}{x_j} \leq 1 & \forall (i,j) \in \mathcal{H} \text{ con } i \in \mathcal{F}_\mathcal{H} \\
& \frac{A_i}{w_i y_i} \leq 1 & \forall (i,j) \in \mathcal{V} \text{ con } i \in \mathcal{F}_\mathcal{V} \\
& \frac{x_i}{x_j} + \frac{w_i}{x_j} \leq 1 & \forall (i,j) \in \mathcal{H} \text{ con } i \notin \mathcal{F}_\mathcal{H} \\
& \frac{y_i}{y_j} + \frac{A_i}{w_i y_j} \leq 1 & \forall (i,j) \in \mathcal{V} \text{ con } i \notin \mathcal{F}_\mathcal{V} \\
& \frac{L_i^H}{w_i} \leq 1 & \forall i = 1, 2, \ldots, n. \\
& \frac{w_i}{U_i^H} \leq 1 & \forall i = 1, 2, \ldots, n.
\end{array} \tag{6}
$$

Las variables de este problema son todas las x_i tales que $i \notin \mathcal{F}_{\mathcal{H}}$, todas la y_i tales que $i \notin \mathcal{F}_{\mathcal{V}}$, todas la z_i tales que $i \notin \mathcal{F}_{\mathcal{T}}$, $w = w_1, w_2, \ldots, w_n$ y $h = h_1, h_2, \ldots, h_n$.

Este problema es similar al problema (4) (el caso 2D) con la salvedad que se han insertado tres nuevas restricciones, al igual que en aquel caso, este problema no es convexo dado que la función objetivo ni las funciones restricción son convexas,sin embargo, este un *Geometric program*, de modo que se puede convexidad.

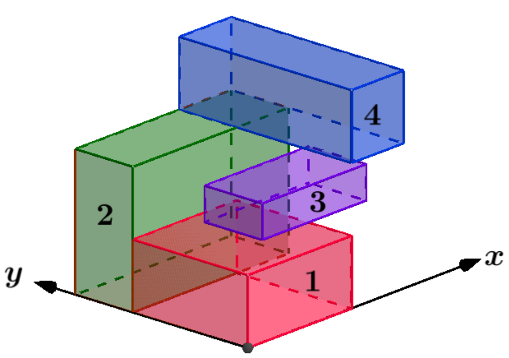
Así pues, de acuerdo a la Sección 3, la convexificación de este problema da lugar a la siguiente formulación

$$
\begin{array}{rlll}
GP_{\text{Área}}: & \text{Minimizar} & X_{n+1} + Y_{n+1} & \\
& \text{Sujeto a} & \overline{H}_i - Z_j \leq 0 & \forall (i,j) \in \mathcal{T} \text{ con } i \in \mathcal{F}_{\mathcal{T}} \\
& & \log\left(e^{Z_i - Z_j} + e^{\overline{H}_i - Z_j}\right) \leq 0 & \forall (i,j) \in \mathcal{T} \text{ con } i \notin \mathcal{F}_{\mathcal{T}} \\
& & -\overline{H}_i + \log L_i^T \leq 0 & \forall i = 1, 2, \ldots, n. \\
& & \overline{H}_i - \log U_i^T \leq 0 & \forall i = 1, 2, \ldots, n. \\
& & W_i - X_j \leq 0 & \forall (i,j) \in \mathcal{H} \text{ con } i \in \mathcal{F}_{\mathcal{H}} \\
& & W_i - Y_j + \log A_i \leq 0 & \forall (i,j) \in \mathcal{V} \text{ con } i \in \mathcal{F}_{\mathcal{V}} \\
& & \log\left(e^{X_i - X_j} + e^{W_i - X_j}\right) \leq 0 & \forall (i,j) \in \mathcal{H} \text{ con } i \notin \mathcal{F}_{\mathcal{H}} \\
& & \log\left(e^{Y_i - Y_j} + A_i e^{-W_i - X_j}\right) \leq 0 & \forall (i,j) \in \mathcal{V} \text{ con } i \notin \mathcal{F}_{\mathcal{V}} \\
& & -W_i + \log L_i^H \leq 0 & \forall i = 1, 2, \ldots, n. \\
& & W_i - \log U_i^H \leq 0 & \forall i = 1, 2, \ldots, n.
\end{array}
\tag{7}
$$

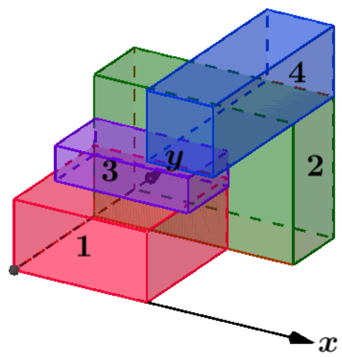
Las variables de este problema son todas las $X_i = \log x_i$ tales que $i \notin \mathcal{F}_{\mathcal{H}}$, todas las $Y_i = \log y_i$ tales que $i \notin \mathcal{F}_{\mathcal{V}}$, todas las $Z_i = \log z_i$ tales que $i \notin \mathcal{F}_{\mathcal{T}}$, $W_i = \log w_i$ para cada $i = 1, \ldots, n$ y $\overline{H}_i = \log h_i$ para cada $i = 1, \ldots, n$. Por lo tanto, resolver el problema (7) es equivalente a solucionar el problema (6) ya que de los óptimos de este ultimo se obtiene los óptimos de (7) vía la exponencial de dichos valores.

5.3 Ejemplo por medio de CVX en el caso 3D

De manera similar al caso 2D, se ha diseñado una rutina en `Matlab` usando CVX a modo de programa llamada `Floorplanning3D.m`, para entender su funcionamiento veamos el siguiente ejemplo:



(a)



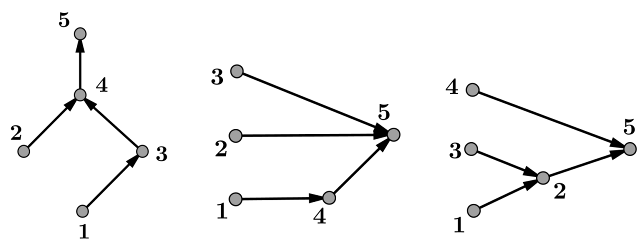
(b)

Fig. 5.3. Propuesta inicial de disposición de bloques. (a) y (b) son vistas desde ángulos diferentes.

Consideremos la configuración de bloques dada en la Figura 5.3 y Supongamos que cada bloque esta sujeto a las restricciones dadas en la Tabla 3, entonces, la Figura 5.4 se presentan los respectivos grafos y las matrices asociadas a ellos.

Table 3. Tabla de restricciones para el caso 3D

Bloque	Área (A_i)	Ancho mínimo (L_i^H)	Ancho máximo (U_i^H)	Alto mínimo (L_i^T)	Alto máximo (U_i^T)
1	4	1.5	2	0.5	1
2	3	2.5	3	1.5	2
3	2	1.5	2	0.5	1
4	3	0.5	1	0.5	1



(a) (b) (c)

$$T = \begin{bmatrix} 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix} \quad H = \begin{bmatrix} 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&1 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix} \quad V = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&0&0&1 \\ 0&1&0&0&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$$

(d) (e) (f)

Fig. 5.4. Grafos y matrices asociados a la disposición de bloques de la Figura 4.1. (a) Grafo de altura. (b) Grafo de ancho. (c) Grafo de profundidad. (d) Matriz del grafo de altura. (e) Matriz del grafo de ancho. (f) Matriz del grafo de profundidad.

`Floorplanning3D` es una función que toma las matrices H, V y T, y los vectores A, U^H, L^H, U^T, L^T (los últimos cuatro deben estar dispuestos verticalmente en `matlab`) y devuelve dos objetos, el primero es una matriz con el numero de filas igual al numero de bloques del problema mas el nodo adicional, es decir, si el problema consiste de n bloques, entonces el largo de dicha matriz es $n+1$, cada fila representa uno de los bloque, respecto a las columnas, esta matriz cuenta con cinco, la primera, segunda, tercera, cuarta y quinta columna representan las coordenadas x, y, z, el largo w y la altura h respectivamente de cada bloque, los valores que están en dicha matriz son los óptimos que permiten minimizar el volumen del menor cubo que contiene todos los bloques. El segundo objeto que devuelve `Floorplanning3D` es precisamente el valor de esa volumen mínimo, de modo que un ingreso adecuado es el siguiente:

```
>> [Optimos,VolMin]=Floorplanning3D(H,V,T,A,LH,UH,LT,UT)
```

Para nuestro ejemplo, en `Floorplanning3D` obtenemos que la menor volumen posible del menor cubo que encierra los diez recuadros es `VolMin` = 18 y los valores óptimos, es decir, `Optimos` son los consignados en la Tabla 4, adicionalmente, `Floorplanning3D` proporciona una imagen con una de las configuraciones optimas (ver Figura 5.5).

Table 4. Tabla de resultados óptimos en el caso 3D

Bloque	coordenada x_i	coordenada y_i	coordenada z_i	Ancho optimo w_i	Alto optimo h_i
1	0	0	0	1.7714	0.5309
2	0	2.2580	0	2.6572	1.5000
3	0	0	0.5785	1.6909	0.5317
4	1.7714	0	1.5000	0.8857	0.5000

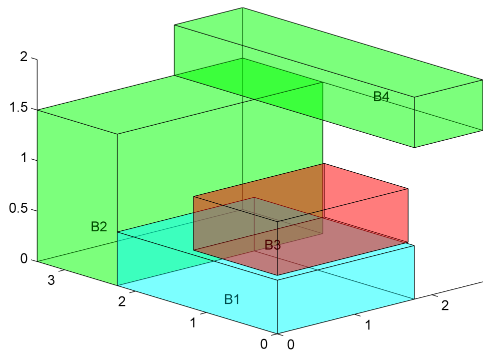


Fig. 5.5. Disposición optima de los bloques.

Referencias

1. S. Boyd & L. Vanderberghe, *Convex optimization*, Cambridge University Press, 160-174 (2004)
2. S. Sutanthavibul & E. Shragowitz & J. B. Rosen , *An Analytical Approach to Floorplan Design and Optimization*, IEEE transactions and computer-aided design, 10(6), 161-169 (1991)