# Minimización del uso de varillas para la producción de sillones de mimbre

May 2, 2022

## 1 Requerimientos

Para construir un sillón de mimbre, se precisan 4 piezas de tamaño 35, 4 piezas de tamaño 30 y 4 piezas de tamaño 20. Para eso, se proveen piezas de tamaño 130. El objetivo del productor es encontrar un tamaño de lote y la "receta" que le permita producir ese lote minimizando el desperdicio de mimbre.

## 2 Formalización del problema

Comenzamos con un poco de notación:

- Llamaremos $l_1, l_2, l_3$ a las tres longitudes de las varillas requeridas para la construcción de cada sillón, ordenados de manera decreciente.
- Sea $l$ la longitud total por varilla del proveedor.
- Cada receta deberá considerar particiones (formas de partir) la varilla de longitud $l$; formalmente $p = (l^1 \dots, l^k)$ donde $l^1 + \cdots + l^k = l$.

Para formalizar la postulación del problema, definimos la noción de *partición óptima* como una partición $(l^1, \dots, l^r)$ donde $l^1 \geq \cdots \geq l^{r-1} > l^r$ y $l^1, \dots, l^{r-1} \in \{l_1, l_2, l_3\}$. La propiedad del orden no es esencial y solamente simplifica el desarrollo posterior (y la presentación para el fabricante), lo importante es que todas las piezas corresponden a las medidas en $l_1, l_2, l_3$ y solo habrá una pieza remanente (el retazo) que no será posible utilizar.

Sea $P$ el conjunto de particiones, y $\hat{P}$ el conjunto de particiones óptimas. Será claro que podemos computar el conjunto $\hat{P}$ (lo veremos después) y luego podremos escribir $\hat{P} = (p^1, \ldots, p^k)$, donde $k$ es la cantidad total de particiones para los parámetros $(l, l_1, l_2, l_3)$.

Para cada partición $p$, consideraremos cuatro valores:

1. $N_1(p), N_2(p)$ y $N_3(p)$ que indican respectivamente las multiplicidades de $l_1$, $l_2$ y $l_3$ respectivamente en la partición $p$.

2. $C(p) = l^r$ si $p = (l^1, \ldots, l^r)$. Es decir, $C(p)$ es la longitud del retazo de $p$.

Dicho todo esto, enunciamos el problema de optimización:

Encontrar $(n_1, .., n_k) \in \mathbb{N}^k$ que minimize $\sum_{i=1}^{n} C(p_i) \times n_i$, sujeto a las restricciones

1. $n_1 + \cdots + n_k < n_{tope}$

2. $\sum_{i=1}^{k} N_1(p^i) \times n_i = \sum_{i=1}^{k} N_2(p^i) \times n_i = \sum_{i=1}^{k} N_3(p^i) \times n_i$

siendo $n_{tope}$ el máximo tolerable de varillas por lote.

# 3 Tratamiento del problema usando IP

Lo que veremos a continuación es que, si fijamos $n < n_{tope}$, podremos tratar el problema en cuestión como un problema clásico de programación entera.

Definimos la matriz $A \in \mathbb{N}^{2\times k}$ como:

$$\begin{pmatrix} N_1(p_1) - N_2(p_1) & \cdots & N_1(p_k) - N_2(p_k) \\ N_2(p_1) - N_2(p_1) & \cdots & N_3(p_k) - N_3(p_k) \\ 1 & \cdots & 1 \end{pmatrix} \tag{1}$$

y el vector $\vec{c} \in \mathbb{N}^k$

$$\begin{pmatrix} C(p_1) \\ \vdots \\ C(p_k) \end{pmatrix} \tag{2}$$

Podemos enunciar el siguiente problema de programación entera:

Encontrar $\vec{n} \in \mathbb{Z}^k$ que minimize $\vec{c} \cdot \vec{n}$, sujeto a las restricciones:

1. $A\vec{n} = (0, 0, n_0)^t$
2. $\vec{n} \geq \vec{0}$

Es claro que:

1. La restricción 2. hace que $\vec{n} \in \mathbb{N}^k$
2. Es claro que

$$\begin{array}{rcl} (A\vec{n})_0 = 0 & \iff & \sum_{i=1}^{k} (N_1(p_i) - N_2(p_i)) \times n_i = 0 \\ & \iff & \sum_{i=1}^{k} N_1(p_i) \times n_i = \sum_{i=1}^{k} N_2(p_i) \times n_i \end{array}$$

   , luego $((A\vec{n})_0, (A\vec{n})_1) = (0, 0) \iff$ se cumple la restricción 1. del problema original.

3. Es claro que $(A\vec{n})_2 = n_0$ implica $n_1 + \cdots + n_k = n_0 < n_{tope}$, que es la restricción 1. del problema original.

Luego, concluimos que para cada $n < n_{tope}$, el problema anterior está dentro de las restricciones del problema original y que el mínimo de las soluciones obtenidas variando $n < n_{tope}$ nos da la solución al problema original.

# 4 Implementación