Generación de columnas

Profesor: Jorge Vera (jvera@ing.puc.cl)
Ayudante: Moisés Saavedra (mmsaavedra1@ing.puc.cl)

A modo de complementar lo visto en clases a continuación se expone la logica detrás del algoritmo de generación de columnas.

Se posee un a materia prima (tronco de árbol) que posee un largo de M. Esta materia prima se puede cortar en trozos (items) m denotados como w_i. Existe una demanda por cada trozo que será definida como d_i. Ahora, se definen **patrones** de corte, cada uno de ellos representa la cantidad de trozos w_i obtenidos al cortar la materia prima de largo M, los que son almacenados en una matriz donde las columnas representan el patrón de corte y las filas el trozo al que corresponde.

Ahora un ejemplo de lo descrito anteriormente: Se posee una materia prima de largo M = 100, que puede ser cortado en m = 3 tipos de trozos, $w_i = 25, 35, 45$. Para cada trozo se tiene la siguiente demanda $d_i = 100, 200, 300$. Ahora los distintos posibles patrones de corte son:

- Patrón 1: 4 trozos de tamaño $w_1 = 24$
- Patrón 2: 1 trozo de tamaño $w_1 = 25$ y 2 trozos de tamaño $W_2 = 35$
- Patrón 3: ...

Se pueden crear todos los patrones que uno desee en base a combinatoria, pero esto computacionalmente es muy costoso, aquí viene el desafío de optimizar la cantidad de trozos cortados para ocupar la menor cantidad de materia prima posible y satisfacer la demanda.

Este es un problema de optimización entera y su resolución es **muy costosa computacionalmente** por lo que se inventó el método de generación de columnas que va alterando la matriz de patrones y en cada iteración saca una columna e ingresa aquella que otorga el mayor costo reducido negativo.

Para esto se divide en dos partes el problema: un problema maestro y un problema satélite. La gracia detrás es que una vez resuelto el problema maestro se ocupan los valores del duales como parámetros para el satélite, luego se resuelve el satélite y los valores obtenidos actualizan la matriz de patrones del maestro.

Para el problema **maestro** se tiene:

$$\begin{aligned}
\min \quad & \sum_{\mathrm{j=1}}^{\mathrm{k}} \mathrm{x_j} \\
\text{s.a} \quad & \sum_{j=1}^{k} a_{ij}x_j \geq d_i \quad i = 1, ..., m \\
& x_j \in Z_+ \qquad\qquad j = 1, ..., k
\end{aligned}$$

Aquí los parámetros son:

- a_{ij}: Número de veces que se corta el trozo i con el patrón j.
- d_i: Demanda del trozo i.

La variable involucrada es:

- x_j: Número de veces que el patrón j es usado.

Donde el problema **maestro dual** asociado es:

$$\begin{aligned}
\max \quad & \sum_{\mathrm{i=1}}^{\mathrm{m}} \mathrm{d_i}\pi_{\mathrm{i}} \\
\text{s.a} \quad & \sum_{i=1}^{m} a_{ij}\pi_i \leq 1 \quad j = 1, ..., k \\
& \pi_i \geq 0 \qquad\qquad i = 1, ..., m
\end{aligned}$$

De aquí los costos duales son los que se rescatan para ser traspasados al problema **satélite**, que es un problema de *textbf* knapsack definido como:

$$\begin{aligned}
\max \quad & \sum_{\mathrm{i=1}}^{\mathrm{m}} \pi_{\mathrm{i}}\mathrm{y_i} \\
\text{s.a} \quad & \sum_{i=1}^{m} w_i y_i \leq M \\
& y_i \in Z_+ \qquad\qquad i = 1, ..., m
\end{aligned}$$

Aquí los parámetros son:

- w_i: Largo del trozo i.
- M: Largo de la materia prima.
- π_i = Valores óptimos del dual del maestro, ya resuelto anteriormente.

La variable involucrada es:

- y_i: Representa la cantidad de trozos obtenidos para el item i, y si uno se fija mejor es un patrón de corte.

Ahora el algoritmo es el siguiente:

a. **Comienzo:** Con un caso base simple de patrones de corte (matriz A), un simple patrón de corte es para cada trozo calcularlo como $\frac{W}{w_i}$ y considerar el residuo como pérdida y no cortar más trozos de diferentes piezas, de esta forma se obtendrá una matriz con valores en su diagonal y el resto de valores 0.

b. **Repetición**

1. Resolver el **problema maestro relajado**, es decir, que no se considera $x_j \in Z_+$, si no que $x_j \geq 0$. Aquí queda un problema que puede ser resuelto con gurobi usando SIMPLEX, que gurobi detecta solo. La gracia de gurobi es que da la ópcion de almacenar las soluciones del problema dual π_i.
2. Con la información anterior, resolver el problema en gurobi y este internamente sabe que debe resolverlo como un problema **KNAPSCAK**.
3. Se agrega la nueva columna al **problema maestro relajado** mediante el objeto **Column** de gurobi.
4. Finaliza el loop: Según el error que se le desee dar o que el valor óptimo del **problema satélite** sea ≥ 0.

c. **Termina:** Cuando se tiene la matriz de patrones óptima que resuelve el problema satélite se resuelve el **problema maestro** que considera $x_j \in Z_+$ mediante un algotimo de Branch Bound, que nuevamente gurobi reconoce solo.

Todo lo anterior queda descrito en el script ***generación_columnas.py***, donde se puede modificar los valores de tamaño de materia prima (W), los trozos de tamaño (w_i) y la demanda (d_i) solo modificando las variables: cantidad_de_piezas, tamano_materia_prima, valor_minimo_demanda, valor_maximo_demanda.