Linear Programming

Chongnan Li
chongnanli1997@hotmail.com

Additional content 2: Cutting stock problem

- Problem formulation

Suppose the **fixed width** of large rolls is W. There are m customers and customer i wants n_i items of width w_i $(i = 1,\cdots,m)$ $(w_i \leq W)$.

Notations:
K: index set of available rolls, in other words, $k = 1,\cdots,K$

$$y_k = \begin{cases} 1, & \text{if roll } k \text{ is cut} \\ 0, & \text{otherwise} \end{cases}$$

x_i^k: number of times item i is cut on roll k.

The integer programming formulation by *Kantorovich*:

$$(\mathrm{P}_1) \quad \begin{cases} \text{minimize} & \displaystyle\sum_{k=1}^{K} y_k \quad \text{(minimize the total number of rolls that are utilized)} \\ \text{subject to} & \displaystyle\sum_{k=1}^{K} x_i^k \geq n_i \quad i = 1,\cdots,m \quad \text{(demand should be satisfied)} \\ & \displaystyle\sum_{i=1}^{m} w_i x_i^k \leq W y_k \quad k = 1,\cdots,K \quad \text{(width limitaion)} \\ & \boldsymbol{x}_i^k \in \mathbb{Z}^+ \cup \{0\},\ y_k \in \{0,1\}, \quad \forall i = 1,\cdots,m,\ \forall k = 1,\cdots,K \end{cases}$$

- Problem **RE**formulation : column generation

New version of Notation:
x_j: number of times pattern j is uesd
a_{ij}: number of times item i is cut in pattern j

Q: what is "pattern"?
A: for example, $W = 100$, $n_i = \{100,200,300\}$, $w_i = \{25,35,45\}$ $(i = 1,2,3)$.
The large roll can be cut into:
Pattern 1: 4 items with each width is $w_1 = 25 \rightarrow a_{11} = 4$, $a_{21} = 0$, $a_{31} = 0$

Pattern 2: 1 item with width is $w_1 = 25$ and 2 items with each width is $w_2 = 35$
$\rightarrow a_{12} = 1, a_{22} = 2, a_{32} = 0$

Pattern 3: 2 items with each width is $w_3 = 45$ $\rightarrow a_{13} = 0, a_{23} = 0, a_{33} = 2$
……

i: 物件(item) $i = 1,\cdots,m$
j: 切割模式(pattern) $j = 1,\cdots,n$
PS: we assume that the total number of patterns is n. But actually, we do not need to know the exact number for n, this notation is just for modeling.
a_{ij}: 切割模式 j 提供物件 i 的数量（在一条卷钢上）
x_j: 切割模式 j 使用的次数

$$(\mathrm{P}_2)\begin{cases} \text{minimize} & \displaystyle\sum_{j=1}^{n} x_j \qquad (\text{minimize the total number of rolls}) \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} a_{ij}x_j \geq n_i \quad i = 1,\cdots,m \quad (\text{the demands for items should be satisfied}) \\ & x_j \in \mathbb{Z}^+ \cup \{0\} \quad j = 1,\cdots,n \end{cases}$$

One potential (essential) constraint is missing: the width limitaion!
Namely: $\sum_{i=1}^{m} a_{ij}w_i \leq W$

- Linear relaxation of (P_2): master problem

$$(\text{Master Problem})\begin{cases} \text{minimize} & \displaystyle\sum_{j=1}^{n} x_j \qquad (\text{minimize the total number of rolls}) \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} a_{ij}x_j \geq n_i \quad i = 1,\cdots,m \quad (\text{the demands for items should be satisfied}) \\ & x_j \in \mathbb{R}^+ \cup \{0\} \quad j = 1,\cdots,n \ (\iff x_j \geq 0 \quad j = 1,\cdots,n) \end{cases}$$

Find $\boldsymbol{a}_j$, such that $c_j - \boldsymbol{c}_B^T\boldsymbol{B}^{-1}\boldsymbol{a}_j < 0$
(我的问题: 为什么这里要谈到主问题的对偶问题，求解时只是用到了对偶乘子)
Recalling the symmetric pair, consider the dual problem of the master problem:

$$(\text{Dual Problem})\begin{cases} \text{maximize} & \displaystyle\sum_{i=1}^{m} n_i y_i \\ \text{subject to} & \displaystyle\sum_{i=1}^{m} a_{ij}y_i \leq 1 \quad j = 1,\cdots,n \\ & y_i \geq 0 \quad i = 1,\cdots,m \end{cases}$$

$$\text{minimize} \quad c_j - \boldsymbol{\pi}^T \mathbf{y}$$

$$\text{subject to} \quad \sum_{i=1}^{m} a_{ij} y_i \leq W$$

where $\boldsymbol{\pi}^T = \boldsymbol{c}_B^T \boldsymbol{B}^{-1}$

We can look for a column (cutting pattern) such that:

$$\kappa := \text{minimize } \{1 - \boldsymbol{\pi}^T \mathbf{y}\} = \text{minimize } \{1 - \sum_{i=1}^{m} \pi_i y_i\} = 1 - \text{maximize} \sum_{i=1}^{m} \pi_i y_i$$

while subject to:

$$\begin{cases} \sum_{i=1}^{m} a_{ij} y_i \leq W \\ y_i \in \mathbb{Z}^+ \cup \{0\} \quad i = 1, \cdots, m \end{cases}$$

This process is equivalent to solve:

$$\text{(Knapsack problem)} \begin{cases} \text{maximize} & \sum_{i=1}^{m} \pi_i y_i \\ \text{subject to} & \sum_{i=1}^{m} a_{ij} y_i \leq W \\ & y_i \in \mathbb{Z}^+ \cup \{0\} \quad i = 1, \cdots, m \end{cases}$$

Knapsack Problem is an "easy" NP-hard problem and can be solved in $O(mW)$ time by dynamic programming.

- The key steps of column generation

Start with initial columns of master problems. For instance, use the simple pattern to cut a roll into $\lfloor W/w_i \rfloor$ pieces with each width is w_i, thus $\boldsymbol{A}$ is a diagonal matrix.

Loop:

1. Solve the restricted LP master problem.
   Let $\boldsymbol{\pi}$ be the optimal simplex multipliers ($\boldsymbol{\pi}^T = \boldsymbol{c}_B^T \boldsymbol{B}^{-1}$).
2. Identify a new feasible column by solving the knapsack subproblem.
   Calculate the reduced cost κ.
3. Add the new column to master problem.

until $\kappa \geq 0$

Finally, generate the optimal plan for cutting stock.