Lecture 8: Column Generation

(3 units)

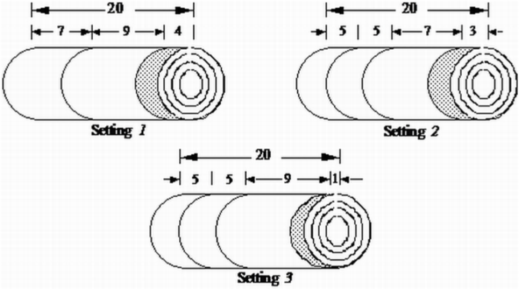
Outline

- Cutting stock problem
- Classical IP formulation
- Set covering formulation
- Column generation
- A dual perspective

Cutting stock problem

Problem description

- A paper mill has a number of rolls of paper of fixed width. Different customers want different numbers of rolls of various-sized widths.
- How are you going to cut the rolls so that you minimize the waste (amount of left-overs)?

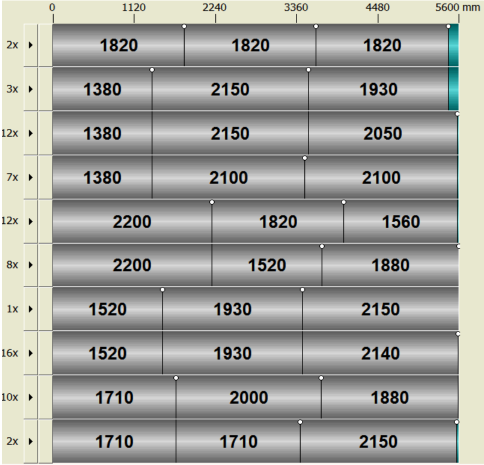


An example

The width of large rolls: 5600mm. The width and demand of customers:

Width	1380	1520	1560	1710	1820	1880	1930	2000	2050	2100	2140	2150	2200
Demand	22	25	12	14	18	18	20	10	12	14	16	18	20

An optimal solution:



Classical IP formulation

- Suppose the fixed width of large rolls is W. The m customers want n_i rolls of width w_i ($i = 1, \dots, m$), ($w_i \leq W$).
- Notations:
  - $\mathcal{K}$: Index set of available rolls
  - $y_k = 1$ if roll k is cut, 0 otherwise
  - x_i^k: number of times item i is cut on roll k
- The IP formulation of Kantorovich:

$$
\begin{aligned}
(P_1) \qquad & \min \sum_{k \in \mathcal{K}} y_k \\
& \text{s.t.} \sum_{k \in \mathcal{K}} x_i^k \geq n_i, \quad i = 1, \dots, m, \quad \text{(demand)} \\
& \qquad \sum_{i=1}^{n} w_i x_i^k \leq W y_k, \quad k \in \mathcal{K}, \quad \text{(width limitation)} \\
& \qquad x_i^k \in \mathbb{Z}_+, \; y_k \in \{0, 1\}.
\end{aligned}
$$

- However, the IP formulation (P_1) is inefficient both from computational and theoretical point views. For example, when the number of rolls is 100 and the number of items is 20, the problem could not be solved to optimality in days (CPLEX).
- The main reason is that the linear program (LP) relaxation of (P_1) is poor. Actually, the LP bound of (P_1) is $\frac{\sum_{i=1}^m w_i n_i}{W}$.

$$
\begin{aligned}
Z^{LP} &= \sum_{k\in\mathcal{K}} y_k = \sum_{k\in\mathcal{K}} \sum_{i=1}^m \frac{w_i x_i^k}{W} \\
&= \sum_{i=1}^m w_i \sum_{k\in\mathcal{K}} \frac{x_i^k}{W} = \sum_{i=1}^m \frac{w_i n_i}{W}.
\end{aligned}
$$

- Question: Is there an alternative IP formulation?

Set covering formulation of Gilmore and Gomory

- Let
  - x_j = number of times pattern j is used
  - a_{ij} = number of times item i is cut in pattern j
- For example, the fixed width of large rolls is $W = 100$ and the demands are $n_i = 100, 200, 300$, $w_i = 25, 35, 45$ $(i = 1, 2, 3)$. The large roll can be cut into
  - Pattern 1: 4 rolls each of width $w_1 = 25 \Rightarrow a_{11} = 4$
  - Pattern 2: 1 roll with width $w_1 = 25$ and 2 rolls each of width $w_2 = 35 \Rightarrow a_{12} = 1$, $a_{22} = 2$
  - Pattern 3: 2 rolls with width $w_3 = 45 \Rightarrow a_{33} = 2$
  - $\cdots$

- Set covering formulation:

$$
\begin{aligned}
(P_2) \qquad & \min \ \sum_{j=1}^{n} x_j \\
& \text{s.t.} \ \sum_{j=1}^{n} a_{ij} x_j \geq n_i, \quad i = 1, \ldots, m, \quad \text{(demand)} \\
& \quad x_j \in \mathbb{Z}_+, \ j = 1, \ldots, n,
\end{aligned}
$$

where n is the total number of cutting patterns satisfying $\sum_{i=1}^{m} w_i a_{ij} \leq W$.

- Each column in (P_2) represents a cutting pattern.
- How many columns (cutting patterns) are there? It could be as many as $\frac{m!}{\bar{k}!(m-\bar{k})!}$, where $\bar{k}$ is the average number of items in each cutting patterns. Exponentially large!.

- Let's first consider the LP relaxation of (P_2):

$$
\begin{aligned}
(LPM) \qquad \min \ & \sum_{j=1}^{n} x_j \\
\text{s.t. } & \sum_{j=1}^{n} a_{ij} x_j \geq n_i, \quad i = 1, \ldots, m, \quad \text{(demand)} \\
& x_j \in \mathbb{R}_+, \ j = 1, \ldots, n,
\end{aligned}
$$

  which is called Linear Programming Master Problem. The solution to (LPM) could be fractional, It is possible to round up the fractional solution to get a feasible solution to (P_2).
- The Simple algorithm for Master LP:
  - Select initial solution
  - Find the most negative reduced cost (new basic variable)
  - Find variable to replace (new non-basic var)
  - Repeat until there exists no variables with negative reduced cost exists

- Consider the standard LP and its dual:

$$\begin{aligned} \min\ & c^T x \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \min\ & b^T \pi \\ \text{s.t. } & A^T \pi \leq c \end{aligned}$$

- Simplex Tableau

$$\begin{array}{cc|c} x_B & x_N & \text{rhs} \\ \hline B & N & b \\ \hline c_B^T & c_N^T & 0 \\ \hline \end{array} \Longrightarrow \begin{array}{cc|c} x_B & x_N & \text{rhs} \\ \hline I & B^{-1}N & b \\ \hline 0 & c_N^T - c_B^T B^{-1} N & -c^T B^{-1} b \\ \hline \end{array}$$

- Reduced cost of non-basic variable: $c_j - c_B^T B^{-1} a_j$, where a_j is a column of A. If $c_j - c_B^T B^{-1} a_j < 0$ implies the current basis can be improved, otherwise, if $c_j - c_B^T B^{-1} a_j \geq 0$ for all $j \in \mathcal{N}$, then the current solution is optimal and $\pi^T = c_B^T B^{-1}$ is dual feasible (optimal).

- What if we have extremely many variables?
- Even if we had a way of generating all the columns (cutting patterns), the standard simplex algorithm will need to calculate the reduced cost for each non-basic variable, which is impossible if n is huge (out of memory).
- A crucial insight: The number of non-zero variables (the basis variables) is equal to the number of constraints, hence even if the number of possible variables (columns) may be large, we only need a small subset of these columns in the optimal solution. In the case of cutting stock problem, the number of items is usually much smaller than the number of cutting patterns.

- The main idea of column generation is to start with a small subset $\mathcal{P} \subset \{1, \dots, n\}$ such that the following subproblem is feasible:

$$
\begin{aligned}
(RLPM) \qquad \min \ & \sum_{j \in \mathcal{P}} x_j \\
\text{s.t. } & \sum_{j \in \mathcal{P}} a_{ij} x_j \ge n_i, \quad i = 1, \dots, m, \\
& x_j \ge 0, \ j \in \mathcal{P},
\end{aligned}
$$

- Recall that the dual problem of (LPM) is

$$
\begin{aligned}
\max \ & \sum_{i=1}^{m} n_i \pi_i \\
\text{s.t. } & \sum_{i=1}^{m} a_{ij} \pi_i \le 1, \quad j \in \mathcal{P}, \\
& \pi_i \ge 0, \ i = 1, \dots, m.
\end{aligned}
$$

Generating columns

- Our next task is to find a column (cut pattern) in $\{1, \dots, n\} \setminus \mathcal{P}$ that could improve the current optimal solution of the linear relaxation ($RLPM$).
- Given the optimal dual solution $\bar{\pi}$ of ($RLPM$), the reduced cost of column $j \in \{1, \dots, n\} \setminus \mathcal{P}$ is

$$1 - \sum_{i=1}^{m} a_{ij}\bar{\pi}_i.$$

- A naive way of finding the new column:

$$\min\{1 - \sum_{i=1}^{m} a_{ij}\bar{\pi}_i \mid j \in \{1, \dots, n\} \setminus \mathcal{P}\},$$

which is impractical because we are not able to list all cutting patterns in real applications.

Knapsack subproblem

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

$$\begin{aligned} \min\ & 1 - \sum_{i=1}^{m} \bar{\pi}_i y_i = 1 - \max \sum_{i=1}^{m} \bar{\pi}_i y_i \\ \text{s.t. } & \sum_{i=1}^{m} w_i y_i \leq W, \\ & y_i \in \mathbb{Z}_+,\ i = 1, \ldots, m, \end{aligned}$$

  where $y = (y_1, \ldots, y_m)$ represents a column $(a_{1j}, \ldots, a_{mj})^T$ (a cutting pattern) and the constraints $\sum_{i=1}^{m} w_i y_i \leq W$, $y \in \mathbb{Z}_+^m$, enforce that y satisfies the conditions for a feasible cutting pattern.
- This is a Knapsack Problem (an "an easy NP-hard problem) and can be solved in $O(mW)$ time by dynamic programming.

Column generation algorithm

The Column Generation Algorithm

Start with initial columns A of (LPM). For instance, use the simple pattern to cut a roll into $\lfloor W/w_i \rfloor$ rolls of width w_i, A is a diagonal matrix.

repeat

1. Solve the restricted LP master problem $(RLPM)$. Let $\bar{\pi}$ be the optimal multipliers ($\bar{\pi}^T = c_B^T B^{-1}$).
2. Identify a new column by solving the knapsack subproblem with optimal value κ.
3. Add the new column to master problem $(RLPM)$

until $\kappa \geq 0$

An example

- A steel company wants to cut the steel rods of width 218cm. The customers want 44 pieces of width 81 cm., 3 pieces of width 70 cm. and 48 pieces of width 68 cm.
- Initial master problem (one item in each large rod):

$$\begin{aligned} \min \ & x_1 + x_2 + x_3 \\ \text{s.t. } & x_1 \geq 44, \\ & x_2 \geq 3, \\ & x_3 \geq 48. \end{aligned}$$

  Optimal multipliers: $\bar{\pi} = (1, 1, 1)^T$.
- Initial knapsack subproblem:

$$\begin{aligned} \max \ & y_1 + y_2 + y_3 \\ \text{s.t. } & 81y_1 + 70y_2 + 68y_3 \leq 218, \\ & y \in \mathbb{Z}^3_+. \end{aligned}$$

- Optimal solution to the initial knapsack subproblem: $y = (0, 0, 3)^T$ with $\kappa = 1 - 3 = -2 < 0$.
- Second master problem:

$$\begin{aligned} \min \ & x_1 + x_2 + x_3 + x_4 \\ \text{s.t. } & x_1 \geq 44, \\ & x_2 \geq 3, \\ & x_3 + 3x_4 \geq 48, \end{aligned}$$

  Optimal multipliers: $\bar{\pi} = (1.0, 1.0, 0.33)^T$.
- Second knapsack subproblem:

$$\begin{aligned} \max \ & y_1 + y_2 + 0.33y_3 \\ \text{s.t. } & 81y_1 + 70y_2 + 68y_3 \leq 218, \\ & y \in \mathbb{Z}^3_+. \end{aligned}$$

  Optimal solution: $y = (0, 3, 0)^T$ with $\kappa = 1 - 3 = -2 < 0$.
  Continue ...

Getting integer solutions

- After solving the master linear program, we still have to find the integer solution to the original problem (P_2).
- Let $x'_j = \lfloor x_j \rfloor$, x_j is integral and $\sum_{j=1}^{n} a_{ij} x'_j \geq \sum_{j=1}^{n} a_{ij} x_j \geq n_i$. So the rounding up solution is feasible to (P_2).
- We can also use branch-and-bound framework to get the optima; integral solution. Branch-and-Price algorithm.

A Dual perspective

- Consider the classical formulation of cutting stock problem:

$$
\begin{aligned}
(P_1) \qquad & \min \sum_{k \in \mathcal{K}} y_k \\
& \text{s.t.} \sum_{k \in \mathcal{K}} x_i^k \geq n_i, \quad i = 1, \ldots, m, \quad \text{(demand)} \\
& \qquad \sum_{i=1}^{n} w_i x_i^k \leq W y_k, \quad k \in \mathcal{K}, \quad \text{(width limitation)} \\
& \qquad x_i^k \in \mathbb{Z}_+, \ y_k \in \{0, 1\}.
\end{aligned}
$$

- How to get a Lagrangian relaxation? Observe that if the demand constraints are removed, the problem can be decomposed into K knapsack problem!

- For $u_i \geq 0$, $i = 1, \ldots, m$, the Lagrangian function is

$$
\begin{aligned}
L(u) \quad = \quad & \min \sum_{k \in \mathcal{K}} y_k + \sum_{i=1}^{m} u_i \left(n_i - \sum_{k \in \mathcal{K}} x_i^k \right) \\
& \text{s.t.} \ \sum_{i=1}^{m} w_i x_i^k \leq W y_k, \ \ k \in \mathcal{K}, \\
& \qquad x_i^k \in \mathbb{Z}_+, \ \ y_k \in \{0, 1\}.
\end{aligned}
$$

- So

$$
L(u) = \sum_{k \in \mathcal{K}} L_k(u) + \sum_{i=1}^{m} n_i u_i
$$

where

$$
\begin{aligned}
L_k(u) \quad = \quad & \min y_k - \sum_{i=1}^{m} u_i x_i^k \\
& \text{s.t.} \ \sum_{i=1}^{m} w_i x_i^k \leq W y_k, \\
& \qquad x_i^k \in \mathbb{Z}_+, \ \ y_k \in \{0, 1\}.
\end{aligned}
$$

- $L_s(u) = \min(0, 1 - z^*)$, where

$$\begin{aligned} z^* &= \max \sum_{i=1}^{m} u_i x_i^k \\ &\quad \text{s.t. } \sum_{i=1}^{m} w_i x_i^k \leq W, \\ &\quad\quad x_i^k \in \mathbb{Z}_+. \end{aligned}$$

  This is a knapsack problem. Notice that $L_s(u)$ is independent of k.

$$L(u) = KL_s(u) + \sum_{i=1}^{m} n_i u_i,$$

  where $K = |\mathcal{K}|$.

- Theorem: the dual problem $\max_{u\geq 0} L(u) = v(LPM)$.
- Proof: Let $z_j = (a_{1j}, \ldots, a_{mj})^T$, $j = 1, \ldots, n$, be the extreme points of the convex hull of the integer set

$$X = \{x \in \mathbb{Z}_+^m \mid \sum_{i=1}^m w_i x_i \leq W\}.$$

Then

$$L_s(u) = \min(0, 1 - \max_{j=1,\ldots,n} \sum_{i=1}^m a_{ij} u_i) = \min_{j=1,\ldots,n} \min(0, 1 - \sum_{i=1}^m a_{ij} u_i).$$

So that

$$\max_{u\geq 0} L(u) = \max_{u\geq 0} \min_{j=1,\ldots,n} \left(K \min(0, 1 - \sum_{i=1}^m a_{ij} u_i) + \sum_{i=1}^m n_i u_i \right)$$

- This can be reduced to

$$
\begin{aligned}
&\max\ z\\
&\text{s.t. } z \leq \sum_{i=1}^{m} n_i u_i,\\
&\qquad z \leq K(1 - \sum_{i=1}^{m} a_{ij} u_i) + \sum_{i=1}^{m} n_i u_i, \quad j = 1, \ldots, n,\\
&\qquad u_i \geq 0,\ i = 1, \ldots, m.
\end{aligned}
$$

The dual of the above problem is

$$
\begin{aligned}
&\min\ \sum_{j=1}^{n} K\lambda_j\\
&\text{s.t. } \lambda_0 + \sum_{j=1}^{n} \lambda_j = 1,\\
&\qquad -n_i(\lambda_0 + \sum_{j=1}^{n} \lambda_j) + \sum_{j=1}^{n} a_{ij} K \lambda_j \geq 0, \quad i = 1, \ldots, m\\
&\qquad \lambda_j \geq 0,\ j = 0, 1, \ldots, n.
\end{aligned}
$$

- Notice that λ_0 can be eliminated from the second constraint. So the constraint $\lambda_0 + \sum_{j=1}^{n} \lambda_j = 1$ is redundant provided that there is $(\lambda_1, \ldots, \lambda_n)$ satisfying

$$\sum_{j=1}^{n} a_{ij} K \lambda_j \geq n_i, \;\; i = 1, \ldots, m, \; \text{and} \; \sum_{j=1}^{n} \lambda_j \leq 1.$$

  This is always possible when K is large (enough rolls).
- Now, let $x_j = K\lambda_j$, we obtain

$$\begin{aligned} \min \;\; & \sum_{j=1}^{n} x_j \\ \text{s.t.} \;\; & \sum_{j=1}^{n} a_{ij} x_j \geq n_i, \quad i = 1, \ldots, m, \\ & x_j \geq 0, \; j = 0, 1, \ldots, n. \end{aligned}$$

  This is exactly the LP relaxation (LPM).