***

**

LI · Zikang

IEDA
The Hong Kong University of Science and Technology

Table of contents

Dynamic Model

$$V_t(x) = \sum_i \lambda_i \left[\max_{u \in \{0,1\}} \{ s_i u + V_{t+1}(x - (1+s_i)u) \} \right] + (1 - \sum_i \lambda_i) V_{t+1}(x)$$

$$= \sum_i \lambda_i \left[\max\{ s_i + V_{t+1}(x - 1 - s_i), V_{t+1}(x) \} \right] + (1 - \sum_i \lambda_i) V_{t+1}(x)$$

$$= V_{t+1}(x) + \sum_i \lambda_i \left[\max\{ s_i + V_{t+1}(x - 1 - s_i) - V_{t+1}(x), 0 \} \right]$$

- x, remaining capacity.
- s_i, the number of people ith type of group contains. If we suppose the number is continuous, then $s = \{1, \ldots, n\}$.
- λ_i: the probability of an arrival of s_i.
- u, binary variable. $u = 1$, accept.

Properties of marginal value

- $f(x) = \max_{a=\{0,1\}}\{a(p-1) + g(x-ap)\}$, $p = s_i + 1 > 0, a = u$.
- $f(x) = V_t(x)$, $g(x) = V_{t+1}(x)$.

- $f(x)$ is not concave means the marginal values $\Delta V_j(x)$ is not decreasing in the remaining capacity at a given stage j, is not increasing in the number of stages remaining at a given capacity level x.

- $V_{t+2}(x - s_i - 1) - V_{t+1}(x - s_i - 1) \leq V_{t+2}(x) - V_{t+1}(x) \Rightarrow$ $V_t(x) - V_{t+1}(x) \geq V_{t+1}(x) - V_{t+2}(x)$.

- Only have $V_t(x) \geq V_{t+1}(x)$, $V_t(x) \leq V_t(x+1)$.
- No general rules.

Our model

$$V_t(x) = V_{t+1}(x) + \sum_i \lambda_i \left[\max\{s_i + V_{t+1}(x-1-s_i) - V_{t+1}(x), 0\}\right]$$

Let $\Delta_i V_{t+1}(x) = V_{t+1}(x) - V_{t+1}(x-1-s_i)$

To accept s_i means $s_i \geq \Delta_i V_{t+1}(x)$.
To reject s_i means $s_i < \Delta_i V_{t+1}(x)$.

Some corollaries

- If $s_i = k$ is accepted, we will accept $s_i = k+1$ only when $1 \geq V_{t+1}(x-1-s_i) - V_{t+1}(x-2-s_i)$.
- If $s_i = k$ is rejected, we will reject $s_i = k-1$ only when $1 \geq V_{t+1}(x-s_i) - V_{t+1}(x-1-s_i)$.

- Fix some stage, when remaining seats $\to \infty$, the decision is to accept all groups of different sizes.

- E_T denotes the convergence value at the last stage, which equals to the expectation of s_i, i.e. $E_T = E(s)$. s_i is the number of people in a group.

- $E_t = (T-t+1)E_T$. S_t denotes the number of remaining seats when $V_t(x)$ converges for the first time at stage t, then S_{t-1} is at most $S_t + (\max\{s_i\}+1)$. That is, the period is $\max\{s_i\}+1$.

Explanation

- When $s_i \geq V_{t+1}(x) - V_{t+1}(x-1-s_i)$, then $s_i + 1 \geq V_{t+1}(x) - V_{t+1}(x-1-s_i) + 1 \geq V_{t+1}(x) - V_{t+1}(x-2-s_i)$, the latter inequality holds because one seat can only provide at most one unit of value. That is, $1 \geq V_{t+1}(x-1-s_i) - V_{t+1}(x-2-s_i)$.
- Similarly, because $V_{t+1}(x-s_i) \leq V_{t+1}(x-1-s_i) + 1$, $s_i - 1 \leq V_{t+1}(x) - V_{t+1}(x-1-s_i) - 1 \leq V_{t+1}(x) - V_{t+1}(x-s_i)$
- Fix the last stage, it is clear that when remaining seats are large enough, the convergence value of $V_T(x)$ will be $E_T = E(s)$ because we will accpet any group.

A special case

- For the case that we only have $\{1, 2\}$ people in a group.
- For $V_t(3)$, always reject 1; For $V_t(2)$, always reject 2.
- For $V_t(s), s = 4, 5$, we will always accept $\{1, 2\}$ when $\lambda_1 \geq \lambda_2$.
- For $V_t(s), s \geq 6$, this policy is invalid.
- But $V_t(x) \leq V_t(x-1) + 1$ always holds, for any $x > 1$ when given any stage t.
- $V_t(3) - V_t(2) \leq 1, V_t(2) - V_t(1) \leq 1 \Rightarrow V_{t-1}(5) - V_{t-1}(4) \leq 1$.
- To prove t at first, then prove s.

Proof

When $\lambda_1 \geq \lambda_2$.

- Periodicity
  $1 + V_t(3n+1) < V_t(3n+3), t < T_0$.
  $1 + V_t(3n) \geq V_t(3n+2), 1 + V_t(3n-1) \geq V_t(3n+1)$.
- Under the assumption, show $V_t(3n+1) - V_t(3n) \leq 1$, $V_t(3n+2) - V_t(3n+1) \leq 1$ and $V_t(3n+3) - V_t(3n+2) \leq 1$ respectively.
- $V_{t-1}(x) - V_t(x) \geq V_{t-1}(x-s-1) - V_t(x-s-1) \Rightarrow$ $V_t(x) \leq V_t(x-s-1) + s$ always holds. $s = \max\{s_i\}$.

Some Notes

- At first, $\lambda_1 \geq \lambda_2 \Rightarrow V_T(3) \leq V_T(2) + 1 \Rightarrow V_t(3) \leq V_t(2) + 1$.
- Similarly, we have $V_t(4) \leq V_t(3) + 1, V_t(5) \leq V_t(4) + 1$.
- In order to prove that $V_t(x) \leq V_t(x-1) + 1$, we have to make sure whether we reject $\{1\}$ first.
- $V_{T-1}(6) \leq V_{T-1}(4) + 1$ needs $\lambda_1^2 \geq \lambda_2 \Rightarrow \lambda_2 \leq \frac{3-\sqrt{5}}{2}$.
- Fix s_i, $\Delta_i V_t(x+1) \leq \Delta_i V_t(x)$ and $\Delta_i V_{t+1}(x) \leq \Delta_i V_t(x)$ do not hold.

Notation Description

- S State: Remaining seats.
- a Action: Accept,1 or Reject,0.
- T Stage: $|T|$ The remaining number of groups need to be served.
- $V(S,T)$: Value function at state S in stage T.
- Reward(value at each stage): $|s_i|$ The number of people the current group contains.

Bellman equation

- When $T > 1$

$$V(S,T) = \sum_{i \in N} p_i \max\{[V((S - s_i - 1), T - 1) + s_i], V(S, T - 1)\}$$

- When $T = 1$

$$V(S,1) = \sum_{i \in N} p_i \max\{[V((S - s_i), 0) + s_i], V(S, 0)\}$$

S remaining seats and T remaining stages.
Explanation: At the last stage, no need to leave space.

Boundary conditions

- When $t \geq 0$

$$\mathrm{V(k,t)} = \begin{cases} -\infty, k < 0 \\ 0, \ k = 0 \end{cases}$$

- When $t = 0$

$$\mathrm{V(k,t)} = 0, \mathrm{k} > 0$$

An example

Suppose that the groups $s_i, i \in N$ satisfy a certain distribution. For example, a fixed discrete distribution.

s_i	1	2	3	4
p_i	0.2	0.2	0.3	0.3

Let $S = 14, T = 5$, then we have

$$\begin{aligned} V(14,5) = 0.2[\max\{V(12,4)+1, V(14,4)\}] \\ +0.2[\max\{V(11,4)+2, V(14,4)\}] \\ +0.3[\max\{V(10,4)+3, V(14,4)\}] \\ +0.3[\max\{V(9,4)+4, V(14,4)\}] \end{aligned}$$

$$\ldots$$

Result

stage	one	two	three	four	five
0	0	0	0	0	0
0	0.2	0.2	0.2	0.2	0.2
0	0.6	0.68	0.744	0.7952	0.83616
0	1.5	1.6	1.68	1.744	1.7952
0	2.7	2.79	2.853	2.8971	2.92797
0	2.7	3.24	3.468	3.6276	3.73932
0	2.7	3.78	3.908	3.9956	4.05692
0	2.7	4.41	4.624	4.7226	4.7857
0	2.7	5.04	5.473	5.6491	5.75197
0	2.7	5.4	6.147	6.4551	6.63945
0	2.7	5.4	6.696	6.9912	7.18344
0	2.7	5.4	7.245	7.6032	7.76592
0	2.7	5.4	7.695	8.2977	8.56207
0	2.7	5.4	7.992	8.9742	9.32434
0	2.7	5.4	8.1	9.5409	9.96271

Deterministic Model

Deterministic Situation

- For example, $(g_1, g_2, g_3, g_5, g_6) = (3, 5, 7, 10, 6)$, where g_i indicates the number of group containing i people.
- Use DP to solve $f_n(g_1, g_2, \ldots, g_m) = f_1(\cdots) \times f_{n-1}(\cdots)$.
- $f_1(x_1, x_2, \ldots, x_m) = 1$, we can find a way to place these groups in one line of seats.
- Drawback: the dimension is high.
- Change it to a cutting stock problem.

Deterministic Situation

- Introduce the formulation and the column generation method as follows. Let the k-th placing pattern of a line of seats with length S into some of the m group types be denoted as a vector $(t_1^k, t_2^k, \ldots, t_m^k)$. Here, t_i^k represents the number of times group type i is placed in the k-th placing pattern. For a pattern $(t_1^k, t_2^k, \ldots, t_m^k)$ to be feasible, it must satisfy: $\sum_{i=1}^{m} t_i^k w_i \leq S$.
- Denote by K the current number of placing patterns. This problem is to decide how to place a total number of groups type i at least g_i times, from all the available placing patterns, so that the total number of lines of seats used is minimized.

Deterministic Situation-Use Column Generation

- Master Problem

$$\begin{aligned} \min \quad & \sum_{k=1}^{K} x_k \\ \text{s.t.} \quad & \sum_{k=1}^{K} t_i^k x_k \geq g_i \quad \text{for } i = 1, \ldots, m \\ & x_k \geq 0, \text{integer} \quad \text{for } k = 1, \ldots, K. \end{aligned}$$

- Consider the linear relaxation of the master problem, and the optimal dual variable vector λ. Using λ as the value assigned to each group type i, the next problem is to find a feasible pattern $(y_1, y_2, \ldots, y_m)$ that maximizes the product of λ and y.

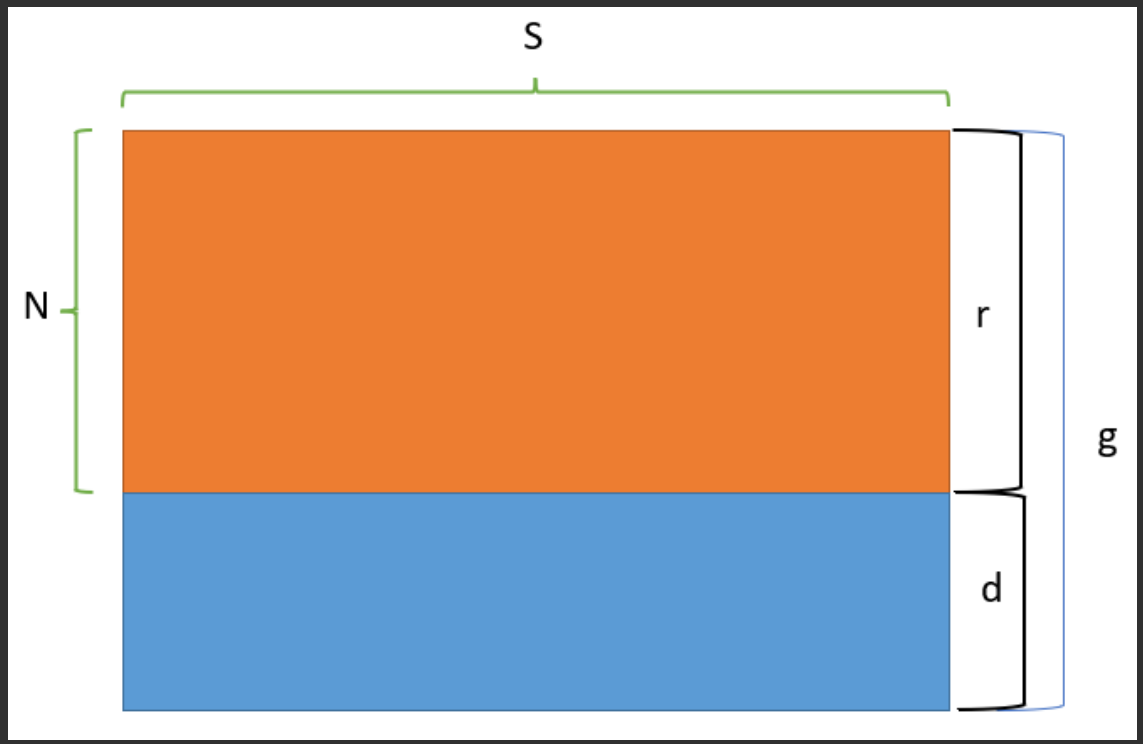
Deterministic Situation-Use Column Generation

- Sub-Problem

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} \lambda_i y_i \\ \text{s.t.} \quad & \sum_{i=1}^{m} w_i y_i \leq S \\ & y_i \geq 0, \text{integer} \quad \text{for } i = 1, \ldots, m. \end{aligned}$$

- This is a knapsack problem; its solution will be used as an additional pattern in the master problem.
- The reduced cost is $c_k - c_B B^{-1} A_k$, where $c_k = 1, c_B B^{-1} = \lambda$.

Illustration



- N: the number of given lines. S: the number of seats.
- g: total demand r: remaining demand d: discarded demand.

Formulation

- Minimize the trim loss. $\min NS - \sum_{i=1}^{m} r_i(s_i - 1) \rightarrow \max \sum_{i=1}^{m} r_i(s_i - 1) \rightarrow \max \sum_{i=1}^{m} (g_i - d_i)(s_i - 1)$.
- Recall that $\sum_{k=1}^{K} t_i^k x_k + d_i = g_i$, by substituting this equation we can obtain the objective function of the following master problem.

$$\begin{aligned}
\max \quad & \sum_{k=1}^{K} (\sum_{i=1}^{m} (s_i - 1) t_i^k) x_k \\
\text{s.t.} \quad & \sum_{k=1}^{K} x_k \leq N \\
& \sum_{k=1}^{K} t_i^k x_k \leq g_i, \quad i = 1, \ldots, m \\
& x_k \geq 0, \quad k = 1, \ldots, K
\end{aligned}$$

Analysis

- Similarly, we consider the linear relaxation of the master problem and the optimal dual variable vector λ, μ.
- Using λ as the value assigned to the first constraint and μ to the second constraints. The problem is to find a feasible pattern $(t_1^{k_0}, t_2^{k_0}, \ldots, t_m^{k_0})$ that maximizes the reduced cost.
- The reduced cost is $c_{k_0} - c_B B^{-1} A_{k_0}$, where $c_{k_0} = \sum_{i=1}^{m} (s_i - 1) t_i^{k_0}$, $c_B B^{-1} = (\lambda, \mu)$, $A_{k_0} = (1, t_1^{k_0}, t_2^{k_0}, \ldots, t_m^{k_0})^T$.
- Use y_i indicate the feasible pattern instead of $t_i^{k_0}$. We can obtain the sub-problem.

Sub-problem

$$\begin{aligned}
\max \quad & \sum_{i=1}^{m} \left[(s_i - 1) - \mu_i \right] y_i - \lambda \\
\text{s.t.} \quad & \sum_{i=1}^{m} s_i y_i \leq S \\
& y_i \geq 0, \text{integer} \quad \text{for } i = 1, \ldots, m.
\end{aligned}$$

- Use column generation to generate a new pattern until all reduced costs are negative.

IP Formulation

$$\begin{aligned}
\text{max} \quad & \sum_{j=1}^{m}\sum_{i=1}^{n}(s_i - 1)x_{ij} \\
\text{s.t.} \quad & \sum_{i=1}^{n} s_i x_{ij} \leq S, \quad j = 1, \ldots, m \\
& \sum_{j=1}^{m} x_{ij} \leq g_i, \quad i = 1, \ldots, n \\
& x_{ij} \geq 0 \text{ and integer}, \quad i = 1, \ldots, n, j = 1, \ldots, m
\end{aligned}$$

- m indicates the number of lines.
- x_{ij} indicates the number of group type i placed in each line j.

Improvement

- For the cutting stock problem, $\min LP \leq \min LP^r \leq \min IP \leq \min IP^r$. After obtaining all patterns with column generation, restricted LP equals LP relaxation, $LP^r = LP$, which provides a lower bound. To obtain a optimal solution, we should implement branch and bound into column generation. This method is called branch-and-price.
- For the new problem, $\max LP \geq \max LP^r \geq \max IP \geq \max IP^r$. The column generation will give the upper bound (LP relaxation) and lower bound (restricted IP).

Branch scheme

- General branch: $\sum_{k \in K(k^j)} x_k = \alpha$, where $K(p) = \{k \in K : k \geq p\}$ (column subsets) for $p \in Z_+^m$, α is fractional.
  $K = \{k \in N^m : \sum_{i=1}^m s_i k_i \leq S\}$ indicate all feasible patterns, and x_k is the number of times pattern k selected.

- Given a feasible solution x^* of LP that is not integral, take k^* to be any maximal element of the set $\{k \in K : x_k^* \notin Z_+\}$. Then the only fractional term in this sum $\sum_{k \in K(k^j)} x_k^*$ is $x_{k^*}^*$.

- Maximal means that the space left after cutting this pattern is less than the size of the smallest group.

Branch scheme

- Generic branching constraints: $\sum_{k \in K(k^j)} x_k \leq U^j, \forall j \in G^u$ and $\sum_{k \in K(k^j)} x_k \geq L^j, \forall j \in H^u$, where G^u and H^u are sets of branching constraints of the form

$$\sum_{k \in K(k)} x_k \leq \lfloor \alpha \rfloor \tag{1}$$

and

$$\sum_{k \in K(k)} x_k \geq \lceil \alpha \rceil \tag{2}$$

, respectively.

- If we can always generate the maximal pattern, then any fractional column defines a branching set which contains only one member.

In our case

Restricted master problem:

$$\begin{aligned}
\max \quad & \sum_{k\in K}(\sum_{i=1}^{m}(s_i-1)t_i^k)x_k \\
\text{s.t.} \quad & \sum_{k\in K} x_k \le N \\
& \sum_{k\in K} t_i^k x_k \le g_i, \quad i=1,\dots,m \\
& \sum_{k\in K(k^j)} x_k \le U^j, \forall j \in G^u \\
& \sum_{k\in K(k^j)} x_k \ge L^j, \forall j \in H^u \\
& x_k \ge 0, \quad k \in K
\end{aligned}$$

In our case

Let (π, λ, μ, v) be optimal dual variables associated with the added constraints. The sub-problem is:

$$\begin{aligned}
\max \quad & [(s_i - 1) - \lambda_i]\, y_i - \pi - \sum_{j \in G^u} \mu_j z^j - \sum_{j \in H^u} v_j z^j \\
\text{s.t.} \quad & \sum_{i=1}^{m} s_i y_i \leq S \\
& z^j = 1 \text{ if } y \geq k^j; z^j = 0 \text{ otherwise} \\
& y_i \geq 0, \text{integer} \quad \text{for } i = 1, \ldots, m.
\end{aligned}$$

$Y_k = \{(y, z) : z = 1, \text{ if } y \geq k, z = 0 \text{ otherwise}\}$ can be formulated as MIPs.

In our case

- The initial patterns are cut with as many copies as possible of each group size. To have the maximal patterns, just by adding as many copies as possible of the smallest group size.
- Our sub-problem is only affected when an upper bound has been placed on the value of the variable associated with the selected pattern. Because this variable could be a nonbasic variable for the LP. In this case, we have avoid regenerating this maximal pattern.
- Branch priority: select the least dominance pattern, and use the form(1). Because if we want to increase the total capacity, the number of low utilization pattern should be reduced.

Other branch-and-price methods

- Transform the original sub-problem to 0-1 knapsack problem.
- Use DW decomposition. Less compact formulation.
- Use the definition directly. The subprobelm can be expressed as a MIP, not be a knapsack problem anymore.

Branch and Price

- Solve restricted master problem with the initial columns.
- Use pricing problem to generate columns.
- When column generation terminates, is the solution integral?
  Yes, update lower bound.
  No, update upper bound. And fathom node or branch to create two nodes.
- Select the next node until all nodes are explored.

Properties

- When column generation gives an integer solution at the first time, then we obtain the optimal solution.
- α_k indicates the number of items for pattern k. β_k indicates the left space for maximal pattern k. Notice that the left space is the true loss.
- Denote $(\alpha_k + \beta_k)$ as the loss for pattern k, $l(k)$. When $l(k)$ reaches minimum, the corresponding pattern k is the optimal solution for a single line.
- If the group sizes are consecutive integers starting from 2, $\{2, 3, \ldots, u\}$, then a greedy-based pattern is optimal, i.e., select the maximal group size,u, as many as possible and the left space is occupied by the group with the corresponding size. The loss is $k + 1$, where k is the number of times u selected. Let $S = u \cdot k + r$, when $r > 0$, we will have at least $\lfloor \frac{r+u}{2} \rfloor - r + 1$ optimal patterns with the same loss. When $r = 0$, we have only one optimal pattern.

- Let I_1 be the set of patterns with the minimal loss. And I_i be the set of patterns with i-th minimal loss.
- When supply is less than demand, we must select I_1.
- When $N \leq \max_{k \in I_1} \min_i \{\lfloor \frac{d_i}{b_i^k} \rfloor\}$, select the corresponding pattern from I_1 and it is the optimal solution. N is the number of lines, $i = 1, 2, \ldots, m$, d_m is the demand of the largest size, b_m^k is the number of group m placed in pattern k.

Integral Decomposition

- Denote by R_+^n the nonnegative orthant of Euclidean n-space. A polyhedron $P \subset R_+^n$ is called lower comprehensive if $0 \leq y \leq x \in P$ implies $y \in P$.
- Let M be the matrix whose rows are the maximal integral points of P.
- An integral vector $x \in P$ is a maximal integral point of P if there is no integral vector $y \in P$ with $x \neq y \geq x$.
- Let kP be defined as $\{kx | x \in P\}$. The integral decomposition property holds for P if, for each integer $k \geq 1$ and each integral vector $x \in kP$, there exist integral vectors $x^i \in P, 1 \leq i \leq k$, for which $x = \sum_{i=1}^{k} x^i$.
- The knapsack polyhedron, denoted by P, is the convex hull of the set $\{x \in Z_+^n | ax \leq b\}$.

Integer Round Up for CSP

- Cutting stock problem has the integer round-up property if and only if P has the integral decomposition property.
- Any two-dimension CSP has the IRU property.
- The numerical experiment shows that the form of $\{2, 3, \ldots, n\}$ has the IRU property.
- But we can check this property finitely. From S. BAUM and L.E. TROTTER.

Integer Round Up for CSP

- After the column generation gives the LP solution(fractional), calculate the supply quantity. If the number is integral, keep it. If the supply is not integral, we can construct an integer vector which can provide the largest integral profit.
- Construction: Calculate the space taken by fractional supply(the value must be integer), increase the corresponding supply having the same space, delete the fractional part.
- Now we have an integral supply vector. If this case has IRU property, there exists an integral decomposition.
- Minus the integral part of the patterns from the supply. Check if the rest can be placed in several lines.

Need to do

- If the left space is 1? solved.
- If the left, subset sum problem.
- Check when L $= 22$

Reformulate our model as:

$$\begin{aligned} \max \quad & yMc \\ \text{s.t.} \quad & yM \leq w_0 \\ & \sum y \leq N \\ & y \geq 0 \text{ and integer} \end{aligned}$$

w_0 is the demand. N is the number of given lines.

$$\begin{aligned} \min \quad & \sum y \\ \text{s.t.} \quad & yM \geq w \\ & y \geq 0 \text{ and integer} \end{aligned}$$

$$\begin{aligned} \max \quad & cw \\ \text{s.t.} \quad & w \in \{Nx | x \in P\} \\ & w \leq w_0 \end{aligned}$$

When P has the integral decomposition property, there exist N integer vectors decomposing every w. Now, the question is how to find the maximal w.

Question

- Which patterns will be covered?
- Only select I_1, demand has to satisfy which condition?
- We only need I_1 and I_2 when supply and demand?

Lower Bound for Demand

$$\begin{aligned}
\max \quad & \sum_{k=1}^{K}\Big(\sum_{i=1}^{m}(s_i-1)t_i^k\Big)x_k \\
\text{s.t.} \quad & \sum_{k=1}^{K} x_k \le N \\
& \sum_{k=1}^{K} t_i^k x_k \le g_i, \quad i=1,\dots,m \\
& \sum_{k=1}^{K} t_i^k x_k \ge p_i, \quad i=1,\dots,m \\
& x_k \ge 0, \quad k=1,\dots,K
\end{aligned}$$

- Use the similar method to solve.

Sub-problem

$$\begin{aligned}
\max \quad & \sum_{i=1}^{m} \left[(s_i - 1) - \lambda_i - \mu_i\right] y_i - \tau \\
\text{s.t.} \quad & \sum_{i=1}^{m} s_i y_i \leq S \\
& y_i \geq 0, \text{integer} \quad \text{for } i = 1, \ldots, m.
\end{aligned}$$

- Use column generation to generate a new pattern until all reduced costs are negative.

IP Formulation

$$\max \quad \sum_{j=1}^{m}\sum_{i=1}^{n}(s_i - 1)x_{ij}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} s_i x_{ij} \leq S, \quad j = 1, \ldots, m$$

$$p_i \leq \sum_{j=1}^{m} x_{ij} \leq g_i, \quad i = 1, \ldots, n$$

$$x_{ij} \geq 0 \text{ and integer}, \quad i = 1, \ldots, n, j = 1, \ldots, m$$

- m indicates the number of lines.
- x_{ij} indicates the number of group type i placed in each line j.

Question

1. Feasibility. At first, use cutting stock model to check if the following constraints provide a feasible solution. If the minimal number of lines we need is less than the given number of lines, these constraints are feasible.

$$\sum_{i=1}^{n} s_i x_{ij} \leq S, \quad j = 1, \ldots, m$$

$$p_i \leq \sum_{j=1}^{m} x_{ij}, \quad i = 1, \ldots, n$$

The End