

1 Outline

In this lecture, we cover

- complementary slackness,
- row and column generation frameworks for large-scale linear programming models,
- the cutting stock problem.

2 Complementary slackness

Strong duality says that if primal and dual are feasible, then their optimal values coincide. Let x and λ be feasible solutions to the primal LP and the dual LP, respectively, such that

$$c^\top x = b^\top \lambda.$$

From our derivations, we have

$$\begin{aligned} b^\top \lambda &= \sum_{i \in M_L} b_i \lambda_i + \sum_{i \in M_E} b_i \lambda_i + \sum_{i \in M_G} b_i \lambda_i \\ &\leq \sum_{i \in M_L} \lambda_i a_i^\top x + \sum_{i \in M_E} \lambda_i a_i^\top x + \sum_{i \in M_G} \lambda_i a_i^\top x \\ &= \sum_{j \in D_L} (\tilde{a}_j^\top \lambda) x_j + \sum_{j \in D_F} (\tilde{a}_j^\top \lambda) x_j + \sum_{j \in D_G} (\tilde{a}_j^\top \lambda) x_j \\ &\leq \sum_{j \in D_L} c_j x_j + \sum_{j \in D_F} c_j x_j + \sum_{j \in D_G} c_j x_j \\ &= c^\top x. \end{aligned}$$

Since $b^\top \lambda = c^\top x$, the inequalities hold with equality throughout. From the first equality, we have

$$0 = \sum_{i \in M_L} \lambda_i (a_i^\top x - b_i) + \sum_{i \in M_E} \lambda_i (a_i^\top x - b_i) + \sum_{i \in M_G} \lambda_i (a_i^\top x - b_i).$$

Recall that

$$(a_i^\top x - b_i) \lambda_i \geq 0$$

for every i . Therefore, we must have

$$(a_i^\top x - b_i) \lambda_i = 0 \quad \forall i \in M_L \cup M_E \cup M_G.$$

From the second equality,

$$0 = \sum_{j \in D_L} (\tilde{a}_j^\top \lambda - c_j) x_j + \sum_{j \in D_F} (\tilde{a}_j^\top \lambda - c_j) x_j + \sum_{j \in D_G} (\tilde{a}_j^\top \lambda - c_j) x_j.$$

As for the first equality, we deduce that

$$(\tilde{a}_j^\top \lambda - c_j) x_j = 0 \quad \forall j \in D_L \cup D_F \cup D_G.$$

Theorem 10.1 (Complementary slackness). *Let x be an optimal solution to the primal LP, and let λ be an optimal solution to the dual LP. Then the following statements hold.*

- If $a_i^\top x \neq b_i$, then $\lambda_i = 0$.
- If $\lambda_i \neq 0$, then $a_i^\top x = b_i$.

Moreover,

- If $\tilde{a}_j^\top \lambda_i \neq c_j$, then $x_j = 0$.
- If $x_j \neq 0$, then $\tilde{a}_j^\top \lambda = c_j$.

In words, **at optimality**,

- primal constraint i and dual variable i **cannot be slack simultaneously**,
- primal variable j and dual constraint j **cannot be slack simultaneously**.

3 Large-scale linear programming models

Consider a linear program and its dual as follows.

$$\begin{array}{ll} p^* = \min & c^\top x \\ \text{s.t.} & Ax \geq b \\ & x \geq 0 \end{array} \quad \begin{array}{ll} d^* = \max & b^\top \lambda \\ \text{s.t} & A^\top \lambda \leq c \\ & \lambda \geq 0 \end{array}$$

where A is an $m \times d$ matrix. Depending on applications, we may have to deal with **extremely large** linear programs. We may have

- a large number of primal constraints m (i.e., dual variables),
- a large number of primal variables d (i.e., dual constraints).

We will consider the case where m or d is large but not both. Throughout this section, we focus on the case when p^* is finite. Then by strong duality, d^* is also finite.

3.1 Large m : primal row generation

When m is large, we have a large number of constraints. Then, instead of solving with the entire set of m constraints, we may start with a subset of constraints, add necessary constraints later, and re-solve updated problems iteratively. More precisely, we consider

$$\begin{array}{ll} p_M & = \min c^\top x \\ & \text{s.t. } a_i^\top x \geq b_i, \quad i \in M \\ & x \geq 0 \end{array}$$

where M is a subset of $[m]$. This linear program is called the **master problem**. The basic outline is as follows.

1. Solve the master problem and obtain an optimal solution x_M .

2. Check whether there is a constraint $a_i^\top x \geq b_i$ among $i \in [m] \setminus M$ that x_M violates.
3. If $a_i^\top x_M < b_i$ for some $i \in [m] \setminus M$, then add constraint $a_i^\top x \geq b_i$ to the master problem.
4. Repeat this procedure.

This procedure is called **(primal) row generation**. It is also known as **constraint generation** or the **cutting plane method**. What follows is a more detailed pseudo-code.

Algorithm 1 Row generation framework

Initialize $M \subseteq [m]$ such that the master problem has a finite optimum.

Set $S = -\infty$.

while $S < 0$ **do**

Solve the **master problem**

$$\begin{aligned} p_M &= \min c^\top x \\ &\text{s.t. } a_i^\top x \geq b_i, \quad i \in M \\ &\quad x \geq 0 \end{aligned}$$

Optimal an optimal solution x_M to the master problem.

Solve the **subproblem**

$$S = \min_{i \in [m]} \left\{ a_i^\top x_M - b_i \right\}.$$

$M \leftarrow M \cup \{i^*\}$ where $i^* \in \operatorname{argmin}_{i \in [m]} \{a_i^\top x_M - b_i\}$.

end while

Return x_M as an optimal solution to the original linear program.

3.2 Large d : row generation in the dual = primal column generation

When d is large, then the linear program has a large number of variables, in which case, its dual has a large number of constraints. Then we may apply the row generation framework to the dual linear program. Recall that the dual linear program is given by

$$\begin{aligned} d^* &= \max b^\top \lambda \\ &\text{s.t. } \tilde{a}_j^\top \lambda \leq c_j, \quad j \in [d] \\ &\quad \lambda \geq 0 \end{aligned}$$

where $\tilde{a}_1, \dots, \tilde{a}_d$ are the columns of the constraint matrix A . Then to apply the row generation framework to the dual, we start with a subset $D \subseteq [d]$. Consider the corresponding master problem as follows.

$$\begin{aligned} d^D &= \max b^\top \lambda \\ &\text{s.t. } \tilde{a}_j^\top \lambda \leq c_j, \quad j \in D \\ &\quad \lambda \geq 0. \end{aligned}$$

We call this the **dual master problem**. Let us take the dual of the dual master problem.

$$\begin{aligned}
 p^D &= \min \sum_{j \in D} c_j x_j \\
 \text{s.t.} \quad &\sum_{j \in D} \tilde{a}_j x_j \geq b_i \\
 &x_j \geq 0, \quad j \in D.
 \end{aligned}$$

Note that the primal linear program can be written as

$$\begin{aligned}
 p^* &= \min \sum_{j \in D} c_j x_j + \sum_{j \in [d] \setminus D} c_j x_j \\
 \text{s.t.} \quad &\sum_{j \in D} \tilde{a}_j x_j + \sum_{j \in [d] \setminus D} \tilde{a}_j x_j \geq b_i \\
 &x_j \geq 0, \quad j \in D \\
 &x_j \geq 0, \quad j \in [d] \setminus D.
 \end{aligned}$$

Here, the dual of the dual master problem is equivalent to what is obtained from the primal linear program after setting $x_j = 0$ for $j \notin D$. Therefore, adding a row to the dual master problem is equivalent to adding a variable/column to the primal problem. Hence, this procedure of row generation in the dual is called **column generation**.

Algorithm 2 Column generation framework

Initialize $D \subseteq [d]$ such that the dual master problem has a finite optimum

Set $S = +\infty$.

while $S > 0$ **do**

 Solve the **dual master problem**

$$\begin{aligned}
 d^D &= \max \quad b^\top \lambda \\
 \text{s.t.} \quad &\tilde{a}_j^\top \lambda \leq c_j, \quad j \in D \\
 &\lambda \geq 0.
 \end{aligned}$$

 Optimal an optimal solution λ^D to the master problem.

 Solve the **subproblem**

$$S = \max_{j \in [d]} \left\{ \tilde{a}_j^\top \lambda^D - c_j \right\}.$$

$D \leftarrow D \cup \{j^*\}$ where $j^* \in \operatorname{argmax}_{j \in [d]} \left\{ \tilde{a}_j^\top \lambda^D - c_j \right\}$.

end while

Solve the primal linear program with columns \tilde{a}_j for $j \in D$ and return an optimal solution.

The column generation framework will solve the dual linear program. Hence, at termination, we would end up with a subset $D \subseteq [d]$ such that $d^D = d^*$. By strong duality, we have

$$p^D = d^D = d^* = p^*.$$

Therefore, solving the primal linear program with columns \tilde{a}_j for $j \in D$ would in turn solve the primal linear program.

4 Cutting stock problem

A steel mill manages steel plate production lines. Each line can produce large steel plate of width W . The steel mill receives orders of different widths. Suppose that there are m different orders of width values w_1, \dots, w_m . Assume also that the number of orders for width w_i is b_i for $i \in [m]$. Upon receiving these orders, the steel mill decides how to allocate the production orders over production lines. Basically, a steel plate of width W from a production line can be cut into multiple plates of different widths. Here, the steel mill wants to run as few production lines as possible.

To model this problem, we use the idea of **cutting patterns**. The basic idea is as follows. A cutting pattern determines how to cut a steel plate of width W into pieces of different widths. A cutting pattern can be represented as a vector $s \in \mathbb{Z}_+^m$ where s_i represents the number of pieces of width w_i . To make sure that s_i pieces of width w_i for $i \in [m]$ can be produced from a steel plate of width W , we impose a knapsack constraint as follows.

$$\sum_{i=1}^m w_i s_i \leq W.$$

Then the set of nonnegative integer vectors satisfying the knapsack constraint, given by

$$\mathcal{S} = \left\{ s \in \mathbb{Z}_+^m : \sum_{i=1}^m w_i s_i \leq W \right\}$$

collects all possible cutting patterns. Essentially, using a cutting pattern is equivalent to operating a production line.

Let x_s denote the number of production lines that produce cutting pattern s . Then

$$\sum_{s \in \mathcal{S}} x_s$$

is the total number of production lines in use. Moreover,

$$\sum_{s \in \mathcal{S}} s_i x_s$$

is equal to the total number of pieces of width w_i . Then the problem can be formulated as

$$\begin{aligned} \min \quad & \sum_{s \in \mathcal{S}} x_s \\ \text{s.t.} \quad & \sum_{s \in \mathcal{S}} s_i x_s \geq b_i, \quad \forall i \in [m], \\ & x_s \in \mathbb{Z}_+, \quad \forall s \in \mathcal{S}. \end{aligned}$$

Here, the issue with this integer program is that the number of variables is equal to the number of all possible cutting patterns. However, the number of possible patterns is the number of points in \mathcal{S} , which can be huge depending on the problem parameters. Instead of enumerating all cutting patterns, we use column generation.

First, we replace the integrality constraint by $x \geq 0$, and as a result, we obtain a linear program.

$$\begin{aligned} \min \quad & \sum_{s \in \mathcal{S}} x_s \\ \text{s.t.} \quad & \sum_{s \in \mathcal{S}} s_i x_s \geq b_i, \quad \forall i \in [m], \\ & x_s \geq 0, \quad \forall s \in \mathcal{S}. \end{aligned}$$

Then we take its dual given by

$$\begin{aligned}
\max \quad & \sum_{i \in [m]} \lambda_i \\
\text{s.t.} \quad & \sum_{i \in [m]} s_i \lambda_i \leq 1, \quad \forall s \in \mathcal{S} \\
& \lambda_i \geq 0, \quad \forall i \in [m].
\end{aligned}$$

The column generation framework starts with a subset $\mathcal{F} \subseteq \mathcal{S}$ of the cutting patterns. Then we solve

$$\begin{aligned}
\min \quad & \sum_{s \in \mathcal{F}} x_s \\
\text{s.t.} \quad & \sum_{s \in \mathcal{F}} s_i x_s \geq b_i, \quad \forall i \in [m], \\
& x_s \in \mathbb{Z}_+, \quad \forall s \in \mathcal{F}.
\end{aligned}$$

This is an integer program, while solving which we may obtain the corresponding dual multipliers λ . Then to apply the column generation framework, we solve the corresponding subproblem

$$\max \left\{ -1 + \sum_{i \in [m]} \lambda_i s_i : s \in \mathcal{S} \right\}.$$

By plugging in the definition of \mathcal{S} , the subproblem is equivalent to

$$\begin{aligned}
\max \quad & -1 + \sum_{i \in [m]} \lambda_i s_i \\
\text{s.t.} \quad & \sum_{i \in [m]} w_i s_i \leq W \\
& s \in \mathbb{Z}_+^m.
\end{aligned}$$

This problem is an instance of the **(unbounded) knapsack problem**.