IE 331 OR: Optimization **KAIST, Spring 2023**
Lecture #16: Integer programming II April 27, 2023
Lecturer: Dabeen Lee

1 Outline

In this lecture, we cover

- facility location review,
- modeling logical relationships with binary variables,
- transportation problem with ramp-up costs.

2 Facility location recap

When modeling the facility location problem, we had to deal with constraints of the form

$$t \geq \min_{i\in[d]:\ x_i=1} f_{ij}, \quad j \in [d].$$

We learned how to convert this type of constraints into linear inequalities subject to adding more binary variables. In this section, we generalize this argument so that we may take care of other constraints of similar forms.

Note that the constraints can be generalized as

$$t \geq \min_{i\in M(x)} f_i$$

where $M(x) \subseteq [d]$ is some index subset that depends on decision variables x. Then the constraint is satisfied if and only if

$$\exists i \in M(x) \quad \text{such that} \quad t \geq f_i. \tag{$\star$}$$

Let E_i denote the **event** that $t \geq f_i$. Here, our modeling trick is to use a binary variable $y_i \in \{0,1\}$ to indicate event E_i. Let us consider the following constraints.

$$\begin{aligned} &\sum_{i\in[d]} y_i = 1,\\ &t \geq \sum_{i\in[d]} f_i y_i,\\ &y_i = 0 \quad \forall i \notin M(x),\\ &y_i \in \{0,1\} \quad \forall i \in [d]. \end{aligned} \tag{$\star\star$}$$

Lemma 16.1. ($\star$) *holds if and only if there is y satisfying* ($\star\star$).

Proof. ($\Rightarrow$) Suppose that $t \geq f_\ell$ for some $\ell \in M(x)$. Then let $y \in \{0,1\}^d$ be the vector with $y_\ell = 1$ and $y_i = 0$ for $i \neq \ell$. Then $\sum_{i\in[d]} y_i = 1$ and $\sum_{i\in[d]} f_i y_i = f_\ell$. Then ($\star\star$) is satisfied, as required.

($\Leftarrow$) Suppose that a binary vector y satisfies ($\star\star$). Then there exists $\ell \in M(s)$ such that $y_\ell = 1$ and $y_i = 0$ for $i \neq \ell$. This implies that $t \geq f_\ell$. Therefore, ($\star$) is satisfied. $\square$

In the facility location example, $M(x)$ was given by

$$M(x) = \{i \in [d] : \; x_i = 1\}.$$

Thus $y_i = 0$ for all $i \notin M(x)$ if and only if $y_i = 0$ for all $i \in [d]$ such that $x_i = 0$. We imposed this condition by the following constraints

$$y_i \leq x_i, \quad i \in [d].$$

In summary, the constraint $t \geq \min_{i \in [d]:\; x_i = 1} f_{ij}$ from facility location can be equivalently formulated as

$$\begin{aligned}
&\sum_{i \in [d]} y_{ij} = 1, \\
&t \geq \sum_{i \in [d]} f_{ij} y_{ij}, \\
&y_{ij} \leq x_i \quad \forall i \in [d], \\
&y_{ij} \in \{0, 1\} \quad \forall i \in [d].
\end{aligned}$$

3 Modeling logical relationships with binary variables

Much of the applicability of integer programming comes from using binary variables to model **logical relationshipns** between different sets of variables and constraints. There are some standard techniques for doing this, which we go through.

- Products of binary variables.
- No-good constraints.
- Implication relationships.
- Big-M technique.
- Disjunctive constraint.

3.1 Product of binary variables

Let $x = (x_1, \ldots, x_d) \in \{0, 1\}^d$ be a **vector of binary variables** with $d \geq 2$, and let $y \in \{0, 1\}$ be a **single binary variable**. We wish to capture the constraint

$$y = x_1 \times x_2 \times \cdots \times x_d.$$

Here, $y = 1$ holds if and only if $x_i = 1$ for all $i \in [d]$, and $y = 0$ holds if and only if $x_i = 0$ for some $i \in [d]$. Again, this constraint is not linear and is non-convex. How do we convert this into a set of linear inequalities? As before, we can add

$$y \leq x_i, \quad i \in [d].$$

With these constraints, we may force $y = 0$ whenever there exists $i \in [d]$ with $x_i = 0$. However, we may still have $y = 0$ even when $x_i = 1$ for all $i \in [d]$. Then how can we force $y = 1$ when $x_i = 1$ for all $i \in [d]$? The following constraint does it.

$$y \geq \sum_{i \in [d]} x_i - (d - 1).$$

Here, if $x_i = 1$ for all $i \in [d]$, then the right-hand side equals $d - (d-1) = 1$, which forces $y = 1$. On the other hand, when $x_i = 0$ for some $i \in [d]$, then the right-hand side is at most 0, in which case the constraint is implied by $y \in \{0,1\}$. In summary, $y = x_1 \times x_2 \times \cdots \times x_d$ holds if and only if

$$y \leq x_i \quad \forall i \in [d], \quad y \geq \sum_{i \in [d]} x_i - (d-1).$$

3.2 No-good constraints

Let $a = (a_1, \ldots, a_d) \in \{0,1\}^d$ be a **fixed** binary vector, and let $x = (x_1, \ldots, x_d) \in \{0,1\}^d$ be a vector of binary **variables**. We want to model the constraint

$$x \in \{0,1\}^d \setminus \{a\}.$$

In words, we want to prevent x from being equal to a. Let us consider the following constraint.

$$\sum_{i \in [d]} (a_i x_i + (1-a_i)(1-x_i)) \leq d - 1.$$

Note that this is equivalent to

$$\sum_{i \in [d]} (1 - 2a_i) x_i \geq 1.$$

For example, when $a = 0$, the constraint becomes

$$\sum_{i \in [d]} x_i \geq 1.$$

The constraint models $x \neq a$. Why is that? Note that

$$a_i x_i + (1-a_i)(1-x_i) = \begin{cases} 1, & \text{if } x_i = a_i, \\ 0, & \text{if } x_i \neq a_i. \end{cases}$$

Moreover,

$$\sum_{i \in [d]} (a_i x_i + (1-a_i)(1-x_i)) > d - 1$$

holds if and only if $a_i x_i + (1-a_i)(1-x_i) = 1$ for all $i \in [d]$. Therefore, the constraint holds if and only if there exists some $i \in [d]$ such that $x_i \neq a_i$.

3.3 Implication relationships

Suppose that we have two binary variables $x, y \in \{0,1\}$. How can we model the **implication relationship** as follows?

$$x = 1 \quad \Rightarrow \quad y = 1.$$

Another name for this is a **forcing constraint**, since whenever $x = 1$ we need to force $y = 1$. In the facility location example, we saw that if $y_{ij} = 1$ (a station at i is selected by suburb j), then we need to force $x_i = 1$ (a station is located at i). We capture this with the linear constraint

$$x \leq y.$$

3.4 Big-M technique for switching constraints on/off

Suppose that we have a linear constraint $a^\top x \leq b$. We have a model where $a^\top x \leq b$ is imposed only under certain conditions. How can we model **switching** the constraint on and off? We can add a binary variable $y \in \{0,1\}$ and attempt to model the logical implications

$$\begin{aligned} y = 1 \quad &\Rightarrow \quad a^\top x \leq b \\ y = 0 \quad &\Rightarrow \quad \text{no constraint imposed} \end{aligned}$$

How can we linearize the two implications? Suppose that $a^\top x$ will never be larger than $b + M$ for some sufficiently large M. Then we add constraint

$$a^\top x \leq b + M(1 - y).$$

The idea is that when $y = 0$, the constraint we impose is $a^\top x \leq b + M$, which is redundant. In most models, a sufficiently large M can be found. However, solver performance is better when M is as small as possible.

4 Transportation with ramp-up costs

Suppose that we have production plants P, each of which can produce p_i quantities of a certain good. We have retailers R, which demand r_j quantities of the good. Routes between plants and retailers are given by a bipartite network $(P \cup R, A)$ where A is the set of arcs connecting plants and retailers. The problem is to transport the goods from plats to retailers in order to meet demands.

- We assume that transporting one unit from plant i to retailer j incurs cost c_{ij}.
- Deciding to produce goods at plat i incurs some ramp-up costs f_i (warming up machines, starting power, etc.).
- On the other hand, if no goods are produced at plant i, then no ramp-up costs are incurred.

Decisions: We use x_{ij} to decide the amount of goods transported from plant i to retailer j. Moreover, we use variable y_i to decide to use plant i or not.

$$y_i = \begin{cases} 1, & \text{if plant } i \text{ produces goods} \\ 0, & \text{otherwise} \end{cases}$$

Here, $x_{ij} \geq 0$ and $y_i \leq 0$ for all i, j.

Objective: We want to minimize the total cost, which consists of the tranportation cost and the ramp-up costs of plants.

$$\sum_{i \in P} f_i y_i + \sum_{(i,j) \in A} c_{ij} x_{ij}.$$

Supply and demand constraints: We know that the production capacity of plant i is p_i. Hence, we impose

$$\sum_{j:(i,j) \in A} x_{ij} \leq p_i, \quad i \in P.$$

Moreover, the demand of retailer j is r_j. Hence,

$$\sum_{i:(i,j) \in A} x_{ij} \geq r_j, \quad j \in R.$$

Implication constraints: We need to consider how y_i and x_{ij} interact. Note that if plant i is not in operation, then it cannot produce any goods. We may model this by the following implication constraints.

$$\begin{aligned} y_i = 0 \quad &\Rightarrow \quad x_{ij} = 0 \\ y_i = 1 \quad &\Rightarrow \quad \text{no constraint imposed} \end{aligned}$$

Following the previous discussion on the big-M technique, we may impose the constraint by

$$x_{ij} \leq My_i, \quad (i,j) \in A.$$

In fact, we may obtain a proper value for the constant M here. Recall that we have supply constraints given by

$$\sum_{j:(i,j)\in A} x_{ij} \leq p_i, \quad i \in P.$$

This implies that $x_{ij} \leq p_i$ in any case. Therefore, we may set $M = p_i$. As a result,

$$x_{ij} \leq p_i y_i, \quad (i,j) \in A.$$

In fact, we may aggregate these constraints for a fixed i to deduce

$$\sum_{j:(i,j)\in A} x_{ij} \leq p_i y_i, \quad i \in P.$$

Therefore, the optimization model for the transportation problem is given by

$$\begin{aligned} \min \quad & \sum_{i\in P} f_i y_i + \sum_{(i,j)\in A} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{i:(i,j)\in A} x_{ij} \geq r_j, \quad j \in R \\ & \sum_{j:(i,j)\in A} x_{ij} \leq p_i y_i, \quad i \in P \\ & x \geq 0, \ y \in \{0,1\}^P. \end{aligned}$$