Lecture 6: Linear programming

Rajat Mittal *
IIT Kanpur

After a long wait, finally, we introduce Linear programming today and see some preliminary examples of it.

1 Resource allocation problem

Suppose we have a company which makes two kinds of laptop, Apple and Dell. Every Apple gives a profit of 10 Rs. and every Dell 5 Rs.. It is clear that to maximize the profit the company should make as many Apple computers as possible (assuming they can sell everything they build). But life is not so simple, every Apple computer takes 20 people to build, on the contrary Dell just takes 13. Similarly, an Apple needs 4 chips, but the Dell needs only 1. At any particular day, the company has at most 95 people and 28 chips for their disposal. How many Apple’s and Dell’s should the company make?

If you are a Steve Jobs fan, the best strategy would be to make only Apple’s and throw out all the other chips and fire the remaining people. But, from the business point of view (and even mathematical) the problem is quite clear,

\[
\begin{align*}
\max & \quad 10x_1 + 5x_2 \\
\text{s.t.} & \quad 20x_1 + 13x_2 \leq 95 \\
& \quad 4x_1 + x_2 \leq 28 \\
& \quad x_1, x_2 \geq 0.
\end{align*}
\]

Here, \(x_1\) is the number of Apple’s and \(x_2\) is the number of Dell’s. In a real scenario, we want these to be integers. Let’s not worry about this constraint yet, later we will see that this constraints that variables should be integer make certain problems really hard.

In any case, the above optimization approach can be generalized to the following resource allocation problem. Suppose, a manufacturing unit wants to produce items \(i = 1, \ldots, n\) using raw materials \(j = 1, \ldots, m\). The cost of raw material \(j\) is \(\gamma_j\) and the price of item \(i\) is \(\rho_i\). There is only \(b_j\) amount of raw material \(j\) available. If a single unit of item \(i\) requires \(a_{ij}\) amount of raw material, the manager’s job is,

\[
\begin{align*}
\max & \quad \sum_i (\rho_i - \sum_j a_{ij}\gamma_j)x_j \\
\text{s.t.} & \quad \forall j \quad \sum_i a_{ij}x_i \leq b_j \\
& \quad \forall i \quad x_i \geq 0.
\end{align*}
\]

Notice that \(\rho_i - \sum_j a_{ij}\gamma_j\) can be thought of as the profit for item \(i\), we call it \(c_i\). Suppose \(c\) is the vector with co-ordinates \(c_i\), \(x\) with co-ordinates \(x_i\) and so on, then

\[
\begin{align*}
\max & \quad c^Tx \\
\text{s.t.} & \quad \forall j \quad a_j^Tx \leq b_j \\
& \quad x \geq 0.
\end{align*}
\]

* Thanks to the book from R.J.Vanderbei, *Linear Programming: Foundations and Extensions*
Let us look at the same resource allocation problem from a pessimist’s point of view. Suppose, he wants to assign some cost $y_j$ to every raw material so that the cost of his inventory is minimized (for budget purposes). Though the catch is, he should be willing to sell the raw material at the same price to some other competitor manufacturing unit.

These constraint imply, his assigned cost should not be smaller than the market price, $y_j \geq \gamma_j$ (else the competitors can directly buy from him instead of market) and also

$$\forall \ i \ \sum_j a_{ij} y_j \geq \rho_i$$

Otherwise, the competitor can buy the raw material from our unit and make the items cheaper than the market price. Hence, the problem becomes,

$$\min \sum_j b_j y_j$$

s.t. $$\forall \ i \ \sum_j a_{ij} y_j \geq \rho_i$$

$$\forall \ j \ \ y_j \geq \gamma_j.$$  

If we make a change of variable here $z_j = y_j - \gamma_j$, the life will be much simpler,

$$\min \sum_j b_j z_j$$

s.t. $$\forall \ i \ \sum_j a_{ij} z_j \geq c_i$$

$$\forall \ j \ \ z_j \geq 0.$$  

You can see that both problems look similar. Definitely both have linear objective and constraint functions. Let’s make this precise.

2 Standard format

A linear program is an optimization problem where both the constraints as well as objective function is linear in the variables to be optimized. Through this definition there can be inequalities, equalities or different signs on the variables. To make the future analysis and description simple, we assume a standard form. All other kind of linear programs can be converted into this standard form. The standard form of a linear program is,

$$\max \ c^T x$$

s.t. $$\begin{array} \leq \end{array} \begin{array} b \end{array}$$

$$x \geq 0.$$  

Where $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $A$ is an $m \times n$ matrix. The constraint $Ax \leq b$ should be interpreted as, every entry of $Ax$ is smaller than or equal to the corresponding entry of $b$. It is almost clear that the resource allocation problem is a linear program (an LP) in standard form. If not show that it is

Exercise 1. Show that the optimization problem can be converted into an LP in standard form. What are $c, b, A$ now?

Exercise 2. Max flow (look at the figure below): Given a graph, start node $s$ and end node $t$, capacities on every edge $c_{u,v}$; find out the maximum flow possible through edges.
3 Converting one LP into another

We have been talking informally about we can convert any LP into standard form. Intuitively it means that both the LP’s are equivalent. What does it mean mathematically? Suppose we are given two LP’s $L_1$ and $L_2$, when are they equivalent?

Two LP’s ($L_1$ and $L_2$) are equivalent iff

- Any optimal solution of $L_1$ can be converted into a feasible solution of $L_2$ with same objective value.
- Any optimal solution of $L_2$ can be converted into a feasible solution of $L_1$ with same objective value.

NOTE: The solutions for two LP’s having the same value can be defined in various ways, e.g., one could be a simple monotone function of another.

For an example, suppose for input we have $x \in \{0,1\}^n$ and there are sets $C_1, C_2, \ldots, C_m \subseteq \{0,1\}^n$.

Consider the LP,

\[
\text{max } \sum x_u x + v_x \\
\text{s.t. } \forall i \in [m] \sum_{x \in C_i} u_x - v_x \leq |C| \\
\forall x u_x, v_x \in \mathbb{R}.
\]

Observe that by change of variable $y_x = u_x + v_x$ and $z_x = u_x - v_x$, the LP becomes

\[
\text{max } \sum x y_x \\
\text{s.t. } \forall i \in [m] \sum_{x \in C_i} z_x \leq |C| \\
\forall x y_x, z_x \in \mathbb{R}.
\]

Now it is clear that value of $z_x$ doesn’t matter (we can set it to zero) and $y_x$ can be raised as high as possible.

Exercise 3. Show that above two LP’s are equivalent. What if in the first LP, we had constraint $\forall x u_x, v_x \geq 0$?
4 Other formats

Let’s talk about how to convert different kinds of linear constraints into the standard form.

- Inequality in the opposite direction: A constraint like $d^T x \geq e$ can be converted to $(−d^T)x \leq (−e)$.
- Equality: An equality can be converted into two inequalities, $d^T x = e \iff d^T x \leq e$ and $d^T x \geq e$.
- Input variable less than equal to zero: Change variable $x_i$ to $−x_i$.
- No constraint on input variable: If $x_i$ is unconstrained, then $x_i = y_i - z_i$, where $y_i, z_i \geq 0$.

Exercise 4. Show that the two LP’s in this case would be equivalent in the sense described above.

- Strict inequalities: Not allowed in LP’s. Instead we solve the approximate version with inequalities.
- We don’t need to consider sup/inf and can only work with max/min. This can be justified using Fourier-Motzkin elimination.