Chapter 1

What is Linear Programming?

An optimization problem usually has three essential ingredients: a variable vector x consisting of a set of unknowns to be determined, an objective function of x to be optimized, and a set of constraints to be satisfied by x.

A *linear program* is an optimization problem where all involved functions are linear in x; in particular, all the constraints are linear inequalities and equalities. *Linear programming* is the subject of studying and solving linear programs.

Linear programming was born during the second World War out of the necessity of solving military logistic problems. It remains one of the most used mathematical techniques in today's modern societies.

1.1 A Toy Problem

A local furniture shop makes chairs and tables. The projected profits for the two products are, respectively, \$20 per chair and \$30 per table. The projected demands for chairs and tables are 400 and 100, respectively. Each chair requires 2 cubic feet of wood while each table requires 4 cubic feet. The shop has a total amount of 1,000 cubic feet of wood in store. How many chairs and tables should the shop make in order to maximize its profit?

Let x_1 be the number of chairs and x_2 the number of tables to be made. There are the two variables, or unknowns, for this problem. The shop wants to maximize its total profit, $20x_1 + 30x_2$, subject to the constraints that (a) the total amount of wood used to make the two products can not exceed the 500 cubic feet available, and (b) the numbers of chairs and tables to be made should not exceed the demands. In additional, we should not forget that the number of chairs and tables made need to be nonnegative. Putting all these together, we have an optimization problem:

$$\begin{array}{ll} \max & 20x_1 + 30x_2 \\ \text{s.t.} & 2x_1 + 4x_2 \leq 1000 \\ & 0 \leq x_1 \leq 400 \\ & 0 \leq x_2 \leq 100 \end{array} \tag{1.1}$$

where $20x_1 + 30x_2$ is the objective function, "s.t." is the shorthand for "subject to" which is followed by constraints of this problem.

This optimization problem is clearly a linear program where all the functions involved, both in the objective and in the constraints, are linear functions of x_1 and x_2.

1.2 From Concrete to Abstract

Let us look at an abstract production model. A company produces n products using m kinds of materials. For the next month, the unit profits for the n products are projected to be, respectively, $c_1, c_2, \cdots, c_n$. The amounts of materials available to the company in the next month are, respectively, $b_1, b_2, \cdots, b_m$. The amount of material i consumed by a unit of product j is given by $a_{ij} \geq 0$ for $i = 1, 2, \cdots, m$ and $j = 1, 2, \cdots, n$ (some a_{ij} could be zero if prodict j does not consume material i). The question facing the company is, given the limited availability of materials, what is the quantity the company should produce in the next month for each product in order to achieve the maximum total profit?

The decision variables are obviously the amounts of production for the n products in the next month. Let us call them $x_1, x_2, \cdots, x_n$. The optimization model is *to maximize the total profit, subject to the material availability constraints for all m materials, and the nonnegativity constraints on the n variables.*

In mathematical terms, the model is the following linear program:

$$
\begin{array}{ll}
\max & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s.t.} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m \\
& x_1, x_2, \cdots, x_n \geq 0
\end{array}
\tag{1.2}
$$

The nonnegativity constraints can be vital, but are often forgotten by beginners. Why is nonnegativity important here? First, in the above context, it does not make sense to expect the company to produce a negative amount of a product. Moreover, if one product, say product k, is not profitable, corresponding to a $c_k < 0$, without the nonnegativity constraints the model would produce a solution $x_k < 0$ and generate a profit $c_kx_k > 0$. In fact, since a negative amount of product k would not consume any material but instead "generate" materials, one could drive the profit to infinity by forcing x_k to go to negative infinity. Hence, the model would be totally wrong had one forgot nonnegativity.

The linear program in (1.2) is tedious to write. One can shorten the expressions using the summation notation. For example, the total profit can be represented by the left-hand side instead of the right-hand side of the following identity

$$
\sum_{i=1}^{n} c_ix_i = c_1x_1 + c_2x_2 + \cdots + c_nx_n.
$$

However, a much more concise way is to use matrices and vectors. If we let $c = (c_1, c_2, \cdots, c_n)^T$ and $x = (x_1, x_2, \cdots, x_n)^T$. Then the total profit becomes c^Tx. In a matrix-vector notation, the linear program (1.2) becomes

$$
\begin{array}{ll}
\max & c^Tx \\
\text{s.t.} & Ax \leq b \\
& x \geq 0
\end{array}
\tag{1.3}
$$

where $A \in \Re^{m\times n}$ and $b \in \Re^m$. The inequalities are always understood as component-wise comparisons.

1.3 A Standard Form

A linear program can have an objective of either minimization or maximization, while its constraints can have any combination of linear inequalities and equalities. It is impractical to study linear programs and to design algorithms for them without introducing a so-called *standard form* – a unifying framework encompassing most, if not all, individual forms of linear programs. Different standard forms exist in the literature that may offer different advantages under different circumstances. Here we will mostly use the following standard form:

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \tag{1.4}$$

where the matrix A is assumed to be $m \times n$ with $m < n$, $b \in \Re^m$ and $c, x \in \Re^n$. The triple (A, b, c) represents problem data that need to be specified, and x is the variable to be determined. In fact, once the size of A is given, the sizes of all the other quantities follow accordingly from the rule of matrix multiplications.

In plain English, a standard linear program is one that is a minimization problem with a set of equality constraints but no inequality constraints except nonnegativity on all variables.

We will always assume that A has full rank, or in other words, the rows of A are linearly independent which ensures that the equations in $Ax = b$ are consistent for any right-hand side $b \in \Re^m$. We make this assumption to simplify the matter without loss of generality, because redundant or inconsistent linear equations can always be detected, and removed if so desired, through standard linear algebra techniques. Moreover, the requirement $m < n$ ensures that in general there are infinite many solutions to the equation $Ax = b$, leaving degrees of freedom for nonnegative and optimal solutions.

A linear program, say LP-A, is said to be equivalent to a linear program LP-B if an optimal solution of LP-A, if it exists, can be obtained from an optimal solution of LP-B through some simple algebraic operations. This equivalence allows one to solve LP-B and obtain a solution to LP-A.

We claim that every linear program is equivalent to a standard-form linear program though a transformation, which usually requires adding extra variables and constraints. Obviously, maximizing a function is equivalent to minimizing the negative of the function. A common trick to transform an inequality $a^T x \leq \beta$ into an equivalent equality is to add a so-called *slack* variable η so that

$$a^T x \le \beta \iff a^T x + \eta = \beta,\ \eta \ge 0.$$

Let us consider the toy problem (1.1). With the addition of slack variables x_3, x_4, x_5 (we can name them anyway we want), we transform the linear program on the left to the one on the right:

$$\begin{array}{rc} \max & 20x_1 + 30x_2 \\ \text{s.t.} & 2x_1 + 4x_2 \le 1000 \\ & x_1 \le 400 \\ & x_2 \le 100 \\ & x_1, x_2 \ge 0 \end{array} \quad \Longrightarrow \quad \begin{array}{rc} \max & 20x_1 + 30x_2 \\ \text{s.t.} & 2x_1 + 4x_2 + x_3 = 1000 \\ & x_1 + x_4 = 400 \\ & x_2 + x_5 = 100 \\ & x_1, x_2, x_3, x_4, x_5 \ge 0 \end{array}$$

After switching to minimization, we turn the linear program on the right to an equivalent linear program of the standard form:

$$\begin{array}{rc} \min & -20x_1 - 30x_2 + 0x_3 + 0x_4 + 0x_5 \\ \text{s.t.} & 2x_1 + 4x_2 + x_3 + 0x_4 + 0x_5 = 1000 \\ & x_1 + 0x_2 + 0x_3 + x_4 + 0x_5 = 400 \\ & 0x_1 + x_2 + 0x_3 + 0x_4 + x_5 = 100 \\ & x_1, x_2, x_3, x_4, x_5 \ge 0 \end{array} \tag{1.5}$$

where $c^T = -(20\ 30\ 0\ 0\ 0)$, b is unchanged and

$$A = \begin{pmatrix} 2 & 4 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

This new coefficient matrix is obtained by padding the 3-by-3 identity matrix to the right of the original coefficient matrix.

Similarly, the general linear program (1.3) can be transformed into

$$\begin{array}{rc} \min & -c^T x \\ \text{s.t.} & Ax + s = b \\ & x, s \ge 0 \end{array} \tag{1.6}$$

where $s \in \Re^m$ is the slack variable vector. This linear program is in the standard form because it is a minimization problem with only equality constraints and nonnegativity on all variables. The variable in (1.6) consists of $x \in \Re^n$ and $s \in \Re^m$, and the new data triple is obtained by the construction

$$A \leftarrow [A\ \ I], \quad b \leftarrow b, \quad c \leftarrow \begin{pmatrix} -c \\ 0 \end{pmatrix},$$

where I is the m-by-m identity matrix and $0 \in \Re^m$.

1.4 Feasibility and Solution Sets

Let us call the following set

$$\mathcal{F} = \{x \in \Re^n : Ax = b,\ x \geq 0\} \subset \Re^n \tag{1.7}$$

the *feasibility set* of the linear program (1.4). Points in the feasibility set are called *feasible points*, out of which we seek an *optimal solution* x^* that minimizes the objective function $c^T x$. With the help of the feasibility set notation, we can write our standard linear program into a concise form

$$\min\{c^T x : x \in \mathcal{F}\}. \tag{1.8}$$

A *polyhedron* in $\Re^n$ is a subset of $\Re^n$ defined by points satisfying a collection of linear inequalities and/or equalities. For example, a *hyper-plane*

$$\{x \in \Re^n : a^T x = \beta\}$$

is a polyhedron where $a \in \Re^n$ and $\beta \in \Re$, and a *half-space*

$$\{x \in \Re^n : a^T x \leq \beta\}$$

is a polyhedron as well. In geometric terms, a polyhedron is nothing but the intersection of a collection of hyper-planes and half-spaces. In particular, the empty set and a singleton set (that contains only a single point) are polyhedra.

Clearly, the feasibility set $\mathcal{F}$ in (1.7) for linear program (1.4) is a polyhedron in $\Re^n$. It may be empty if some constraints are contradictory (say, $x_1 \leq -1$ and $x_1 \geq 1$), or it may be an unbounded set, say,

$$\mathcal{F} = \{x \in \Re^2 : x_1 - x_2 = 0, x \geq 0\} \subset \Re^2, \tag{1.9}$$

which is the half diagonal-line emitting from the origin towards infinity. Most of times, $\mathcal{F}$ will be a bounded, nonempty set. A bounded polyhedron is also called a *polytope*.

The *optimal solution set*, or just *solution set* for short, of a linear program consists of all feasible points that optimizes its objective function. In particular, we denote the solution set of the standard linear program (1.4) or (1.8) as $\mathcal{S}$. Since $\mathcal{S} \subset \mathcal{F}$, $\mathcal{S}$ is empty whenever $\mathcal{F}$ is empty. However, $\mathcal{S}$ can be empty even if $\mathcal{F}$ is not.

The *purpose of a linear programming algorithm* is to determine where $\mathcal{S}$ is empty or not and, in the latter case, to find a member x^* of $\mathcal{S}$. In case such an x^* exists, we can write $\mathcal{S} = \{x \in \mathcal{F} : c^T x = c^T x^*\}$ or

$$\mathcal{S} = \{x \in \Re^n : c^T x = c^T x^*\} \cap \mathcal{F}. \tag{1.10}$$

That is, $\mathcal{S}$ is the intersection of a hyper-plane with the polyhedron $\mathcal{F}$. Hence, $\mathcal{S}$ itself is a polyhedron. If the intersection is a singleton $\mathcal{S} = \{x^*\}$, then x^* is the unique optimal solution; otherwise, there must exist infinitely many optimal solutions to the linear program.

1.5 Three Possibilities

A linear program is *infeasible* if its feasibility set is empty; otherwise, it is *feasible*.

A linear program is *unbounded* if it is feasible but its objective function can be made arbitrarily "good". For example, if a linear program is a minimization problem and unbounded, then its objective value can be made arbitrarily small while maintaining feasibility. In other words, we can drive the objective value to negative infinity within the feasibility set. The situation is similar for a unbounded maximization problem where we can drive the objective value to positive infinity. Clearly, a linear program is unbounded only if its feasibility set is a unbounded set. However, a unbounded feasibility set does not necessarily imply that the linear program itself is unbounded.

To make it clear, let us formally define the term *unbounded* for a set and for a linear program. We say a set $\mathcal{F}$ is unbounded if there exists a sequence $\{x^k\} \subset \mathcal{F}$ such that $\|x^k\| \to \infty$ as $k \to \infty$. On the other hand, we say a linear program $\min\{c^T x : x \in \mathcal{F}\}$ is unbounded if there exists a sequence $\{x^k\} \subset \mathcal{F}$ such that $c^T x^k \to \infty$ as $k \to \infty$. Hence, for a linear program the term *unbounded* means *objective unbounded*.

When the feasibility set $\mathcal{F}$ is unbounded, whether or not the corresponding linear program is unbounded depends entirely on the objective function. For example, consider $\mathcal{F}$ given by (1.9). The linear program

$$\max\{x_1 + x_2 : (x_1, x_2) \in \mathcal{F}\} = \max\{2x_1 : x_1 \geq 0\}$$

is unbounded. However, when the objective is changed to minimization instead, the resulting linear program has an optimal solution at the origin.

If a linear program is feasible but not (objective) unbounded, then it must achieve a finite optimal value within its feasibility set; in other words, it has an optimal solution $x^* \in \mathcal{S} \subset \mathcal{F}$.

To sum up, for any given linear program there are three possibilities:

1. The linear program is infeasible, i.e., $\mathcal{F} = \emptyset$. In this case, $\mathcal{S} = \emptyset$.

2. The linear program is feasible but (objective) unbounded. In this case, $\mathcal{F}$ must be an unbounded set and $\mathcal{S} = \emptyset$.

3. The linear program is feasible and has an optimal solution $x^* \in \mathcal{S} \subset \mathcal{F}$. In this case, the feasibility set $\mathcal{F}$ can be either unbounded or bounded.

These three possibilities imply that if $\mathcal{F}$ is both feasible and bounded, then the corresponding linear program must have an optimal solution $x^* \in \mathcal{S} \subset \mathcal{F}$, regardless of what objective it has.