Linear Programming

For more info consider taking Math 5770, Introduction to Optimization.

The phrase “Linear Programming Problem” refers to a linearly constrained optimization problem with a linear objective function. Thus “programming” means minimization or maximization. It has nothing to do with writing a computer program.

We begin with some examples.

Example 1: A very simple problem. You have a small business that makes tables and chairs. You have limited resources, and you make a certain profit on each item, according to the following Table:

<table>
<thead>
<tr>
<th>Labor</th>
<th>Material</th>
<th>Profit</th>
<th>number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Chair</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Available</td>
<td>8</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

You are planning to produce $x$ tables and $y$ chairs and you ask what choice of $x$ and $y$ will maximize your profit. We are not going to take apart existing furniture, we cannot exceed our resources, and we want to maximize our profit. This gives rise to the linear programming problem

$$f(x, y) = 3x + 2y = \max$$

subject to the inequalities

$$x + y \leq 8,$$

$$2x + y \leq 12,$$

and the sign restrictions

$$x, y \geq 0.$$
Figure 1. Furniture Problem.

Figure 1 illustrates the problem. The inequalities define the feasible region of points \((x, y)\) that satisfy the constraints. The Profit maximized at the point \((4, 4)\). Thus the best plan is to produce four tables and four chairs per day.

\[
3x + 2y = z = \text{profit} \quad = \max
\]
\[
y = \frac{z - 3x}{2}
\]

That town is full of bachelors who never invite anybody and who will cheerfully buy one table and just one chair.
Example 2: The Transportation Problem. You want to organize the optimal distribution of a commodity (like coal, perhaps). There are \( m \) sources (like coal mines) \( S_1 \) through \( S_m \), and \( n \) consumers (like power plants) \( C_1 \) through \( C_n \). Each consumer \( C_i \) has a demand \( c_i \), and each source \( S_i \) has a capacity \( s_i \). You need to specify how much of the commodity, say \( x_{ij} \) units, to transport from source \( S_i \) to consumer \( C_i \). The cost of transporting one unit of the commodity from source \( S_i \) to consumer \( C_j \) is \( \alpha_{ij} \). You want to minimize the cost subject to the following constraints: No source capacity is exceeded, all consumer demand is met, and you will not ship from a user to a source. This gives rise to the problem:

\[
\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_{ij} x_{ij} = \min
\]

subject to:

\[
\sum_{j=1}^{m} x_{ij} \geq c_i, \quad i = 1 \ldots, n \quad \text{(demand is met)}
\]

\[
\sum_{i=1}^{n} x_{ij} \leq s_j, \quad j = 1 \ldots, m \quad \text{(capacity is not exceeded)}
\]

\[
x_{ij} \geq 0, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m \quad \text{(no backward movement)}
\] (3)

Note that this is not just any old transportation problem, it is universally known as the transportation problem.
Example 3: $L_\infty$ Approximation. Let $A$ be an $m \times n$ matrix and $b \in \mathbb{R}^m$ a given vector. The $L_\infty$ problem

$$\|Ax - b\|_\infty = \max_{i=1,...,m} \left| \sum_{j=1}^{n} a_{ij}x_j - b_i \right| = \min$$

(4)

can be written as the linear programming problem:

$$\gamma = \min$$

subject to

$$Ax - \gamma e \leq b,$$

$$-Ax - \gamma e \leq -b$$

and $e = [1, 1, \ldots, 1]^T \in \mathbb{R}^m$

(5)

Exercise 4. Show why this works.
Example 5 L₁ Approximation. Let A and b be as before. The L₁ problem

\[ \|Ax - b\|_1 = \sum_{i=1}^{m} \left| \sum_{j=1}^{n} a_{ij} x_j - b_i \right| = \min \] (6)

can also be written as a linear programming problem:

\[ \sum_{i=1}^{m} r_i^+ + \sum_{i=1}^{m} r_i^- = \min \]

subject to

\[ Ax - r^+ + r^- = b, \]

and \( r^+, r^- \geq 0 \)

(Here \( r^+ \) and \( r^- \) are vectors in \( \mathbb{R}^m \) such that \( Ax - b = r^+ - r^- \).)

Exercise 6. Again, show why this works.

\[ r = Ax - b \quad \sum r_i = \min \]

\[ r = r^+ - r^- \quad r^+, r^- \geq 0 \]

\[ \begin{bmatrix} -3 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} \]

\[ \|r\| \leq \sum_{i=1}^{m} r_i^+ + r_i^- \]

\[ r = Ax - b \quad r_i^+, r_i^- > 0 \]

\[ r - Ax = -b \]

\[ \sum r_i^+ + r_i^- = \min \]
These examples illustrate the following points:

• The objective function is linear.

• Constrains may occur as equality or inequality con-
  straints, but are linear.

• The problem may call for minimization or maximiza-
  tion.

• There may or may not be sign restrictions of the form
  $x \geq 0$. These are of course a form of inequality con-
  straints, but it is worthwhile to treat them separately.

To discuss methods of solving linear programming prob-
lems it is useful to settle on a particular kind of prob-
lem, called a \textit{canonical} problem, and convert any other
LP problem into that form.

We will consider the following

\textbf{Canonical Problem 7.} Let $A \in \mathbb{R}^{m \times n}$, $m \leq n$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. Find $x \in \mathbb{R}^n$ such that

$$c^T x = \min \text{ subject to } Ax = b \geq 0 \text{ and } x \geq 0. \quad (8)$$

If we are given equality constraints $Ax = b$, but some components of $b$ are negative we simply multiply the cor-
responding equation with -1.

Before discussing the \textit{simplex method} for solving the
canonical problem let’s look at ways of converting from
other forms of LP problems:

Maximization versus Minimization. To convert
from one to the other simply replace the objective function
$f(x) = c^T x$ by its negative.

Replacing Equalities with Inequalities. Replace
$Ax = b$ with the two inequalities $Ax \leq b$ and $Ax \geq b$.

Replacing Inequalities with Equalities. To replace
the inequalities $Ax \leq b \geq 0$ with equalities introduce the
\textit{slack variables} $y \in \mathbb{R}^m$ and use the conditions $Ax + y = b$
where $y \geq 0$. 
Getting rid of Sign Restrictions. To remove sign restrictions simply include them with the inequalities.

Introducing Sign Restrictions. If $x$ is not sign restricted write it at $x = x^+ - x^-$ where $x^+ \geq 0$ and $x^- \geq 0$.

Notice that none of the above techniques reduce the size of the problem and most increase it by increasing the number of variables or the number of constraints.

Exercise 8. Convert the problems given in the above examples to canonical form.
Algorithms

There are two groups of algorithms for the solution of a linear programming problems. The first is the Simplex Method which was invented by George Dantzig in the late 1940s. The second is a set of algorithms called interior point methods which are based on ideas first applied to nonlinear programming problems. The original and best known interior point method is Karmarkar’s Algorithm first presented in 1984. Karmarkar worked for Bell Labs at the time, and his algorithm caused excitement for two reasons. The first is that the theoretical worst case speed of the algorithm is polynomial. The Simplex method, in certain worst cases, can require a number of steps that grows exponentially. The second reason for the excitement about Karmarkar’s algorithm was that Bell Labs tried to patent it.

We could easily spend a semester on interior point methods, whereas the Simplex method can be adequately described in just one class meeting, so we ill focus on it. However, there is a beautiful paper by Margaret Wright, one of the world’s leading optimization researchers, that discusses interior point methods and their historical role in great depth:

The Simplex Method

The following description is from the excellent, if somewhat dated, book


We consider the canonical problem 7. Let’s suppose that the rank of $A$ is $m$. (If it isn’t, and there is in fact a solution of $Ax = b$ then we remove redundant equations and reduce the value of $m$.)

We call $x$ a **basic solution** if $Ax = b$ and at most $m$ components of $x$ are non-zero. We call $x$ **feasible** if the constraints $Ax = b$ and $x \geq 0$ are satisfied.

The **feasible region** is the set of all feasible solutions $x$. **Basic feasible solutions**, i.e., basic solutions that are also feasible, (bfs’s) are vertices of the feasible region. One can show that the solution of the canonical problem is assumed at a bfs. The simplex method moves from one bfs to a neighboring one, at each step improving, or at least not reducing, the value of the objective function, and eventually ending up at the optimal solution.

An initial bfs is easy to obtain if we start with the inequality constraints

$$Ax \leq b \geq 0.$$ \hspace{1cm} (9)

We introduce slack variables $y$ with the system $y + Ax = b$ and now have equality constraints

$$[I \hspace{0.5cm} A] \begin{bmatrix} y \\ x \end{bmatrix} = b$$ \hspace{1cm} (10)

with an initial bfs $s \begin{bmatrix} b \\ 0 \end{bmatrix}$.

If we do start with equality constraints an initial bfs can be found by solving the **auxiliary problem**
\[ \sum_{i=1}^{m} \gamma_i = [e \ 0] \begin{bmatrix} y \\ x \end{bmatrix} = \min \ \text{subject to} \quad y + Ax = b, \quad x \geq 0, \quad y \geq 0 \quad (11) \]

Here \( e \in \mathbb{R}^m \) is the vector with all 1’s, as before, and the obvious initial bfs is \( \begin{bmatrix} b \\ 0 \end{bmatrix} \). If the optimum value of \( \gamma \) is \( \gamma = 0 \) then \( y = 0 \) and we have a bfs for the original problem. Otherwise there is no feasible point for the original problem.

In general, suppose we have solved the system \( Ax = b \) for \( m \) of the variables. For the sake of notation, suppose for the moment that these are the first \( m \). Then we have

\[
\begin{align*}
x_1 &= \bar{b}_1 - \bar{a}_{1,m+1}x_{m+1} - \bar{a}_{1,m+2}x_{m+2} - \ldots - \bar{a}_{1,n}x_n \geq 0 \\
x_2 &= \bar{b}_2 - \bar{a}_{2,m+1}x_{m+1} - \bar{a}_{2,m+2}x_{m+2} - \ldots - \bar{a}_{2,n}x_n \geq 0 \\
&\vdots \\
x_i &= \bar{b}_i - \bar{a}_{i,m+1}x_{m+1} - \bar{a}_{i,m+2}x_{m+2} - \ldots - \bar{a}_{i,n}x_n \geq 0 \\
&\vdots \\
x_m &= \bar{b}_m - \bar{a}_{m,m+1}x_{m+1} - \bar{a}_{m,m+2}x_{m+2} - \ldots - \bar{a}_{mn}x_n \geq 0 \\
\end{align*}
\]

(12)

In block form this can be written as

\[
\begin{bmatrix} I & \bar{A} \end{bmatrix} x = \bar{b}. \quad (13)
\]

which is the same form as (10).

The variables corresponding to the columns of the identity matrix are called the \textbf{basic variables}, and the others are the \textbf{nonbasic} or \textbf{free variables}. They are called free because by choosing suitable values for the basic variables we can assign any values at all to the free variables.

The cost function can now be expressed in terms of the free variables as

\[
f(x) = c^T x = z_0 + \sum_{i=m+1}^{n} \bar{c}_i x_i. \quad (14)
\]

The \( \bar{c}_i \) are called the \textbf{reduced cost coefficients}.
Exercise 9. Find explicit expressions for $z_0$ and the reduced cost coefficients in terms of the $\tilde{b}_i$ and $\tilde{a}_{ij}$.

\[
A_x = \begin{bmatrix} A_{mm} & A_{m,n-m} \end{bmatrix} \begin{bmatrix} x_m \\ x_{n-m} \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}
\]

\[
A_x = A_{mm}x_m + A_{m,n-m}x_{n-m} = b \quad \text{for} \quad A_{mm}^{-1}
\]

\[
x_m = -A_{mm}^{-1}A_{m,n-m}x_{n-m} + A_{mm}^{-1}b
\]

\[
\tilde{A}_x = \begin{bmatrix} I & \tilde{A}_{m,n-m} \end{bmatrix}
\]

\[
\tilde{c}^T x = \sum_{i=1}^{n} c_i x_i
\]

\[
= \sum_{i=1}^{m} c_i x_i + \sum_{i=m+1}^{n} c_i x_i
\]

\[
= \sum_{i=1}^{m} c_i \left( \tilde{b}_i - \sum_{j=m+1}^{n} \tilde{a}_{ij} x_j \right)
\]
\[
\begin{align*}
\sum_{i=1}^{m} c_i \bar{v}_i &= \sum_{i=1}^{m} c_i \sum_{j=m+1}^{n} \bar{a}_{ij} x_j \\
= Z_0 + \sum_{j=m+1}^{n} \left( \sum_{i=1}^{m} c_i \bar{a}_{ij} \right) x_j \\
= Z_0 + \sum_{j=m+1}^{n} c_j x_j
\end{align*}
\]

\[
\begin{bmatrix} I & \bar{A} \end{bmatrix} \begin{bmatrix} x_n \\ x_{m-n} \end{bmatrix} = \bar{b}
\]

\[
\text{Number of \textit{v}tcs} = \binom{n}{m} = \frac{n!}{m! (n-m) !}
\]

\[N = 2m \quad \frac{(2m)!}{(m!)^2}\]
We now consider three cases:

**Case 1.** The reduced cost coefficients are all non-negative, 
\[ \bar{c}_k \geq 0, \quad k = m + 1, \ldots, n \] (15)

Then the value of the objective function can only be increased, we have the smallest possible value of the objective function, and we have found the optimum solution.

So suppose some reduced cost coefficient is negative, say
\[ \bar{c}_k < 0 \] (16)
for some specific \( k \). Then we can reduce the value of the objective function by assigning a positive value to the free variable \( x_k \). Of course, in doing so we must not make any of the basic variables negative.

**Case 2.** Suppose
\[ \bar{c}_k < 0 \quad \text{and} \quad \bar{a}_{ik} \leq 0 \quad \text{for} \quad i = 1, \ldots, m. \] (17)

In this case we can make the value of the objective function arbitrarily small without making any free variable negative. The objective function is unbounded below, and there is no solution.

So suppose that at least one of the \( \bar{a}_{ik} \) is positive. Then we pick the free variable \( x_k \) as large as possible. This will cause one of the basic variables to become zero.

To make \( x_i \) zero we solve
\[ x_i = 0 = \bar{b}_i - \bar{a}_{ik} x_k \] (18)
which would imply
\[ x_k = \frac{\bar{b}_i}{\bar{a}_{ik}}. \] (19)

This gives
Case 3. Suppose
\[ \bar{c}_k < 0 \quad \text{and} \quad \bar{a}_{ik} > 0 \quad \text{for some} \quad i. \] (20)

We choose
\[ x_k = \min_{\bar{a}_{pk} > 0} \frac{\bar{b}_p}{\bar{a}_{pk}} = \frac{\bar{b}_q}{\bar{a}_{qk}}. \] (21)

We then interchange the roles of \( k \) and \( q \). \( x_k \) becomes a basic variable, and \( x_q \) becomes a free variable. We update the reduced cost coefficients, and the entries of \( \bar{A} \), and repeat the process.

Of course, we don’t interchange columns, we just keep track of which variables are basic, and which are free.

We can change the matrix \([I, \bar{A}]\) by Gaussian Elimination, eliminating in the \( k \)-th column, using the \( i \)-th row as the pivot row.

The whole process can be organized in an array, which in this context is called a tableau. There are many descriptions of this process. One description can be found in the before mentioned book by Ralston and Rabinowitz.

**Exercise 10.** Give explicit formulas for the change in \( z_0 \), the reduced cost coefficients, and the entries of \( A \) when interchanging the variables \( x_i \) and \( x_k \).

**Exercise 11.** It is possible to construct linear programming problems where the Simplex Method keeps cycling through a set of basic feasible solutions without decreasing the non-optimal value of the objective function. Think about how this is possible, and what you might do about it.