What is a Linear Programming problem? A few examples we’ll help us to get an idea.

**Example 1:** Raw material problem

Let say that we have a range of raw materials and we have enumerated them as \{1,2,\ldots,m\}

We denote by \(i\) each one of them.

The products, let say \{1,2,\ldots,n\}, are denoted by \(j\).

We are forming a table as the following one:

\[
\begin{array}{cccc}
1 & 2 & \ldots & j & \ldots & n \\
1 & & & & & \\
2 & & & & & \\
\vdots & & & \vdots & \ddots & \vdots \\
m & & & & & \\
\end{array}
\]

\(a_{ij}\) Where \(a_{ij}\) is the amount of raw material \(i\) used for the production of one unit of product \(j\).

We have the following assumptions:

1. The unit price of raw material \(i\) is \(p_i\)
2. The market price of product \(j\) is \(\sigma_j\)
3. Our actions are not able to alter neither the price \(p\) for each of the raw materials nor the price \(\sigma\) for any of the products.
4. The company is able to sell all of its products.

The question here is "how many units of product \(j\) should the company produce in aim to maximize its profits?"

For each unit of product \(j\) the associated cost is: \(a_{ij}p_i + \ldots + a_{mj}p_m = \sum_{i=1}^{m} a_{ij}p_i\)

and the net revenue (\(C_j\)) is: \(C_j = \sigma_j - \sum_{i=1}^{m} a_{ij}p_i\)

\[
\sum_{j=1}^{n} c_{ij}x_j
\]
The net revenue of \( x_i \) units of product \( j \) is:

\[ \text{The goal is to determine the values } x_j \text{ which maximize the total net revenue: } \sum_{j=1}^{n} c_j x_j \]

Constraints:
For the raw material: the maximum of raw material \( i \) that we can use is \( b_i \)
So, \( \sum_{j=1}^{n} a_{ij} x_j \leq b_i \) (we have \( m \) constraints)
we require also: \( x_j \geq 0 \) (we have \( n \) constraints)

Example 2:

Referring to the 1\textsuperscript{st} example, now we want to minimize the inventory cost. Since we do not want to incur opportunity cost, we try to secure such a price, which covers the opportunity cost and doesn't allow competitors to benefit from this price.
Let's denote such a price with \( w_i \) for raw material \( i \).
The total cost (including opportunity cost) now is: \( \sum_{i} b_i w_i \)
and our goal is to minimize this cost.
The constraints in this problem are:
Constraint 1:
\( w_i \geq p_i \quad i=1,2,\ldots,m \)
and Constraint 2
\( \sum_{i=1}^{m} w_i a_{ij} \geq c_j \quad j = 1,2,\ldots,n \)

Example 3:

Another famous example is the **Diet Problem**

Here, we denote nutrients: \{1,2,\ldots,m\}
and the different kinds of food: \{1,2,\ldots,n\}. We are going to use the notation: \( i \) for each of the nutrients, \( j \) for each kind of food and the notation \( c_j \) for the cost for each \( j \) kind of food.

We construct a table similar to this in problem 1:

<table>
<thead>
<tr>
<th>N</th>
<th>Food</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>t</td>
</tr>
<tr>
<td>r</td>
<td>i</td>
</tr>
<tr>
<td>e</td>
<td>-</td>
</tr>
</tbody>
</table>
The amount of nutrient i per unit of product j

The goal here is to minimize the cost of x units of food:  \[
\min \sum_{j=1}^{n} c_j x_j
\]

The constraint in this problem is:  \[
\sum_{j=1}^{n} a_{ij} x_j \geq b_i
\]
for each i, where \( b_i \) is the minimum amount of nutrient aᵢ.

**Example 4:** The transportation problem

In this problem we have m factories and n warehouses. A picture of the structure of this problem is as following:

\[
\begin{array}{ccc}
\text{Factories} & \rightarrow & \text{Warehouses} \\
1 & \rightarrow & 1 \\
2 & \rightarrow & 2 \\
\vdots & \vdots & \vdots \\
m & \rightarrow & n \\
\end{array}
\]

\( a_i \): amount of product i  \hspace{1cm} \( b_j \): demand of product j

The cost of sending a product from factory i to warehouse j is:  \( c_{ij} \) / unit

and the number of units is: \( x_{ij} \)

The problem here is how we minimize the quantity:  \[
\sum_{i} \sum_{j} c_{ij} x_{ij}
\]

Which is the total cost of products shipped, subject to the following constraints:

\[
\sum_{j=1}^{n} x_{ij} = a_i
\]
1. 

2. \[
\sum_{i=1}^{m} x_{ij} = b_i \quad x_{ij} \geq 0
\]

**Example 5:** minimum cost flow problem

![Network Diagram]

This is a network for the flow

Let's say that we have n-nodes \{1,2,…,n\}

and m arcs which we denote as ij (i is the beginning arc and j the ending arc)

The cost of transferring the product \(x_{ij}\) is \(c_{ij}\)

We want to minimize the total cost: \[
\sum_{j \in A} c_{ij}x_{ij}
\]

where \(A=\{1,2,…,8,9\}\) the set of the arcs

The amount of things coming to the \(i_{th}\) unit is:

\[
\sum_{j} x_{ji} - \sum_{k} x_{ik} = b_i \quad (k: \text{outflow})
\]

We determine \(b\) as follow:

<table>
<thead>
<tr>
<th>(b_i)</th>
<th>(b_i) related to</th>
<th>supplier node</th>
<th>transshipment node</th>
<th>demand node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bi &gt; 0</td>
<td>bi = 0</td>
<td>bi &lt; 0</td>
<td></td>
</tr>
</tbody>
</table>

The problem is subject to the constraint:
In linear programming we have 2 kinds of problems:

1. The standard problem:
Problems of this kind usually ask from us to minimize the quantity \( \sum c_i x_i \)
and they are usually subject to constraints:

\[
\sum a_{ij} x_i = b_i \quad i=1,2,\ldots,n \\
x_j \geq 0
\]

2. Canonical form:
Problems of this kind usually ask from us to maximize the quantity \( \sum w_j x_j \)
and they are usually subject to constraints:

\[
\sum a_{ij} x_i \leq b_i \\
x_j \geq 0
\]

The geometric solution of the linear programming problem with two variables is:
We form inequalities from the constraints and we draw the corresponding lines. We discard every time the points which don't satisfy the inequality (one of the two half-planes) and finally we have an area within all the points satisfy all the constraints (usually the points along the borders of this area satisfy the inequalities also). Finally, with the use of the vector \((c_1, c_2)\) we move the line \(c_1x_1 + c_2x_2\) (objective function) to meet the point or set of points which maximize the value of the objective function. This intersection represents the optimal solution.