CHAPTER 3
FORMULATION OF LINEAR PROBLEMS

- How does a linear programming model look
- How to transform any linear problem into a standard form
- How to formulate a linear programming model

Understanding the structure of a linear programming model

Linear programming models are the humblest of all optimization problems, they are simple and solution methods tend to be highly efficient. This is the power of linear programming, when we are targeting size and we are interested in optimal solutions most likely a linear programming model would be a good option (at least from the computational perspective).

The basic linear programming model Mathematically, a general linear programming model is one in which both the objective function and the constraints (equality and inequality) are linear (actually, the correct term is affine). Note that in linear programs all variables are continuous.

Consider the following toy problem:

$$\max(x_1 + x_2)$$
$$s.t. \quad \begin{aligned} x_1 + x_2 &\leq 2 \\ x_1 + x_3 &= 2 \\ -x_1 - x_4 &\leq -5 \end{aligned}$$

This problem can be represented as follows:

$$\min F(x) = c^T x$$
$$s.t \begin{aligned} h(x) = A'x = b \\ g(x) = Ax \leq b' \end{aligned}$$
$$x \in R^n$$

Where:

$$c = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, A' = \begin{bmatrix} -1 & 0 & 1 & 0 \end{bmatrix}, A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & -1 \end{bmatrix}, b = [2], b' = \begin{bmatrix} 2 \\ -5 \end{bmatrix}$$

The canonical form for a linear programming model While the previous form is mathematically correct very often we want to refer to a canonical way of writing the problem. The canonical form looks as follows:

$$\min F(x) = c^T x$$
$$s.t. Ax \leq b$$
$$x \geq 0$$

What is important to note is that every linear programming model can be transformed into a canonical form problem by applying a set of tricks:

- Every maximization problem can be transformed in a minimization problem by inverting the sign of the objective function.

$$\min\big(F(x)\big) = \max\big(-F(x)\big)$$
$$\text{argmin}\big(F(x)\big) = \text{argmin}\big(F(x)\big)$$

- Every equality constraint can be broken down into two inequality constraints:

$$h(x) = 0$$
$$h(x) \leq 0$$
$$h(x) \geq 0$$

- The sign of an inequality can be changed by multiplying it by minus one

$$g(x) \geq 0 \leftrightarrow -g(x) \leq 0$$

- If a variable is unbounded from below, it is always possible to replace this variable for the difference of two positive variables such that.

$$x = x^+ - x^-$$
$$x^+ \geq 0, x^- \geq 0$$

Example: consider the following optimization problem and transform it into the standard form.

$$\max(x_1 + x_2)$$
$$s.t. \begin{array}{l} x_1 + x_2 \leq 2 \\ x_1 + x_3 = 2 \\ x_1 + x_4 \geq 5 \end{array}$$

The first thing that we are going to do is to transform this into a minimization problem such that we get:

$$\min(-x_1 - x_2)$$
$$s.t. \begin{array}{l} x_1 + x_2 \leq 2 \\ x_1 + x_3 = 2 \\ x_1 + x_4 \geq 5 \end{array}$$

Now we are going to transform the equality equation into two inequalities

$$\min(-x_1 - x_2)$$
$$s.t. \begin{array}{l} x_1 + x_2 \leq 2 \\ x_1 + x_3 \leq 2 \\ x_1 + x_3 \geq 2 \\ x_1 + x_4 \geq 5 \end{array}$$

Now, we are going to flip the sign of inequalities such that all of them are of the type lower than:

$$\min(-x_1 - x_2)$$

$$s.t. \begin{array}{r} x_1 + x_2 \leq 2 \\ x_1 + x_3 \leq 2 \\ -x_1 - x_3 \leq -2 \\ -x_1 - x_4 \leq -5 \end{array}$$

Finally, we replace each of the variables unbounded from below (i.e., variables that do not have a lower bound) by two bounded variables. Note that this increases the total number of variables in the problem, but the optimal solution still is the same, it is just a trick to get the problem in the right format. To do that we define the following:

$$\begin{gathered} x_1 = x_1^+ - x_1^- \\ x_2 = x_2^+ - x_2^- \\ x_3 = x_3^+ - x_3^- \\ x_4 = x_4^+ - x_4^- \\ x_i^+ \geq 0, x_i^- \geq 0, i = \{1,2,3,4\} \end{gathered}$$

Using these results we obtain the final reformulation in standard form

$$\min(-(x_1^+ - x_1^-) - (x_2^+ - x_2^-))$$

$$s.t. \begin{array}{r} (x_1^+ - x_1^-) + (x_2^+ - x_2^-) \leq 2 \\ (x_1^+ - x_1^-) + (x_3^+ - x_3^-) \leq 2 \\ -(x_1^+ - x_1^-) - (x_3^+ - x_3^-) \leq -2 \\ -(x_1^+ - x_1^-) - (x_4^+ - x_4^-) \leq -5 \end{array}$$

$$x_i^+ \geq 0, x_i^- \geq 0, i = \{1,2,3,4\}$$

The standard form for a linear programming model. This form of writing a linear programming model is useful when the simplex algorithm is implemented. The standard form is characterized by the following features:

- There are only equalities
- All variables are bounded from below
- The right-hand side of all equalities is positive
- In some references it is said that a maximization problem is required (some sources are more relaxed with respect to this requirement.

$$\begin{gathered} \min F(x) = c^T x \\ s.t. Ax = b \\ x \geq 0 \end{gathered}$$

To transform a LP into the standard form we can use the same tricks that we describe before to obtain the canonical form, additionally, we are going to need the following trick:

- An inequality can be transformed into an equality by adding slack/surplus variables, this is always done after the right-hand side has been fixed to a positive sign.

$$\begin{gathered} f(x) \leq b \leftrightarrow f(x) + slack = b \\ g(x) \geq b \leftrightarrow f(x) - surplus = b \end{gathered}$$

Example: Let's consider the following example:

$$\min(3x_1 + 2x_2 - x_3 + x_4)$$
$$x_1 + 2x_2 + x_3 - x_4 \leq 5$$
$$-2x_1 - 2x_4 + x_3 + x_4 \leq -1$$
$$x_1 \geq 0, x_2 \leq 0$$

Let's start by transforming it into a maximization problem

$$\max -(3x_1 + 2x_2 - x_3 + x_4)$$
$$x_1 + 2x_2 + x_3 - x_4 \leq 5$$
$$-2x_1 - 2x_4 + x_3 + x_4 \leq -1$$
$$x_1 \geq 0, x_2 \leq 0$$

Now let's transform the second inequality into an equality, since this is a less than inequality we will add a slack variable:

$$\max -(3x_1 + 2x_2 - x_3 + x_4)$$
$$x_1 + 2x_2 + x_3 - x_4 + s_1 = 5$$
$$-2x_1 - 2x_4 + x_3 + x_4 \leq -1$$
$$x_1 \geq 0, x_2 \leq 0$$

Now, let's transform the second inequality into an equality, but before doing so we need to change the sign of the right-hand side to a positive value:

$$\max -(3x_1 + 2x_2 - x_3 + x_4)$$
$$x_1 + 2x_2 + x_3 - x_4 + s_1 = 5$$
$$2x_1 + 2x_4 - x_3 - x_4 \geq 1$$
$$x_1 \geq 0, x_2 \leq 0, s_1 \geq 0$$

Now, we can add a surplus variable as follows:

$$\max -(3x_1 + 2x_2 - x_3 + x_4)$$
$$x_1 + 2x_2 + x_3 - x_4 + s_1 = 5$$
$$2x_1 + 2x_4 - x_3 - x_4 - s_2 = 1$$
$$x_1 \geq 0, x_2 \leq 0, s_1 \geq 0, s_2 \geq 0$$

The final thing that we need to do is to ensure that all variables are positive, this implies doing something about $x_2, x_3, and\ x_4$. We can do that by introducing the following new variables:

$$x_2 = (x_2^+ - x_2^-)$$
$$x_3 = (x_3^+ - x_3^-)$$
$$x_4 = (x_4^+ - x_4^-)$$

To obtain

$$\max -(3x_1 + 2(x_2^+ - x_2^-) - (x_3^+ - x_3^-) + (x_4^+ - x_4^-))$$

$$x_1 + 2(x_2^+ - x_2^-) + (x_3^+ - x_3^-) - (x_4^+ - x_4^-) + s_1 = 5$$
$$2x_1 + 2x_4 - (x_3^+ - x_3^-) - (x_4^+ - x_4^-) - s_2 = 1$$
$$(x_2^+ - x_2^-) \leq 0$$
$$x_1, x_2^+, x_2^-, x_3^+, x_3^-, x_4^+, x_4^-, s_1, s_2$$

Tricks to linearize terms that are usually non-linear

In many instances, when we formulate mathematical models, we find terms that are nonlinear, however, in some occasions, it is possible to reformulate these terms as linear models. A few examples of these reformulations are presented below:

Tricks associated with the objective function

- *A min-max objective* can be transformed into a linear programming model by using the following approach

$$\min\left(\max_i\left(\sum_j a_{ij}x_j\right)\right)$$

This can be converted into a conventional linear programming form by introducing a variable z to represent the above objective. Then we can express the model as follows

$$\min(z)$$
$$s.t. \sum_j a_{ij}x_j - z \leq 0, \forall i$$

A numerical example that illustrates this concept is shown below:

$$\min(max(a_{11}x_1 + \cdots a_{1n}x_n, \dots, a_{m1}x_1 + a_{m2}x_2 + \cdots a_{mn}x_n))$$

Then we get:

$$\min(z)$$
$$z \geq a_{11}x_1 + \cdots a_{1n}x_n$$
$$\vdots$$
$$z \geq a_{m1}x_1 + a_{m2}x_2 + \cdots a_{mn}x_n$$

Note that reformulating in terms of inequalities allows us to represent the interior maximum function.

- *Ratio of objectives*: In some applications the following non-linear objective arises

$$max\left(\frac{\sum_j a_j x_j}{\sum_j b_j x_j}\right)$$

This clearly non-linear objective can be linearized using the following procedure

(i) Replace the expression $\frac{1}{\sum_j b_j x_j}$ by a variable t, such that we get

$$max\left(t\sum_j a_j x_j\right)$$

(ii) Represent the products tx_j by variable w_j, the the objective becomes

$$max\left(\sum_{j} a_j w_j\right)$$

(iii) Define variable t explicitly, such that:

$$t\sum_{j} b_j x_j = \sum_{j} b_j w_j = 1$$

(iv) Finally, transform the original constraints from the original form

$$\sum_{j} d_j x_j = | \leq e$$

To the following form, by multiplying both sides by t

$$\sum_{j} d_j w_j - et = | \leq 0$$

The soft constraint trick

A linear programing constraint such as:

$$\sum_{j} a_j x_j \leq b$$

Rules out any solutions in which the sum over *j* exceeds the quantity *b*. In some circumstances, it may be beneficial to have a softer version of this constraint. For example, we may want to buy extra capacity or raw materials at a high price. To represent this type of circumstance, we can introduce a constraint

$$\sum_{j} a_j x_j - u \leq b$$

Where, u is a positive variable and it is assigned an adequate cost coefficient in the objective function.

If instead of an inequality we have an equality, such as this one:

$$\sum_{j} a_j x_j = b$$

Then, it is possible to allow for the right-hand side coefficient b to be overreached or underreached by modelling it as follows:

$$\sum_{j} a_j x_j + u - v = b$$

Where both u and v are positive variables and are weighted adequately in the objective function.

The absolute value trick

Consider the following optimization problem:

$$\min c^T|x|$$
$$s.t. Ax \le b$$

This problem can be transformed into a linear programming model using the following trick

$$\min c^T(x^+ + x^-)$$
$$\begin{aligned} s.t.\quad & Ax \le b \\ & x = x^+ - x^- \\ & x^+ \ge 0, x^- \ge 0 \end{aligned}$$

Note that there are many combinations of x^+ and x^- that would work for example $|3| = 4 - 1$, this combination however (i.e., $x^+ = 4$ and $x^- = 1$) will not work, because it does not minimize the objective $(4 + 1 = 5)$. Note that $|3| = 3 - 0$, and this combination, which is unique, minimizes the objective function (i.e., $3 + 0 = 3$).

Tricks to detect errors

It is possible to build a model with error detection in mind. For example, suppose, we want to define the following constraint:

$$\sum_j a_j x_j \le b$$

If there were any danger of this constraint being too severe such that the model is made infeasible, we could allow the constraint to be violated at a certain high cost (imposed on the objective function). In this way we can rewrite the constraint as follows:

$$\sum_j a_j x_j - u \le b$$

The value u must be penalized in the objective function (what do I mean by that?, that is, how do you penalize something in the objective function?). Note that if it is not zero in the optimal solution, this means that this constraint would be violated in the original form.

Linear programming examples

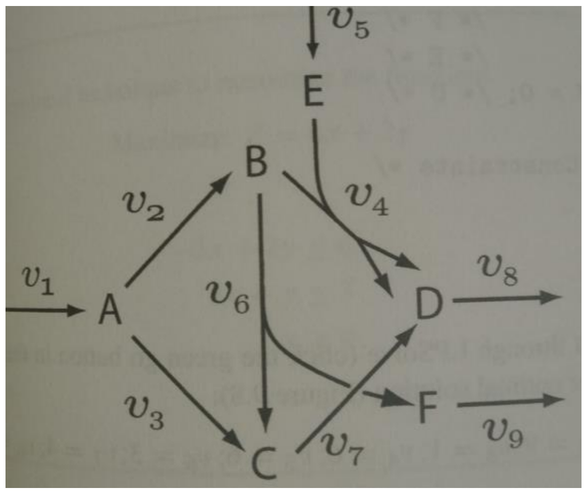
- Example (metabolic flux balance): In metabolic engineering, a key application is what is known as flux analysis, the purpose of this technique is to find the steady state flows across a metabolic network (v_i). The problem in these cases arises because we do not have enough information, and more than one flux distribution is possible. Let's consider the following reaction set, and for the sake of simplicity, let's assume that this set represents metabolism.

$$Feedstock \rightarrow A\ (v_1)$$
$$A \rightarrow B\ (v_2)$$
$$A \rightarrow C\ (v_3)$$
$$B + E \rightarrow 2D(v_4)$$
$$(Feedstock \rightarrow E\ (v_5)$$
$$B \rightarrow C + F(v_6)$$
$$C \rightarrow D(v_7)$$

$$D \rightarrow biomass(v_8)$$
$$F \rightarrow biomass\ (v_9)$$

In general, we will say that a sequence of reactions such as this one constitutes a reaction network. The information that is contained in this network can be captured using two tools

- All these are chemical equations, and they can be represented in a graph as follows:



- We can define what is known as a stoichiometric matrix, this is an $m \times n$ matrix where m is the number of species, and n is the number of reactions. For the example that we are working the stoichiometric matrix looks as follows:

$$N = \begin{array}{c} / \\ A \\ B \\ C \\ D \\ E \\ F \end{array}\begin{array}{c} \begin{array}{ccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 \end{array} \\ \begin{bmatrix} 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

To find the flux distribution in this network, we can use optimization. Toward this end, we need to define both an objective function and a set of valid constraints able to capture the physics of the problem. The objective reflects our understanding of metabolism. Let's assume that our objective is to maximize biomass production (that is we are saying that the objective of a cell is to grow). Since fluxes v_8 and v_9 correspond to biomass. An objective function can be a weighted sum of these fluxes.

$$\max\ (av_9 + bv_8)$$

Secondly, we need to add constraints corresponding to the mass balances at steady state. Toward this end, we note that at steady state the net rate of formation/disappearance is equal to zero, thus:

$$v_1 - v_2 - v_3 = 0 \text{ (For A)}$$
$$v_2 - v_6 - v_4 = 0 \text{ (For B)}$$
$$v_3 + v_6 - v_7 = 0 \text{ (For C)}$$
$$2v_4 - v_8 + v_7 = 0 \text{ (For D)}$$
$$v_5 - v_4 = 0 \text{ (For E)}$$
$$v_6 - v_9 = 0 \text{ (For F)}$$

Note that this set of constraints can be compactly represented using the stoichiometric matrix as follows:

$$Nv = 0$$

Finally, we make assumptions about the incoming fluxes, in practice, you can think that these fluxes may be measured, for example by an experimental colleague. For the sake of discussion let's assume the following values as constraints.

$$v_1 = 10$$
$$v_5 = 6$$

- Example (food manufacturing): a food is manufactured by refining raw oil and blending them together. The raw oils are of two categories:

Vegetable oils	Veg 1
	Veg 2
Non-vegetable oils	Oil 1
	Oil 2
	Oil 3

Each oil may be purchased for immediate delivery (January) or bought on the futures market for delivery in a subsequent month. Prices at present and in the futures market are given below:

	VEG 1	VEG 2	OIL 1	OIL 2	OIL 3
Prices	110	120	130	110	115

The final product sells for 150.

Vegetable oils require different production lines for reining. In any different month, it is not possible to refine more than 200 tons of vegetable oil and more than 250 tons of non-vegetable oils.

There is a technological restriction of hardness on the final product. In the units in which hardness is measured, this must lie between 3 and 6. It is assumed that hardness blends linearly and that the hardness of the raw oils are:

VEG 1	8.8
VEG 2	6.1
OIL 1	2.0
OIL 2	4.2

OIL 3	5.0

What buying and manufacturing policy should the company pursue in order to maximize profit?'

In this case we can define the objective function as the profit to be obtained:

$$Profit = 150y - 110x_1 - 120x_2 - 130x_3 - 110x_4 - 115x_5$$

The first constraints come from understanding the limits on the amount of vegetable and non-vegetable oil that can be purchased:

$$x_1 + x_2 \leq 200$$

$$x_3 + x_4 + x_5 \leq 250$$

Then we need to consider the hardness constraints (what does it mean that blending is linear?)

$$8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_6 \leq 6y$$

$$8.8x_1 + 6.1x_2 + 2x_3 + 4.2x_4 + 5x_6 \geq 3y$$

Finally, we need to define the total amount of product that is made in the system, as follows:

$$x_1 + x_2 + x_3 + x_4 + x_5 = y$$

- Example (multiperiod planning): Let's consider a small modification to the previous problem, let's assume that we have a time horizon of three months now, and that prices change on a monthly basis as follows:

	VEG 1	VEG 2	OIL 1	OIL 2	OIL 3
Month 1	110	120	130	110	115
Month 2	130	130	110	90	115
Month 3	120	110	120	120	125
Month 4	90	100	140	80	135

In this case we will assume that you can store up to 1000 tons each raw oil for use later. The cost of storage is 5/ton-month. The final product cannot be stored.

At present there are 500 tons of product of each type of raw oil in storage. It is required that these stocks will also exist at the end of the planning period.

The fundamental difference here is that now we need to handle the fact that we can store and that the amount of product that we are storing can change on a monthly basis. In order to achieve this we need couple of things (1) we need to define independent variables for each month, (2) we need to have variables to track the product that is bought (B), used (U), and stored (S) each month. The following table summarizes them:

	VEG 1	VEG 2	OIL 1	OIL 2	OIL 3
Month 1	BV_{11}	BV_{12}	BO_{11}	BO_{12}	BO_{13}

P_1	UV_{11} SV_{11}	UV_{12} SV_{12}	UO_{11} SO_{11}	UO_{12} SO_{12}	UO_{13} SO_{13}
Month 2 P_2	BV_{21} UV_{21} SV_{21}	BV_{22} UV_{22} SV_{22}	BO_{21} UO_{21} SO_{21}	BO_{22} UO_{22} SO_{22}	BO_{23} UO_{23} SO_{23}
Month 3 P_3	BV_{31} UV_{31} SV_{31}	BV_{32} UV_{32} SV_{32}	BO_{31} UO_{31} SO_{31}	BO_{32} UO_{32} SO_{32}	BO_{33} UO_{33} SO_{33}
Month 4 P_4	BV_{41} UV_{41} SV_{41}	BV_{42} UV_{42} SV_{42}	BO_{41} UO_{41} SO_{41}	BO_{42} UO_{42} SO_{42}	BO_{43} UO_{43} SO_{43}

This is a lot of variables, and we do not write that much all the time, it is boring, so to deal with this it is very common that while we write models, we define sets, and use them as indexes to define variables and equations. In this case I am going to define the following sets

$$V = \{VEG1, VEG2\}$$

$$O = \{OIL1, OIL2, \text{OIL 3}\}$$

$$M = \{MONTH1, MONTH2, MONT3, MONTH4\}$$

Now that we are equipped with this, we can start writing our model. Our objective function is to maximize profit. So we can write the objective function as follows:

$$\sum_{m \in M} 150 P_m - \sum_{m \in M} \sum_{v \in V} pr_{v,m} BV_{v,m} - \sum_{m \in M} \sum_{o \in O} pr_{o,m} BO_{o,m} - \sum_{m \in M} \sum_{v \in V} scSV_{v,m} - \sum_{m \in M} \sum_{o \in O} scSO_{o,m}$$

Now we proceed to write constraints for this system, the first of this constraint can be interpreted as a mass balance at each point in time, it establishes that the amoiun bough plus the amount store in the previous month, is equal to the amount used plus the amount stored in the current month

$$SV_{m-1,v} + BV_{m,v} = UV_{m,v} + SV_{m,v}, \forall\, m \in M, v \in V$$

$$SO_{m-1,o} + BO_{m,o} = UO_{m,o} + SO_{m,o}, \forall\, m \in M, v \in V$$

Now we see that in writing the equations following this approach a variable $SV_{o,v}$ and $SO_{0,v}$ appears in the model, this variable constitutes the initial inventory. Likewise, $SV_{4,v}$ and $SO_{4,v}$ appear in the model, they represent the inventory in month 4, so we can write the following constraints based on the inventory requirements for the problem

$$SV_{o,v} = SV_{4,v} = SO_{0,v} = SO_{4,v} = 500$$

Now, we need to slightly modify the quality control, capacity, and product definition constraints in the original problem, the idea is that we now need an equation per period.

For the capacity, now we can write:

$$\sum_{v \in V} UV_{m,v} \leq 200, \forall m \in M$$

$$\sum_{v \in V} OV_{m,v} \leq 200, \forall m \in M$$

The product definition can be written as follows:

$$P_m = \sum_{v \in V} UV_{m,v} + \sum_{o \in O} UO_{m,v}\,, \forall m \in M$$

Finally, the product quality must be enforced at each period such that

$$\sum_{v \in V} qi_v UV_{m,v} + \sum_{o \in O} qi_o UO_{m,v} \leq 6P_m, \forall m \in M$$

Finally, we constraint the stored amount as follows:

$$SV_{m,v} \leq 1000, \forall m \in M, v \in V$$

$$SO_{m,o} \leq 1000, \forall m \in M, o \in O$$