CHAPTER 1

THE IMPORTANCE AND NATURE OF OPTIMIZATION PROBLEMS

All materials in this class rely on the original sources on the syllabus, mistakes are mine, merit belong to others

- The fundamental point that you should remember from here is that in optimization problems you have **degrees of freedom**.
- The second point that you should remember is that optimization is about decision making, a degree of freedom is a choice that as a **designer you need to make**.
- The third point that you should remember is that in optimization problems **you may have local and global minima**.
- The nature and **complexity of the models** that you generate will depend on (1) the **nature of the variables** that you use to write the model (real or integer) (2) the **number of objectives** (one or multiple), (3) your treatment of **uncertainty**, and (4) the existence or inexistence of constraints.

What is optimization anyways?

In your engineering education you have been exposed to several fundamental concepts that allow you to talk about the world. These concepts are usually discussed in the study of physics, chemistry, and biology, and in general you can use them to build conceptual models of the world. A model in this context can be understood as an abstraction capable of retaining **some** features of the real world. As chemical engineers, most often, we deal with transport equations, kinetics, thermodynamics (in all of its flavors), and also with biological systems and the laws and empirical relations used to describe them.

This class is a bit different, it is not aimed at gaining physical insights per-se about these different domains in chemical engineering, but rather, to illustrate how we can think about engineering problems as optimization problems, that is, we aim to understand how we can systematically construct and probe models of the world with the aim of making better design choices. The models that we are interested in studying are mathematical in nature (that is, they consist of a series of equations, even more, we will see that in most cases we can say that the model can be reduced to a series of algebraic equations). The models that we are interested in exploring have a particularity, in these problems, we have the freedom to make some decisions, such that the value of these decisions impacts a measure of performance that we are interested in evaluating. We will refer to this type of problems as **mathematical programming** (the programming here is more in the sense of scheduling/organizing rather than in the sense of coding) problems or optimization problems. For the sake of illustration, let's consider the following problem.

Problem: Find the dimensions of a rectangle with area 1000m^2 whose perimeter is as small as possible.

This is a very simple calculus problem, that we have all solved at some point in life, yet, let me use it to introduce couple of concepts. The first one is the notation in itself of the optimization problem. We can translate this problem into math, if we define two variables x and y representing the sides of the rectangle. Additionally, we define an objective function P that should be minimized. Written as this we can formulate the problem as follows:

$$\min_{x,y} P \quad (1)$$

$$s.t.\quad \begin{array}{c} xy = 1000 \\ P = 2x + 2y \\ x \geq 0 \\ y \geq 0 \end{array}$$

We can determine the degrees of freedom as the difference between the number of variables (3 in total, x, y, P), and the number of **equalities** (2 in total). Note the emphasis on equalities, inequalities do not affect the number of degrees of freedom. Thus, the number of degrees of freedom in this case is 1. Having one degree of freedom means that there is a decision to make, that is, there is more than one set of variables satisfying the area condition that we imposed (For example we can choose, $x = 1$, y=1000; or x=100, $y = 10$, and so on, and so forth). In optimization problems there is always a positive number of degrees of freedom. If the number of degrees of freedom is zero, then we are dealing with a simulation problem, only one solution exists, therefore, there are no decisions to be made.

We see that in the previous problem there are several elements that we need to consider to define an optimization problem:

- We have an objective function, something that we want to use as a metric for our success, and we want to maximize or minimize, here the perimeter.
- We have a set of constraints, these constraints limit the decisions that we can make, in our case there is one constraint related to the area of the rectangle, and two constraints establishing that the lengths of the rectangle sides must be positive. These constraints can be either equalities or inequalities.
- There are some variables that will be obtained as a result of solving the optimization problem, in the example, x and y

Formulating and building these models is an activity that may render numerous advantages to an engineer:

- Building a model reveals relations that are not apparent to many people. Thus, deeper understanding is achieved about the object of interest.
- Having built a model, you can use it to determine possible courses of action.
- Experimentation is possible with a model, usually, the testing of conditions that are dangerous can be achieved computationally with little to no risk.

An important consideration to keep in mind is that the model that we have used to represent our problem is not unique, other (perhaps more efficient formulations are possible). For example, one does not need to define the variable perimeter, it is enough to write:

$$\min_{x,y} 2x + 2y \quad (2)$$

$$
\begin{aligned}
& xy = 1000 \\
s.t. \quad & x \geq 0 \\
& y \geq 0
\end{aligned}
$$

Now that we have briefly introduced how optimization problems are typically expressed, we can define optimization, the following, is ChatGPT response (I wanted to see what it came up with and I liked it):

> *"Optimization is the process of finding the best solution to a problem by maximizing or minimizing an objective function, subject to constraints. It involves selecting the most efficient option from a set of feasible(possible) choices, often applied in engineering, economics, and operations research to improve performance."*

Structure of an optimization problem

Many optimization problems can be represented using the following mathematical form:

$$
\begin{gathered}
\min \; F(x,y) \text{ (3)} \\
s.t \begin{matrix} h(x,y) = 0 \\ g(x,y) \leq 0 \end{matrix} \\
x \in R^n, y \in \{0,1\}^m
\end{gathered}
$$

In this representation, $F(x,y)$ is a function that contains the objective(s) that I care about (yes, it can be more than one, as we will see). $h(x,y)$ represents equality constraints that must be satisfied and $g(x,y)$ represent inequality constraints. Note that we have included in this general representation both real variables $x \in R^n$ and binary variables $y \in \{0,1\}^m$ which we will use extensively to model decision making.

Associated with this optimization formulation, there are several concepts that allow us to talk about this problem

- *Feasible region:* the feasible region is given by the set of points $[x, y]$ that satisfy all equality and inequality constraints.
    - Likewise, we say that a point (x, y) is feasible if it satisfies all the constraints.
- *Local minimum:* Let $Y = \{x \in R^n \; and \; y \in \{0,1\}^m | h(x) = 0, g(x) \leq 0 \}$ be the feasible set (another way to refer to the feasible region). Let $F: (R^n, \{0,1\}^m \to R)$ (that is a scalar function, meaning a single objective, we will deal with more objectives later). The vector (x^*, y^*) is called a local minimum of the problem if there exists $\varepsilon > 0$ such that:

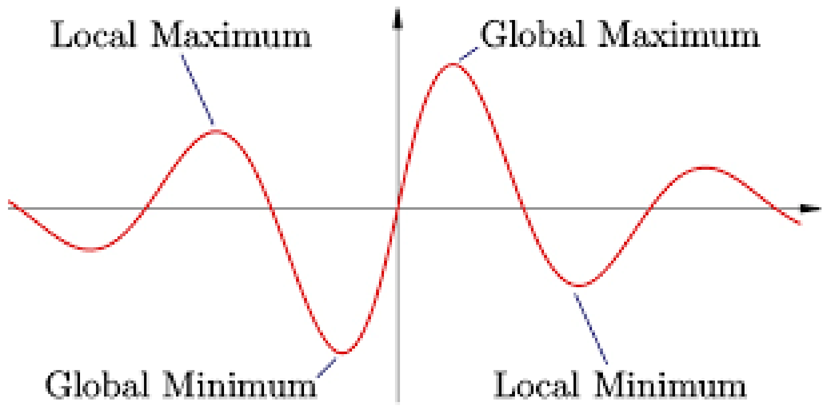
  $$F(x^*, y^*) \leq F(x,y) \forall x \in Y \text{ such that } \left||x - x^*|\right| < \varepsilon \text{ (4)}$$

  Note that what we are saying here is that there is a point (x^*, y^*) and a neighborhood (a ball) around that point $\left||x - x^*|\right| < \varepsilon$, such that the lower value of the function, in this neighborhood is attained at (x^*, y^*).

  If the inequality is hold strictly ($<$) we call this a strict local minimum.
- *Global minimum:* Let $Y = \{x \in R^n \; and \; y \in \{0,1\}^m | h(x) = 0, g(x) \leq 0 \}$ be the feasible set. Let $F: (R^n, \{0,1\}^m \to R)$. The vector (x^*, y^*) is called a global minimum of the problem if:

$$F(x^*, y^*) \leq F(x, y) \forall x \in Y$$

Again, if the inequality is hold strictly (<) we call this a strict global minimum.



Types of optimization problems

The structure and nature of an optimization problem will determine how easy is to find the solution for this problem. In general, we will discriminate problems based on four fundamental criteria (1) the type of variables that we are using to model the system (2) the number of objective functions (3) the treatment that we do of uncertainty, and (4) the existence or inexistence of constraints.

Models based on variable types: a general optimization problem, in which we assume that all parameters are known with certainty, and we have a single objective can be written as follows:

$$\begin{gathered} \min \; F(x, y) \text{ (3)} \\ s.t \begin{matrix} h(x, y) = 0 \\ g(x, y) \leq 0 \end{matrix} \\ x \in R^n, y \in \{0,1\}^m \end{gathered}$$

- If all functions are linear, and we only have real variables, then we will have a linear programing problem

$$\begin{gathered} \min \; F(x)(linear)\text{(4)} \\ s.t \begin{matrix} h(x) = 0(linear) \\ g(x) \leq 0 \; (linear) \end{matrix} \\ x \in R^n \end{gathered}$$

- If all functions are linear and we have both integer and real variables then we have a mixed integer linear programing problem (MILP)

$$\begin{gathered} \min F(x, y) \; (linear) \text{ (5)} \\ s.t \begin{matrix} h(x, y)(linear) = 0 \\ g(x, y)(linear) \leq 0 \end{matrix} \\ x \in R^n, y \in \{0,1\}^m \end{gathered}$$

- If any function is non-linear and we have only real variables then we have a non-linear programing problem (NLP)

$$\min\ F(x)(linear\ or\ non-linear)\ (6)$$
$$s.t \begin{matrix} h(x) = 0\ (linear\ or\ non-linear) \\ g(x) \leq 0\ (linear\ or\ non-linear) \end{matrix}$$
$$x \in R^n$$

- Finally, if we have any non-linear function, and integer and real variables, we have a mixed integer non-linear programming problem

$$\min\ F(x,y)(linear\ or\ non-linear)\ (7)$$
$$s.t \begin{matrix} h(x,y) = 0\ (linear\ or\ non-linear) \\ g(x,y) \leq 0\ (linear\ or\ non-linear) \end{matrix}$$
$$x \in R^n, y \in \{0,1\}^m$$

Models based on number of objective functions. If there is a single objective function, that is, if $F(x,y)$ is a scalar function, then we are dealing with a single objective problem. If $F(x,y)$is a vectorial function, then we are dealing with a multi-objective optimization problem.

Models based on their treatment of uncertainty. If all the parameters in the model are known with certainty, then we call this a deterministic problem. If uncertainty in the parameters is considered then we are dealing with an optimization under uncertainty problem, and different approaches such as stochastic programming, robust optimization, or chance constraint optimization can be used.

Models based on the existence of constraints: depending on the presence or absence of constraints model can be considered constrained or unconstrained. This last group is significantly more complex.

A workflow for optimization problems

The question that we pose as chemical engineers is how I go on an use optimization tools to solve the problems that we care about. In general, we can break out the process in three steps:

1. *Modelling*: You need to model your system, such that you pose your problem in an amenable way to be solved via optimization.
2. *Coding*: You need to translate your model into a version that a computer may read, typical tools that we use include GAMS, Julia, and Python (the Pyomo environment is the one that is used for optimization)
3. *Solving:* You need to find the solution to the optimization problem using a suitable algorithm
4. *Analyzing:* Finally, you need to make sense of your results

Most classes on optimization are focus on the third step, that is they typically present formal results related to the nature of optimization followed by a collection of algorithms and the theory behind them assuming that you have a working model. This class is a bit different, here we will train and devote some time to model formulation, and as we progress, we will also study the most important theoretical results in optimization, some algorithms in detail, and we will discuss other relevant algorithms that exists. The focus of the class is not in training you on how to implement an optimization algorithm, but rather on how to formulate an optimization model and then select and tweak an algorithm to find the solution to a problem of interest.

Homework:

A food is manufactured by refining raw oils and blending them together. The raw oil come in two categories Vegetable (V1 and V2) and non-vegetable (O1M O2M O3). Vegetable and non-vegetable oils require different production lines for refining. In any month it is not possible to refine more than 200 tons of vegetable oil and more than 250 tons of non-vegetable oil. There is no loss of weight in the refining process and the cost of refining may be ignored. There is a technological restriction of hardness in 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 with mass fraction. The final product sells for $150/ton. The costs per ton and hardness of the oils are:

	V1	V2	O1	O2	O3
Cost	110	120	130	110	115
Harness	8.8	6.1	2	4.2	5

Formulate an optimization problem to find the optimal final product composition.