CHAPTER 10
SOLVING MULTIOBJECTIVE OPTIMIZATION PROBLEMS

- The central goal of this chapter is to learn how we can deal with optimization problems in which there is more than one objectives. We will discuss a few methods, the most prominent ones are:
    - Weighted sum method
    - Epsilon constraint method
    - Chance constraint method

Introduction

Multiobjective optimization problems are ubiquitous in engineering, often we care about more than one thing. Specially, in sustainability analysis, this type of problem appears commonly, as one is always interested in making money (maximize profit), and if one cares about sustainability in minimizing environmental impacts. Formally, multi-objective optimization problems can be written as follows:

$$\min \boldsymbol{Z} = [Z_1(x,y), \dots, Z_k(x,y)]$$

$$\begin{aligned} & h(x,y) = 0 \\ s.t. \quad & g(x,y) \leq 0 \\ & x \in R^n, y \in \{0,1\}^m \end{aligned}$$

Mult objective optimization can bet considered as a methodology or group of methodologies for generating a preferred solution or range of efficient solutions to a decision problem.

Non dominated set

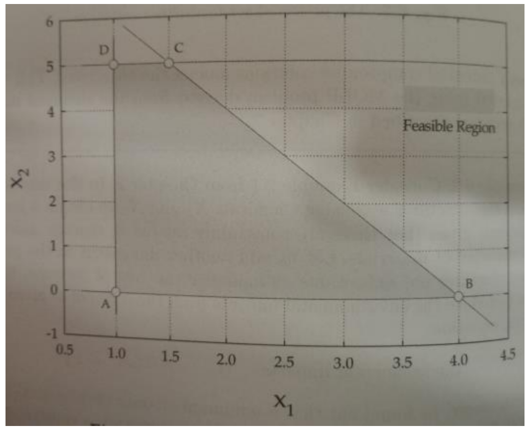
The non dominated set consists of the collection of alternatives that represent a potential compromise among objectives.

Example: Let's consider the following multi-objective optimization problem.

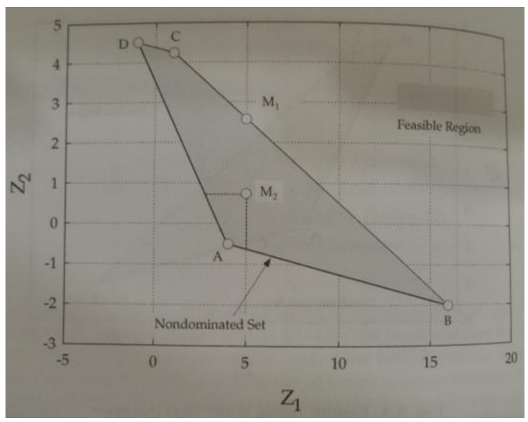
$$\min \begin{matrix} Z_1 = 4x_1 - x_2 \\ Z_2 = -0.5x_1 + x_2 \end{matrix}$$

$$\begin{gathered} x_1 \geq 1 \\ 2x_1 + x_2 \leq 8 \\ x_2 \leq 5 \\ x_1 - x_2 \leq 4 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{gathered}$$

The feasible set associated with the previous problem is shown in the following figure:



Each point on the feasible set can be mapped into the space given by the two objective functions of interest. When we do that, we obtain the following representation:



The notion of a non-dominated/dominated point, and non-dominated set can be easily understood if we look at points M1 and M2 in the figure. Note that in both cases it is possible to find other points for which there is an improvement in both objectives, that is the case of point A. In this case, we say that point A dominates both points M1 and M2. We can see that the points along the boundary DAB have a special quality, that is, for these points, it is not possible to find another feasible point that simultaneously improves both objectives. This boundary is known as the set of non-dominated points or the Pareto set.

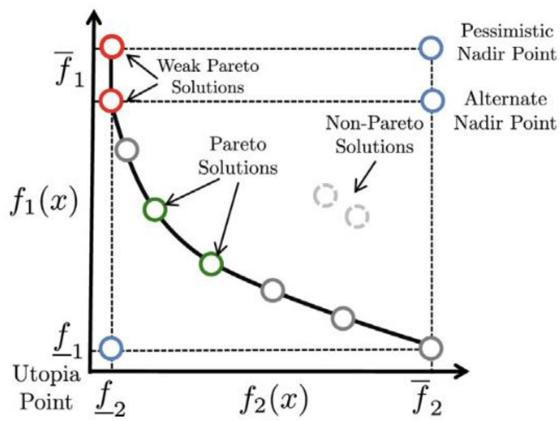
Formally, the non-dominated solution can be defined as follows, if x^* is feasible and nondominated, then there is no other feasible solution x_2^* such that

$$Z_p(x^*) \leq Z_p(x_2^*), \forall p \in P$$

With at least one of this inequalities being a strict inequality (assuming that all objectives are to be minimized).

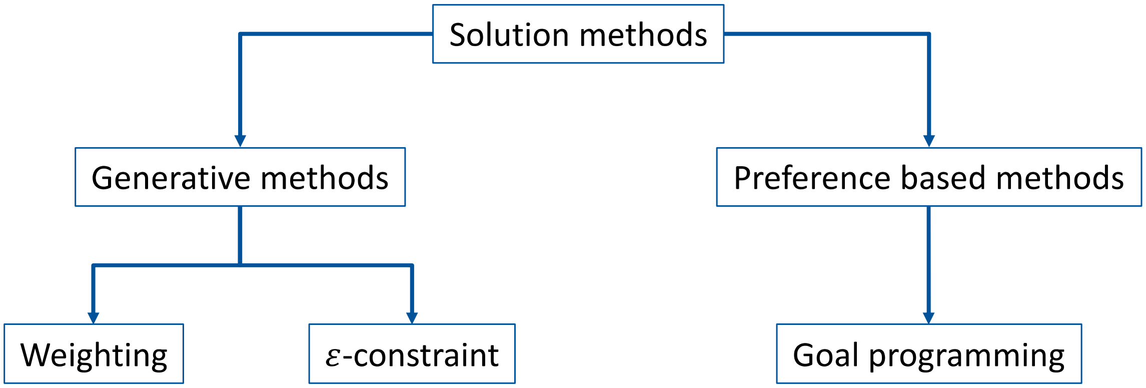
Conceptually, the Pareto front represents the compromise solution between a set of objectives. Some geometric considerations and concepts are summarized in the following figure:

- Weak and strong pareto solutions: A weak Pareto solution is a feasible outcome where no objective can be improved without worsening at least one other, but some objectives might remain unchanged. It contrasts with a strict Pareto optimal solution, where improving any objective necessarily worsens at least one other.
- The **utopia point** represents the ideal outcome in multi-objective optimization, where each objective achieves its best possible value. It is typically unattainable but serves as a reference.
- The **nadir point** indicates the worst objective values among all Pareto optimal solutions.



Solution methods

Broadly speaking solution methods for multi-objective optimization can be divided into two big groups: preference-based methods and generative methods. In the first case, the goal is to capture the decision making preferences in order to produce a single solution to the optimization problem. In the second case, the decision maker selects an optimal solution that accommodates their expectations by selecting the most convenient one from the Pareto front. The main advantage of preference-based methods is that they do not intend to produce the whole Pareto frontier, thus they are more efficient. However, they require a lot of previous knowledge and understanding on the decision maker side. Generative methods provide a lot more information, specially, they are typically tailored to produce the Pareto set. Furthermore, they do not require the preferences of the decision maker as an input. The main limitation is (may be) comuputational. Since you need to estimate the Pareto front, it may be necessary that you employ a lot of computational resources.



- *Weighting method:*

The fundamental idea of the weighting method is to associate each objective function with a weighting coefficient and minimize the weighted sum of the objectives. In this way, the multi-objective optimization problem is transformed into a series of single optimization problems. In general, the problem takes the following form:

$$\min Z = \sum_{o \in O} w_o Z_o(x, y)$$

$$\begin{aligned} s.t. \quad h(x,y) &= 0 \\ g(x,y) &\leq 0 \\ x \in R^n, y &\in \{0,1\}^m \end{aligned}$$

Where O represents the set of objectives of interest. An important theoretical result establishes that if the following two conditions are met, then we have a Pareto front:

$$w_o > 0, \forall o \in O$$

$$\sum_{o \in O} w_o = 1$$

Example: We can return to the previous example that we were disussing, in this case, the multi-objective problem can be transformed as follows:

$$\begin{gathered} \min \begin{matrix} Z_1 = 4x_1 - x_2 \\ Z_2 = -0.5x_1 + x_2 \end{matrix} \\ x_1 \geq 1 \\ 2x_1 + x_2 \leq 8 \\ x_2 \leq 5 \\ x_1 - x_2 \leq 4 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{gathered}$$

$$\begin{gathered} \min w_1 Z_1(x_1, x_2) + w_2 Z_2(x_1, x_2) \\ x_1 \geq 1 \\ 2x_1 + x_2 \leq 8 \\ x_2 \leq 5 \\ x_1 - x_2 \leq 4 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ Z_1(x_1, x_2) = 4x_1 - x_2 \\ Z_2(x_1, x_2) = -0.5x_1 + x_2 \end{gathered}$$

- *ε-Constraint method:*

In the ε-Constraint method the goal is similar to the weighting method in the sense that one is looking to transform a multi-objective problem into a single objective optimization problem. The way in which this is by selecting a preferential objective and then using all other objectives as parametric constraints. The idea is to systematically explore this parametric space with the goal of reconstructing the Pareto front. In general, the multiobjevtive optimization problem takes the following form:

$$\min Z = Z_i(x,y)$$

$$s.t.\quad \begin{array}{c} Z_j(x,y) \le \varepsilon_j, \forall j \ne i \\ h(x,y) = 0 \\ g(x,y) \le 0 \\ x \in R^n, y \in \{0,1\}^m \end{array}$$

Example: For the problem that we have been working, this transformation looks as follows:

$$\min \begin{array}{c} Z_1 = 4x_1 - x_2 \\ Z_2 = -0.5x_1 + x_2 \end{array}$$
$$\begin{array}{c} x_1 \ge 1 \\ 2x_1 + x_2 \le 8 \\ x_2 \le 5 \\ x_1 - x_2 \le 4 \\ x_1 \ge 0 \\ x_2 \ge 0 \end{array}$$

$$\begin{array}{c} \min Z_1(x_1,x_2) \\ x_1 \ge 1 \\ 2x_1 + x_2 \le 8 \\ x_2 \le 5 \\ x_1 - x_2 \le 4 \\ x_1 \ge 0 \\ x_2 \ge 0 \\ Z_1(x_1,x_2) = 4x_1 - x_2 \\ \boldsymbol{Z_2(x_1,x_2) = -0.5x_1 + x_2 \le \varepsilon} \end{array}$$

- *Goal programming method:*

In this method, the fundamental goal is not to find the Pareto front, but rather find a solution that accommodates to an original set of expectations. Implicitly, we are stating that we have clear what are these expectations. Mathematically, we can formulate the problem as follows:

$$\min Z = \sum_{o \in O} |Z_i(x,y) - G_i|$$

$$s.t.\quad \begin{array}{c} h(x,y) = 0 \\ g(x,y) \le 0 \\ x \in R^n, y \in \{0,1\}^m \end{array}$$

Where the value of G_i represents the goal of the decision maker. One limitation of the proposed approach is that it is non-differentiable at the origin. This can be simplified by reformulating the problem as follows:

$$\min Z = \sum_{o \in O} \delta_i^+ + \delta_i^-$$

$$s.t. \begin{gathered} h(x,y) = 0 \\ g(x,y) \le 0 \\ Z_i(x,y) - G_i = \delta_i^+ - \delta_i^- \\ \delta_i^+, \delta_i^- \ge 0 \\ x \in R^n, y \in \{0,1\}^m \end{gathered}$$

Example: Let's consider the problem that we have been exploring and let's assume that we have a target for goal Z_1 to be -5, and for Z_2 to be -5. In this case, the multiobjective optimization problem can be written as follows:

$$\min \begin{gathered} Z_1 = 4x_1 - x_2 \\ Z_2 = -0.5x_1 + x_2 \end{gathered}$$
$$\begin{gathered} x_1 \ge 1 \\ 2x_1 + x_2 \le 8 \\ x_2 \le 5 \\ x_1 - x_2 \le 4 \\ x_1 \ge 0 \\ x_2 \ge 0 \end{gathered}$$

$$\min \sum_{i \in \{1,2\}} \delta_i^+ + \delta_i^-$$
$$\begin{gathered} Z_1(x_1,x_2) + 5 = \delta_1^+ + \delta_1^- \\ Z_2(x_1,x_2) = \delta_2^+ + \delta_2^- \\ x_1 \ge 1 \\ 2x_1 + x_2 \le 8 \\ x_2 \le 5 \\ x_1 \ge 0 \\ x_2 \ge 0 \\ Z_1(x_1,x_2) = 4x_1 - x_2 \\ Z_2(x_1,x_2) = -0.5x_1 + x_2 \\ \delta_i^+, \delta_i^- \ge 0, \forall i \in \{1,2\} \end{gathered}$$

- *Goal attainment method:*

In the goal attainment method, goals are set for each of the objective ($F_i(x,y)$), these goals are often determined by solving single objective optimization problems. Such that

$$G_i^* = \min F_i(x,y)$$

$$s.t.\, g_j(x,y) \le 0, \forall j \in J$$

Now, we can define weights w_i, to represent the relative importance of the objectives. Finally, we introduce a scalar variable γ. Using these elements, we can formulate the multi-objective optimization problem as follows:

$$\min \gamma$$

$$s.t. \begin{gathered} g_j(x,y) \le 0, \forall j \in J \\ F_i(x,y) - \gamma w_i \le G_i^* \\ \sum_{i=1}^{k} w_i = 1 \end{gathered}$$

- *Lexicographic method:*

In the lexicographic method, the objectives are ranked in order of importance by the designer (meaning that the person in charge of the problem formulation needs to know their priorities). The optimal value is found by minimizing the objective functions sequentially, starting by the most important objective. The optimal value obtained for each objective higher in the hierarchy, is added as a constraint for the estimation of subsequent objectives.

The is, if we have F_k objective functions, we proceed in the following way

$$F_1^* = \min F_1(x,y)$$

$$s.t.\, g_j(x,y) \leq 0, \forall j \in J$$

Then for the second objective, we solve the following problem:

$$F_2^* = \min F_2(x,y)$$

$$s.t. \begin{array}{c} g_j(x,y) \leq 0, \forall j \in J \\ F_1(x,y) = F_1^* \end{array}$$

This procedure is repeated until all the objectives have been considered. For the ith problem, we can write that:

$$F_i^* = \min F_i(x,y)$$

$$s.t. \begin{array}{c} g_j(x,y) \leq 0, \forall j \in J \\ F_l(x,y) = F_l^*, \forall l \in \{1,2,\dots,i-1\} \end{array}$$

The solution obtained at the end is taken as the desired solution of the original multi-objective problem.

Conceptually, the lexicographic method relies on the assumption that it is possible that there is more than one solution to the problems higher in the objective hierarchy, we can use objectives with lower priority, to select the best among these available solutions. For example, if one has two solutions with equal economic cost, one an use an environmental impact to select the best among them.

- *Evolutionary methods:*

Evolutionary algorithms exist on their own right, they can be used to solve both real valued problems, and integer problems. However, in the case of multi-objective optimization they are specially popular. Conceptually evolutionary algorithms are aimed at mimicking some natural processes (like evolution), or at least they are inspired by them. They cannot be described as gradient based, since they do not use gradient information, instead they rely on a stochastic search of the design space. Note that the search is global, but there is no guarantee of optimality, regardless of the nature of the problem. The most popular of these methods is the genetic algorithm, the most popular application of genetic algorithms to multi-objetive optimization is called the Nondominated Sorting Genetic Algorithm (NSGA). In pseudocode, this algorithm is presented below:

Algorithm 2 Pseudocode of a MOEA.

$t \leftarrow 0$
Generate an initial population $P(t)$
while the stopping criterion is not fulfilled **do**
    Evaluate the objective vector $\mathbf{f}$ for each individual in $P(t)$
    Assign a fitness for each individual in $P(t)$
    Select from $P(t)$ a group of parents $P'(t)$ preferring the fitter ones
    Recombine individuals of $P'(t)$ to obtain a child population $P''(t)$
    Mutate individuals in $P''(t)$
    Combine $P(t)$ and $P''(t)$ and select the best individuals to get $P(t+1)$
    $t \leftarrow t + 1$
end while

This type of algorithm is popular in multi-objective optimization because it allows to reconstruct an approximation to the Pareto front. Unlike other methods, such as the weight method, this approach is not sensitive to discontinuities in the pareto frontier. Genetic algorithms usually require a large number of function evaluations (one per each population member). In multi-objective optimization, we take advantage of this feature, such that we use this large amount of information to better reconstruct the pareto frontier of the problem.

Note that this are stochastic methods, this means that if you run the problem two times, it is almost certain that you will not get exactly the same solution.