Lesson 1
Definitions

For more information visit:
www.esteco.com or send an e-mail to:
modeFRONTIER@esteco.com

modeFRONTIER
the multi-objective optimization and design environment

Introduzione al corso

1. Introduzione all'ottimizzazione – introduzione a modeFRONTIER

2. Pianificazione degli Esperimenti, analisi dei risultati
3. Algoritmi di ottimizzazione e metamodelli
4. Tecniche di supporto alle decisioni e verifica della robustezza delle soluzioni
5. Strategie di ottimizzazione, esempi industriali

Definitions

- Optimization problem
- Input Variables
- Objectives
- Pareto Dominance
- Robustness and Accuracy
- Constraints
- Utility Function
- Robust Design

companies need to optimise products & processes

What is optimization?

Selection of the **best option** from a range of possible choices.

What makes it a complex task?

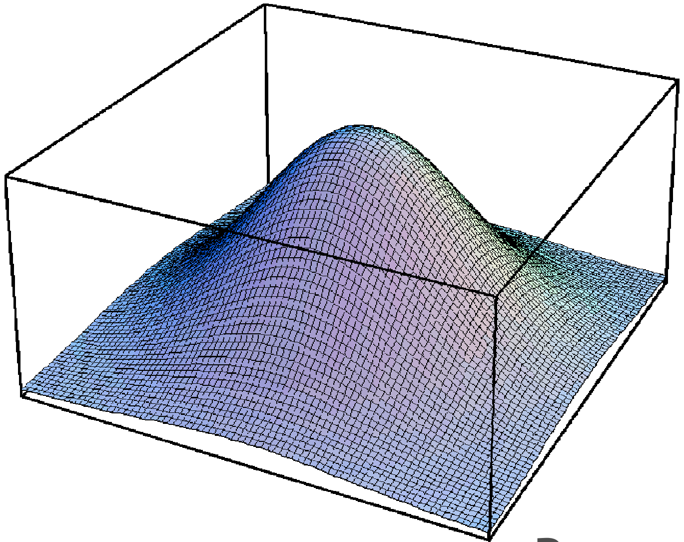
The potentially **huge number** of options to be tested

What qualifies as an optimization technique?

The **search strategy**

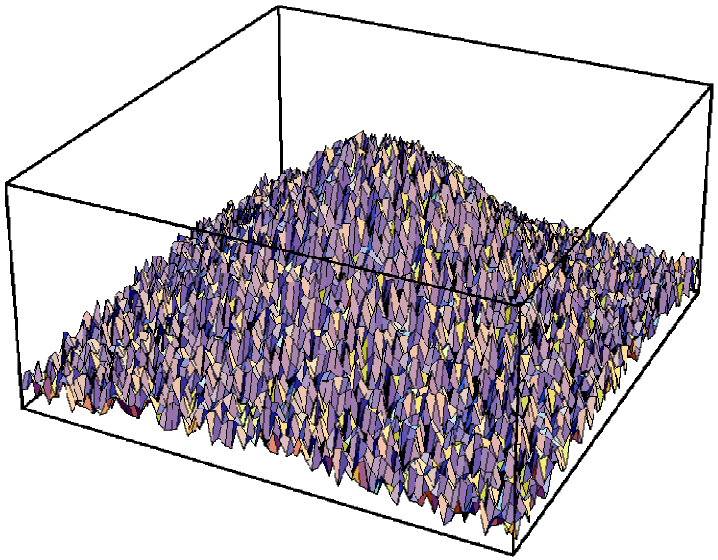
Real-world optimization

There is a huge difference between mathematical optimization and optimization in the real-world applications



Ideal function in the mathematical world

Rugged hill in the experimental world



Optimization Problem

- **Mathematical formulation**

$$\max \quad [f_1(x_1, \ldots, x_n), f_2(x_1, \ldots, x_n), \ldots, f_k(x_1, \ldots, x_n)]$$

$$\text{subject to} \quad \begin{cases} g_i(\overline{x}) \leq 0 \\ g_j(\overline{x}) \geq 0 \\ g_l(\overline{x}) = 0 \\ \overline{x} \in S \end{cases}$$

Note : When k>1 and the functions are in contrast, we speak about multi-objective optimization.

Variables

Variables:

Variables are the free parameters, i.e. the quantities that the designer can vary or the choices the designer can make.

- **Continuous** variables:
  - point coordinates
  - process variables
- **Discrete** variables:
  - components from a catalogue
  - number of components

Objectives

Objectives:

Objectives are the **response parameters**, i.e. the quantities that the designer wish to be MAX or MIN

MAX

- efficiency
- performance
- etc...

MIN

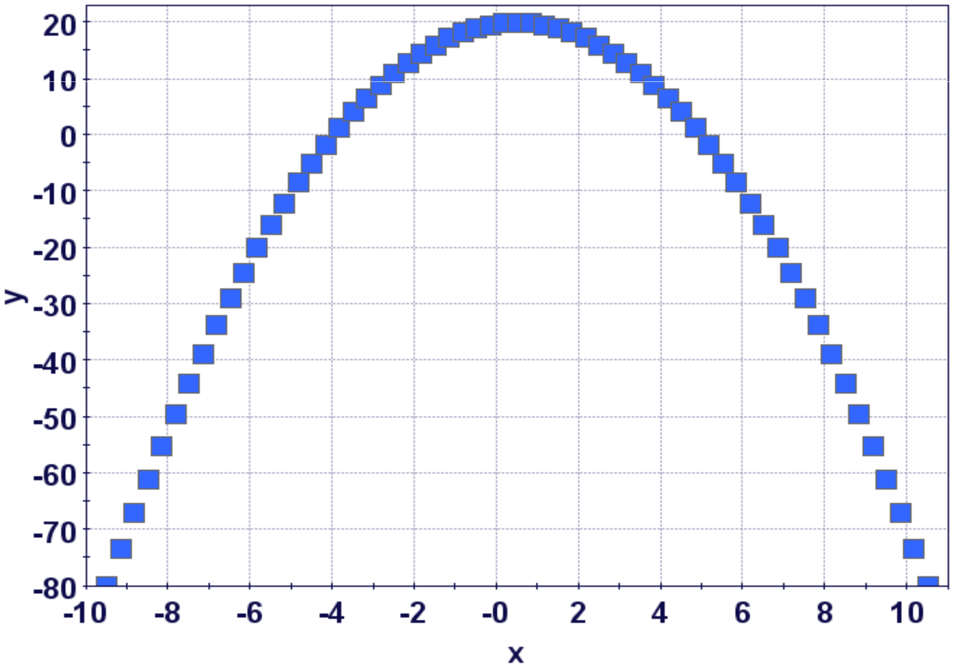
- cost
- weight
- etc...

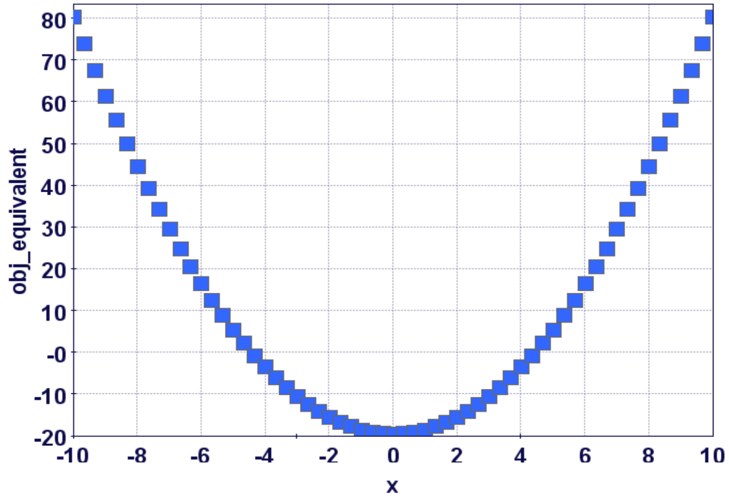
Note : A MAX problem can always be transformed into a MIN problem.

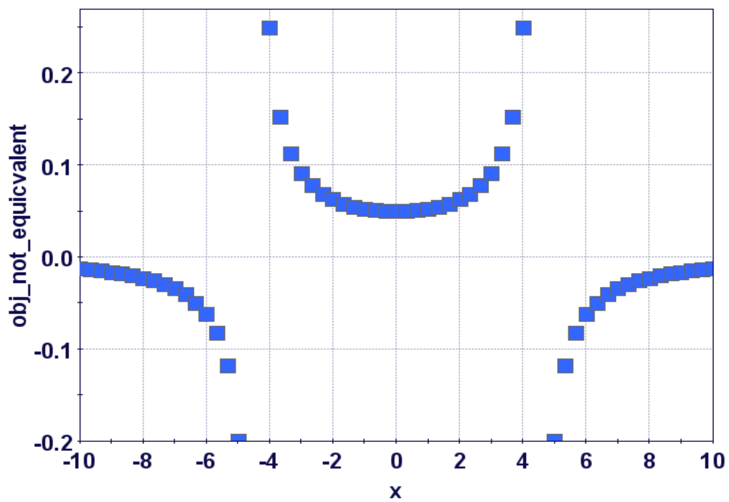
Objectives

In order to transform a MAX into a MIN:

- Fnew(x) = - F(x) equivalent
- Fnew(x) = 1/F(x) NOT equivalent







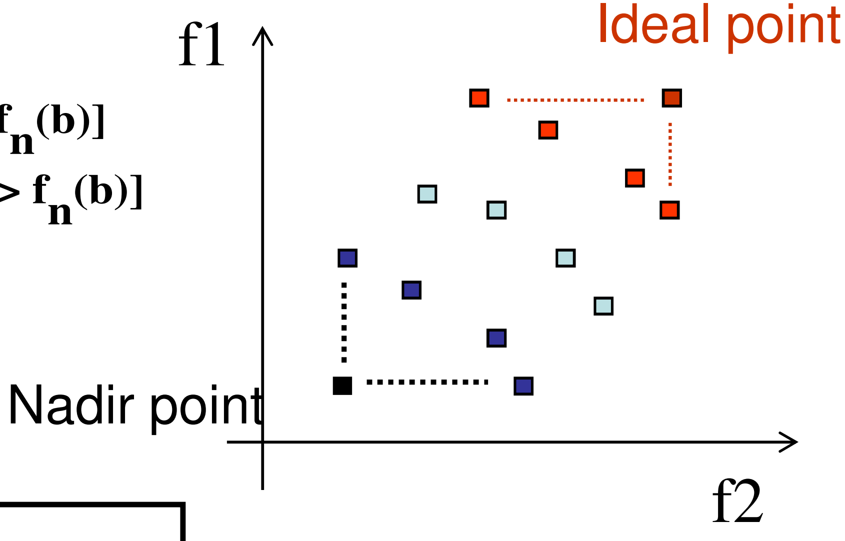
Pareto dominance

- Pareto Dominance:
- Design a **dominates** Design b if:
  - [f1(a) >= f1(b) and f2(a) >= f2(b)...and fn(a) >= fn(b)]
  - **and** [f1(a) > f1(b) or f2(a) > f2(b)...or fn(a) > fn(b)]
- In the Pareto frontier none of the components can be improved without deterioration of at least one of the other component.
- Pareto dominance for one objective coincides with a classical optimization approach
- Pareto dominance defines a group of efficient solutions: in case of n objectives, the group of efficient solutions contains at Max $\infty(n-1)$ points

Pareto dominance

a *dominates* b if and only if:

$$[f_1(a) \geq f_1(b) \textit{ and } f_2(a) \geq f_2(b) \ldots \textit{and } f_n(a) \geq f_n(b)]$$
$$\textit{and } [f_1(a) > f_1(b) \text{ or } f_2(a) > f_2(b) \ldots \text{or } f_n(a) > f_n(b)]$$

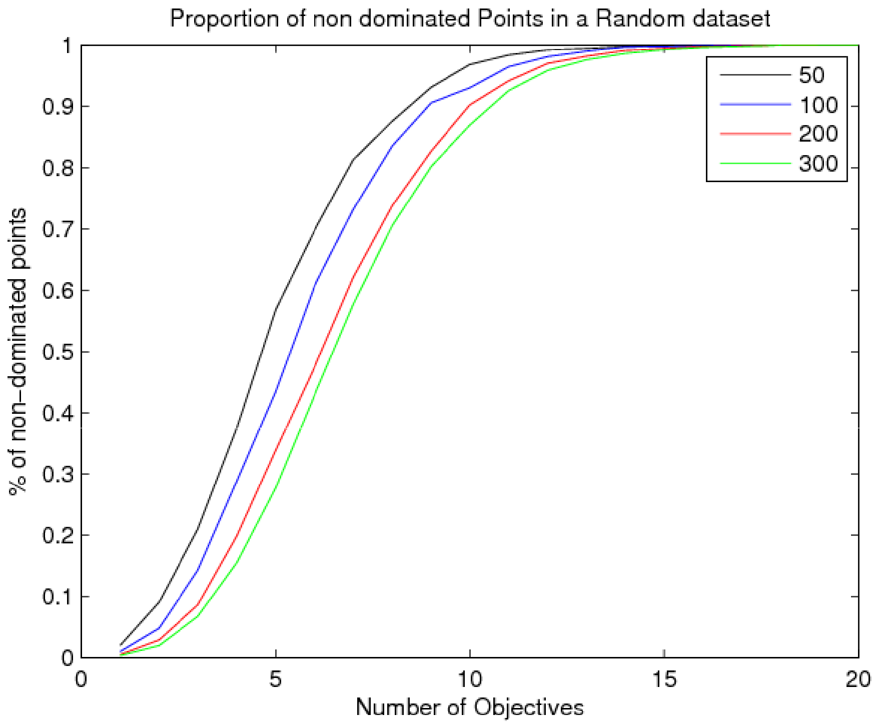


Red dots are the efficient solutions

The optimization algorithm must look for many, not one solution

The designers must take decisions

Pareto Dominated Points



- Rapidly decreasing probability of having a dominated solution in a randomly generated dataset
- Rapidly increasing search effort for when the number of objective is large
- Fortunately, in real-case applications the number of dimensions can collapse

$$r = \frac{\sum_{k=1}^{m} \frac{(-1)^{k+1}}{k^{n-1}} \binom{m}{k}}{m}$$

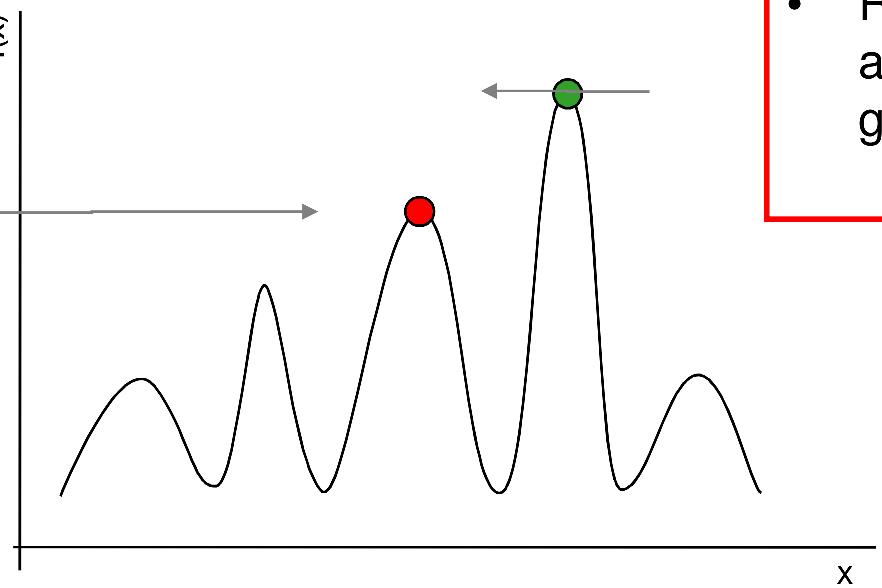
Where m is number of points and n is number of objectives

Robustness

Robustness:

The robustness of an optimization algorithm is the ability to reach the absolute extreme of the objective function.

- Non-Robust algorithms get stuck in local extremes

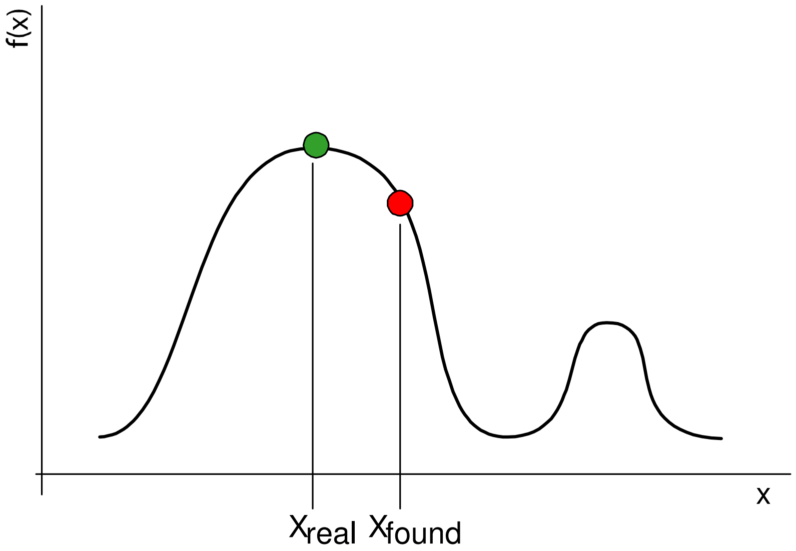


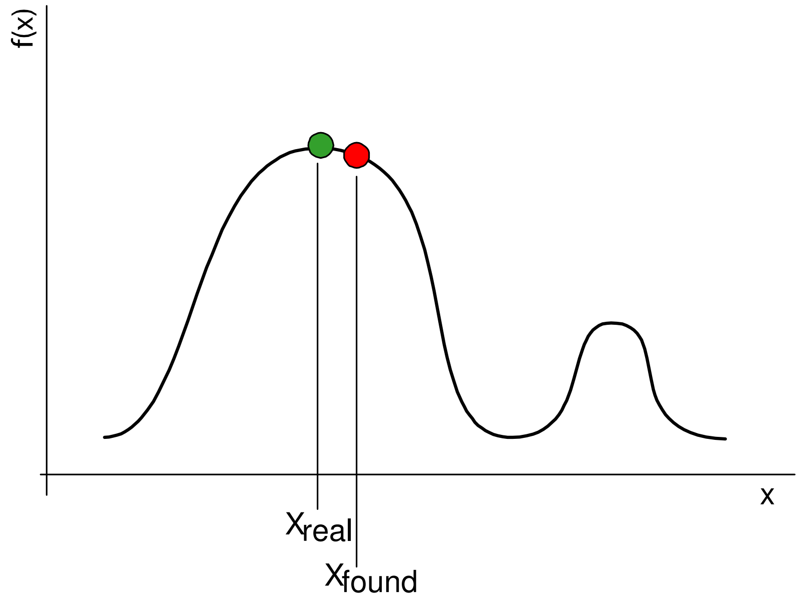
- Robust algorithms reach global extremes

Accuracy

Accuracy:

The accuracy measures the capability of the optimization algorithm to find the function's extreme.





Constraints

Constraint:

Constraints are the quantities imposed to the project, i.e. restrictions and limits that the designer must meet due to norms, functionalities, etc. They define a feasible region.

- **General constraints**
  - Max admissible stress
  - Max deformation
  - Max acceleration
  - min performance
- **Constraints on variables**
  - total volume
  - thickness
  - explicit function of the variables

Weighted Function

Weighted Function:

- n objectives can be added in a single objective using weights:
  - F(x) = w1*Obj1+w2*Obj2+wj3*Obj3...

- **Pro:**
  - simple formulation
- Cons:
  - weights are problem-dependent and must be empirically defined
  - weights are connected to objectives values and might lose significance for different objectives values

Why is the Weighting Method Ineffective

- Although this type of scalarization is widely used in many practical problems, it has a serious drawback: it cannot provide solutions for non-convex cases

- Depending on the structure of the problem, the linearly weighted sum can not necessarily provide a solution that the Decision Maker (DM) desires

- The DMs tend to misunderstand that a desirable solution can be obtained by adjusting the weights but there is no positive correlation between the weights and the value of functions

Why is the Weighting Method Ineffective (Example)

$$\min_x \left(y_1 = f_1(x), y_2 = f_2(x), y_3 = f_3(x)\right)$$

$$\text{s.t.} \sum_{i=1,2,3} (y_i - 1)^2 \le 1$$

Suppose the DM want to reduce more y_1 and even a bit y_2

The minimum of the linearly weighted sum with all the weights equal to 1 is given by:

$$(y_1, y_2, y_3) = \left(1 - 1/\sqrt{3}, 1 - 1/\sqrt{3}, 1 - 1/\sqrt{3}\right)$$

The DM changes the weights:

$$(\omega_1, \omega_2, \omega_3) = (10, 2, 1)$$

$$(y_1, y_2, y_3) = \left(1 - 10/\sqrt{105}, 1 - 2/\sqrt{105}, 1 - 1/\sqrt{105}\right)$$

The value of y2 is worse than before, despite the weights given by the DM

Why is the Weighting Method Ineffective (Example)

- Someone might suspect that this is due to a missing normalization of the weights!
- Normalization of the weights do not solve the problem
- It is usually very difficult to adjust the weights to obtain a solution as the DM wants.

Utility functions

Utility Functions:

- K objectives can be combined into a unique monotone function using the preference relations:

$$U : R^k \rightarrow R$$

$$\text{such that} \quad U(\overline{z_1}) > U(\overline{z_2})$$

$$\text{when} \quad \overline{z_1} \succ \overline{z_2}$$

- Correct formulation of the subjective importance of the objectives

- Not simple formulation

Multi-Objective Optimization

Example

For more information visit:
www.esteco.com or send an e-mail to:
modeFRONTIER@esteco.com

modeFRONTIER
the multi-objective optimization and design environment

Maximize a Mathematical function

Maximize:

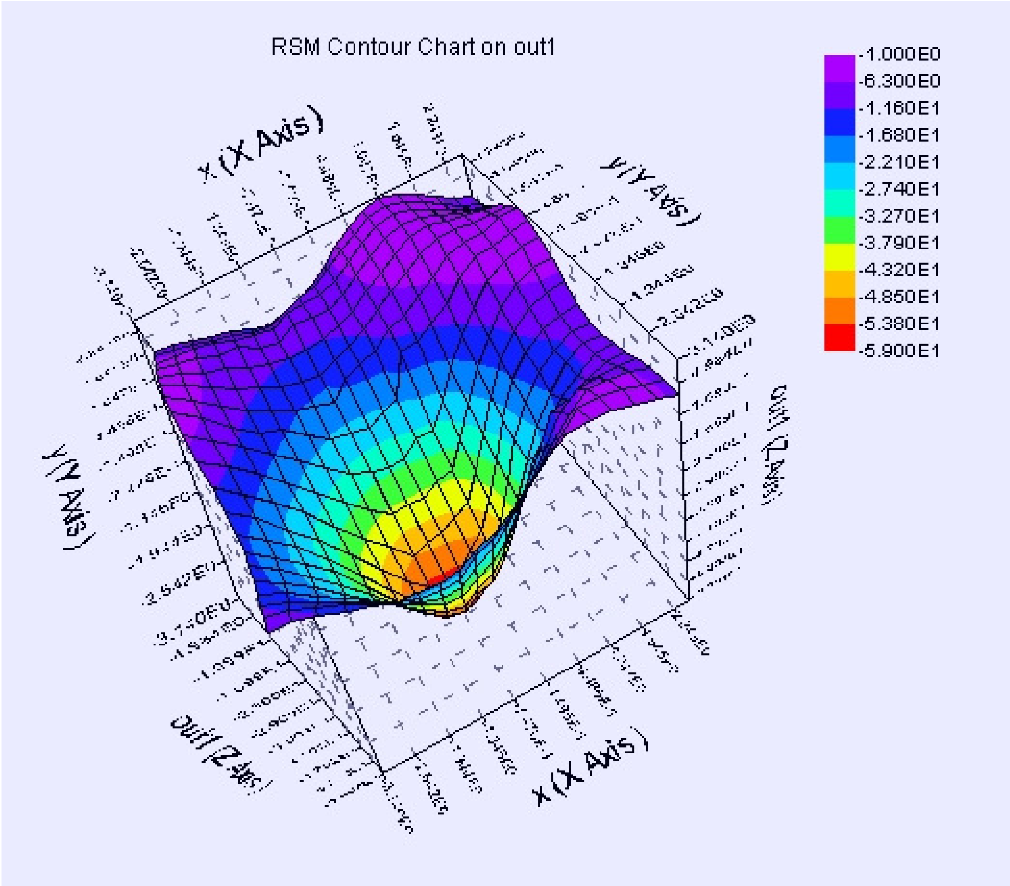
$$F_1(x,y) = -[1 + (A_1 + B_1)^2 + (A_2 + B_2)^2]$$

$$A_i = \sum_{j=1}^{2} (a_{i,j} \cdot sin(\alpha_j) + b_{i,j} \cdot \cos(\alpha_j))$$

$$B_i = \sum_{j=1}^{2} (a_{i,j} \cdot sin(\beta_j) + b_{i,j} \cdot \cos(\beta_j))$$

$$a = \begin{bmatrix} 0.5 & 1.0 \\ 1.5 & 2.0 \end{bmatrix} \qquad b = \begin{bmatrix} -2.0 & -1.5 \\ -1.0 & -0.5 \end{bmatrix} \qquad \alpha = [1.0 \quad 2.0]$$

$$\beta = (x, y) \quad \in \quad [-\pi, \pi]$$

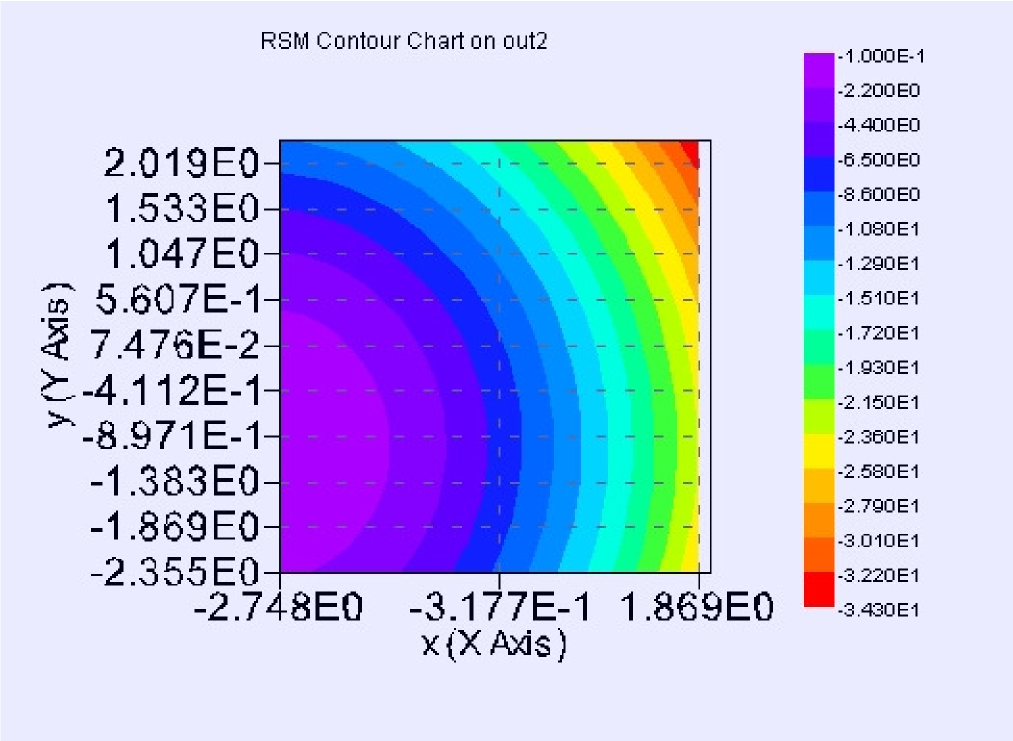


Mathematical functions

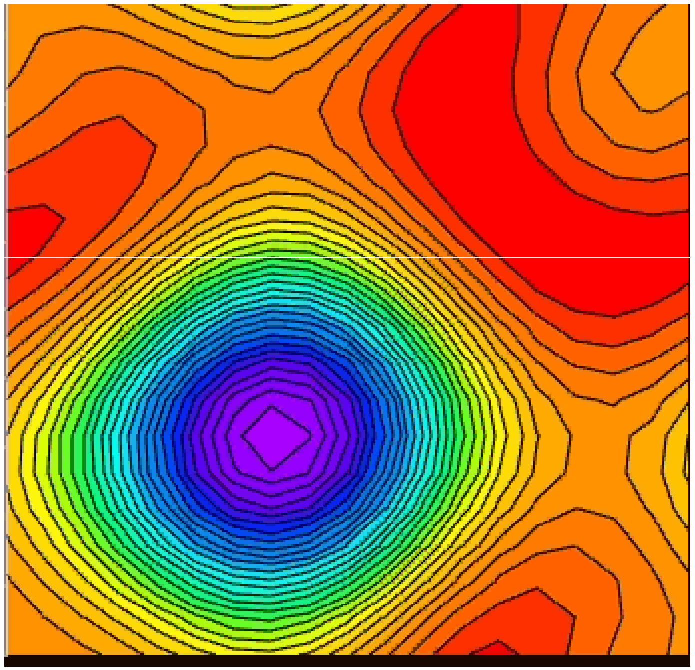
Maximize:

$$F_2(x,y) = -[(x+3)^2 + (y+1)^2]$$

$$x, y \in [-\pi, \pi]$$



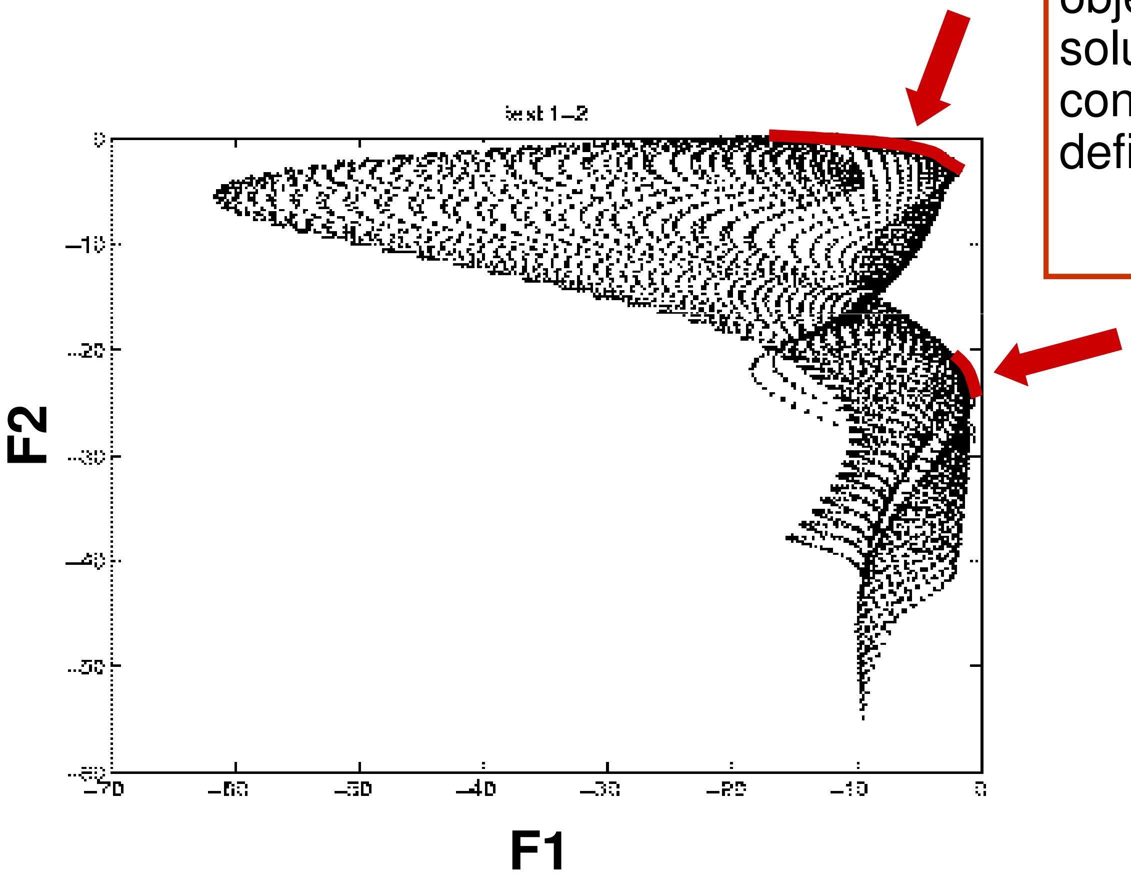
Weighted Sum



Weighted Sum:

- F= (1-k)*F1+k*F2
- The parameter k is varied from 0 to 1 with a step of 0.1
- The weighted sum goes progressively from F1 to F2
- The red zones indicate higher values for the weighted sum

Pareto Frontier



Two variables, two objectives, infinite **efficient** solutions in two regions not connected in the variables definition domain.

General Remarks

Facing a design problem:

- Rarely there is a clearly identified and unique objective

- There is a vague distinction between constraints and objectives

- Even if algorithms and numerical optimization theories exists in the academic world since many years, the practical impact until today was negligible and limited to very specific applications:

    - It is necessary to extend the concept of mathematical optimization to several objectives
    - It is necessary to have “robust” tools to explore the design configuration space

Conclusion

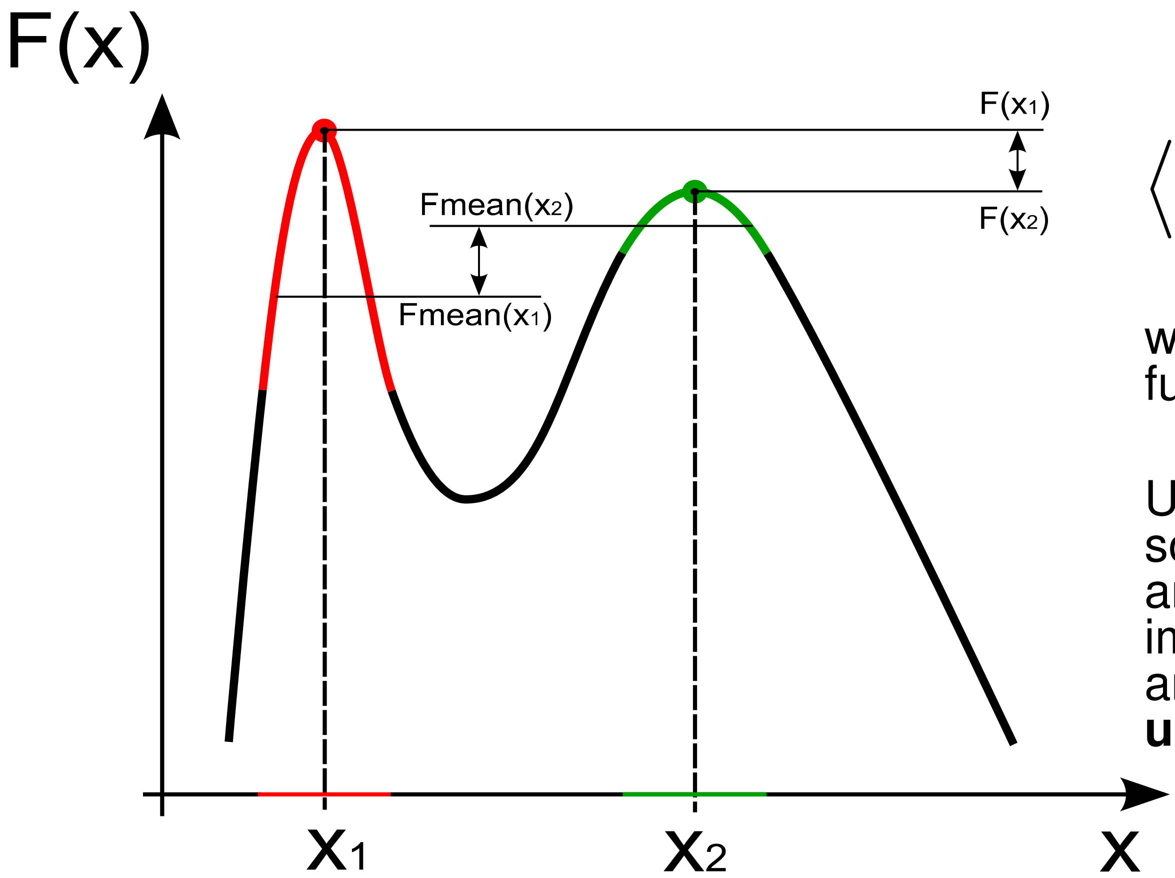
Optimization problems with more than one objective need ROBUST algorithms capable of producing several different solutions to the problem both in the case of Pareto dominance approach as well as with weighted sum or utility function approach.

Robust Design Optimization

For more information visit:
www.esteco.com or send an e-mail to:
modeFRONTIER@esteco.com

Robust Design

Maximisation problem where the **expectation value** should be computed.



$$\langle F(x) \rangle = \int F(x) P(x)$$

where P(x) is the probability density function .

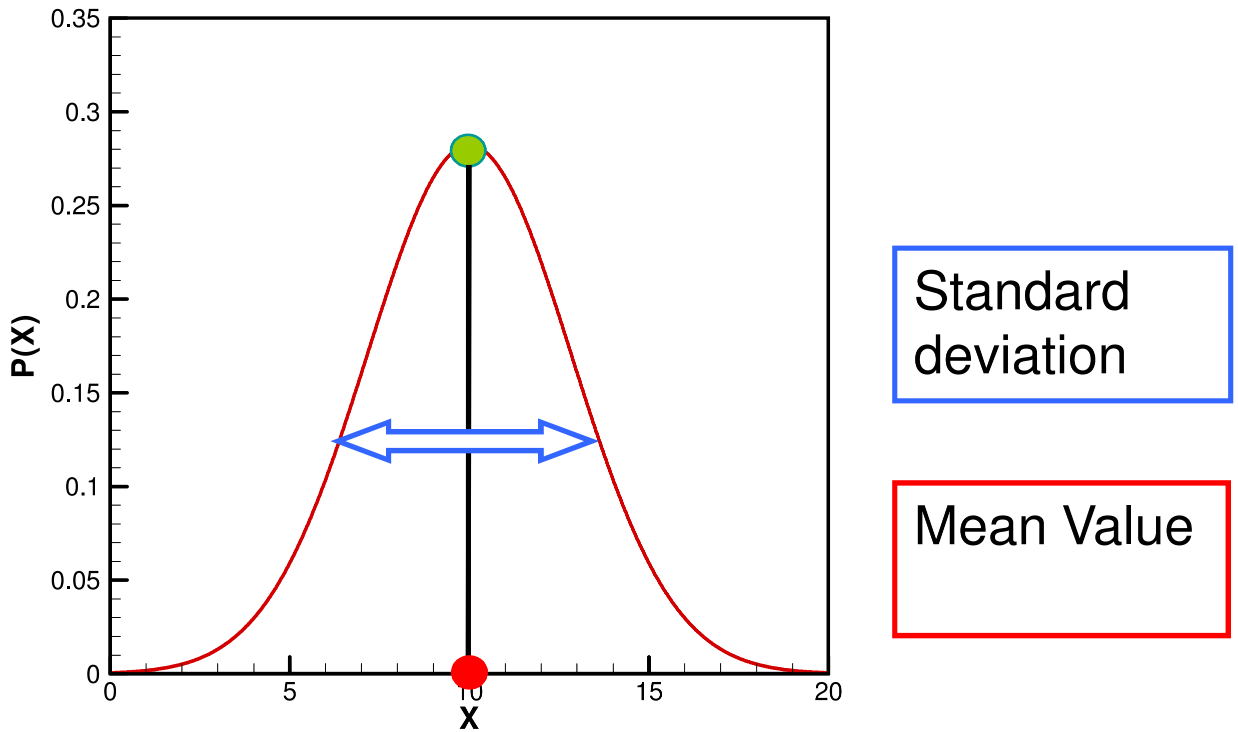
Unfortunately, most of the times solving the uncertainty quantification analytically may be difficult (or even impossible) for complex functions and, moreover, the function can be **unknown.**

Robust Design

In many real world optimization problems, the design parameters are not fixed, normally are known with a **mean value** and a **standard deviation.**

Example:

X=10 mm $\pm$ σ (=1.25 mm)



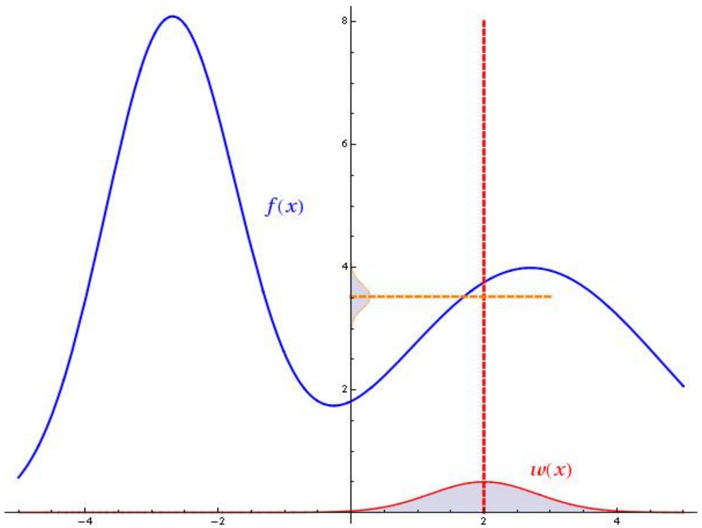
The Problem

- The first step is to find the mean and the standard deviation

- The optimization problem may be re-written as:

$$\min_{x \in \mathbf{R}^n} \left(\mu_{f_1(x)} + \sigma_{f_1(x)}, \ldots, \mu_{f_k(x)} + \sigma_{f_k(x)} \right)$$

- And this problem may be solved as a multiobjective optimization problem



Robust Design

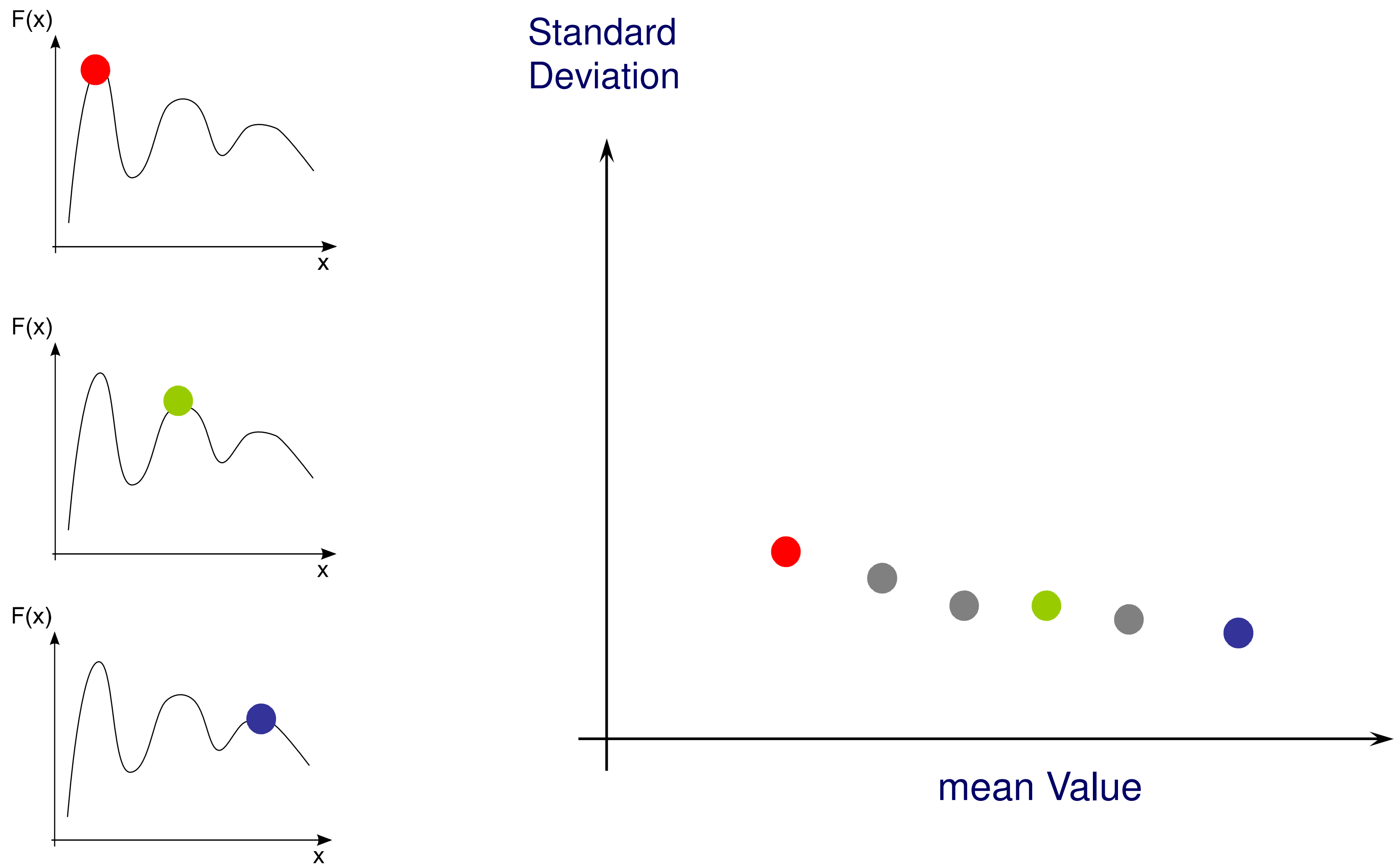
- The **BEST ROBUST** solution could not be always identified with the **BEST GLOBAL** solution.

- For these reasons we have to introduce a different objective for this kind of problems:
  - **Maximize the average value** of the function considering the variables distribution;
  - **Keep under control the standard deviation**.

- **OR for example**

$$\max_{x}\left(\mu_{f(x)} - k\sigma_{f(x)}\right)$$

Robust Design



References

- S. Rao, (1996), "Engineering Optimisation" Theory and Practice", J.Wiley and Sons
- D.Quagliarella, J.Periaux, C.Poloni, G.Winter, (1998), "Genetic Algorithms and Evolution Strategies in Engineering and Computer Science", J.Wiley and Sons
- K.Miettinen, (1999), "Nonlinear Multiobjective Optimisation", Kuwler Academic Publishers, ISBN 0-7923-8278-1
- P. Sen and J.B.Yang, (1998), "Multiple Criteria Decision Support in Engineering Design", Springer-Verlag, London , ISBN 3-540199-32-2.
- Geoge W.Cobb, (1997), "Introduction to Design and Analysis of Experiments", Springer-Verlag, ISBN 0-387-94607-1