

# Numerical optimization

Mines Nancy – Fall 2024

session 1 – general introduction

**Lecturer:** Julien Flamant (CNRS, CRAN)

✉ [julien.flamant@univ-lorraine.fr](mailto:julien.flamant@univ-lorraine.fr)

📍 Office 425, FST 1er cycle

**Course material:**

🌐 [arche.univ-lorraine.fr/course/view.php?id=74098](http://arche.univ-lorraine.fr/course/view.php?id=74098)

⌚ [github.com/jflamant/mines-nancy-fall24-optimization](https://github.com/jflamant/mines-nancy-fall24-optimization)

# Outline of the course

36h of classes, including lectures, exercices and lab work (Python)

Course material in english, lecture given in french or english

Main results will be on lecture slides, but detailed computations, proofs and exercises solution will not, in general. [Take notes!](#)

Prerequisites: elementary differential calculus and linear algebra

## Evaluation

- intermediate exams (40%)
- final exam (60%): January 7th, 2025

**Lecturers** Julien Flamant (CNRS) and Christophe Zhang (INRIA)

**Contact info** [julien.flamant@univ-lorraine.fr](mailto:julien.flamant@univ-lorraine.fr)

# What to expect from this course

in short: an introduction to numerical optimization

## Objectives

- learn the basic language of optimization;
- characterize objective functions and critical points;
- solve least squares problems;
- learn about classical first and second-order descent algorithms;
- first principles and algorithms in constrained optimization.

hands-on approach with [Python](#) notebooks

# Some useful references

## Books

- D. Bertsekas (1999). *Nonlinear programming*. Second edition. Belmont, MA: Athena Scientific
- J. Nocedal and S. J. Wright (1999). *Numerical optimization*. Springer
- S. P. Boyd and L. Vandenberghe (2004). *Convex optimization*. Cambridge university press
- A. Beck (2017). *First-order methods in optimization*. SIAM

## Online material (include lecture slides and notebooks)

- MIT OpenCourseWare | Optimization (Boyd and Bertsekas courses)
- Numerical tours | Optimization

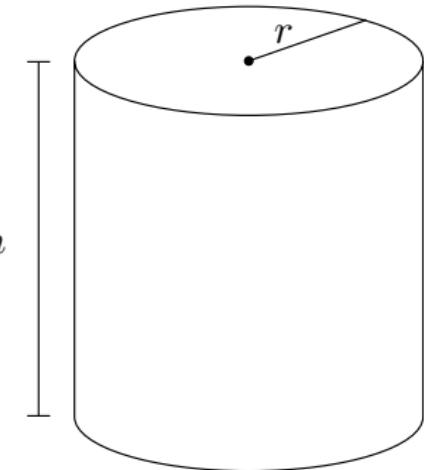
# Outline

- ① A first example
- ② More examples
- ③ Optimization: vocabulary and nomenclature
- ④ Some standard optimization problems
- ⑤ Recap

# A first example

A can producer wants to **optimize** his consumption of raw material.

The only **constraint** is that their volume  $V = 1\text{L}$  is fixed.



How to formulate this shape optimization problem?

quantity of raw material  $\equiv$  area of cylinder  $\times$  constant thickness

→ how to **minimize** the area  $A$  of a cylinder with given volume  $V$ ?

## Formulation of the optimization problem

Expression of the area  $A$  (**objective function or criterion**)

$$A(r, h) = A_{\text{cylinder}}(r, h) + A_{\text{lids}}(r, h) = 2\pi rh + 2\pi r^2$$

# Formulation of the optimization problem

Expression of the area  $A$  (**objective function or criterion**)

$$A(r, h) = A_{\text{cylinder}}(r, h) + A_{\text{lids}}(r, h) = 2\pi rh + 2\pi r^2$$

The dimensions  $r$  and  $h$  are the **variables** of the optimization problem

$r$  and  $h$  must be non-negative  $\rightarrow$  **constraint**  $r \geq 0, h \geq 0$

# Formulation of the optimization problem

Expression of the area  $A$  (**objective function or criterion**)

$$A(r, h) = A_{\text{cylinder}}(r, h) + A_{\text{lids}}(r, h) = 2\pi rh + 2\pi r^2$$

The dimensions  $r$  and  $h$  are the **variables** of the optimization problem

$r$  and  $h$  must be non-negative  $\rightarrow$  **constraint**  $r \geq 0, h \geq 0$

The condition of constant volume is also a **constraint**.

$$V(r, h) = \pi r^2 h = V_0$$

# Formulation of the optimization problem

Expression of the area  $A$  (**objective function or criterion**)

$$A(r, h) = A_{\text{cylinder}}(r, h) + A_{\text{lids}}(r, h) = 2\pi rh + 2\pi r^2$$

The dimensions  $r$  and  $h$  are the **variables** of the optimization problem

$r$  and  $h$  must be non-negative  $\rightarrow$  **constraint**  $r \geq 0, h \geq 0$

The condition of constant volume is also a **constraint**.

$$V(r, h) = \pi r^2 h = V_0$$

The **constrained optimization problem** is finally written as

Find  $(\hat{r}, \hat{h})$  such that  $A(\hat{r}, \hat{h})$  is minimal and  $\begin{cases} r \geq 0 \text{ and } h \geq 0 \\ \pi r^2 h = V_0 \end{cases}$

## Simplification of the problem

We notice that  $h$  and  $r$  are related by the constraint  $V_0 = \pi r^2 h$ .

→ The area  $A(r, h)$  can be rewritten as a function of  $r$  only.

$$h(r) = \frac{V_0}{\pi r^2} \quad (h \geq 0 \quad \text{OK!})$$

## Simplification of the problem

We notice that  $h$  and  $r$  are related by the constraint  $V_0 = \pi r^2 h$ .

→ The area  $A(r, h)$  can be rewritten as a function of  $r$  only.

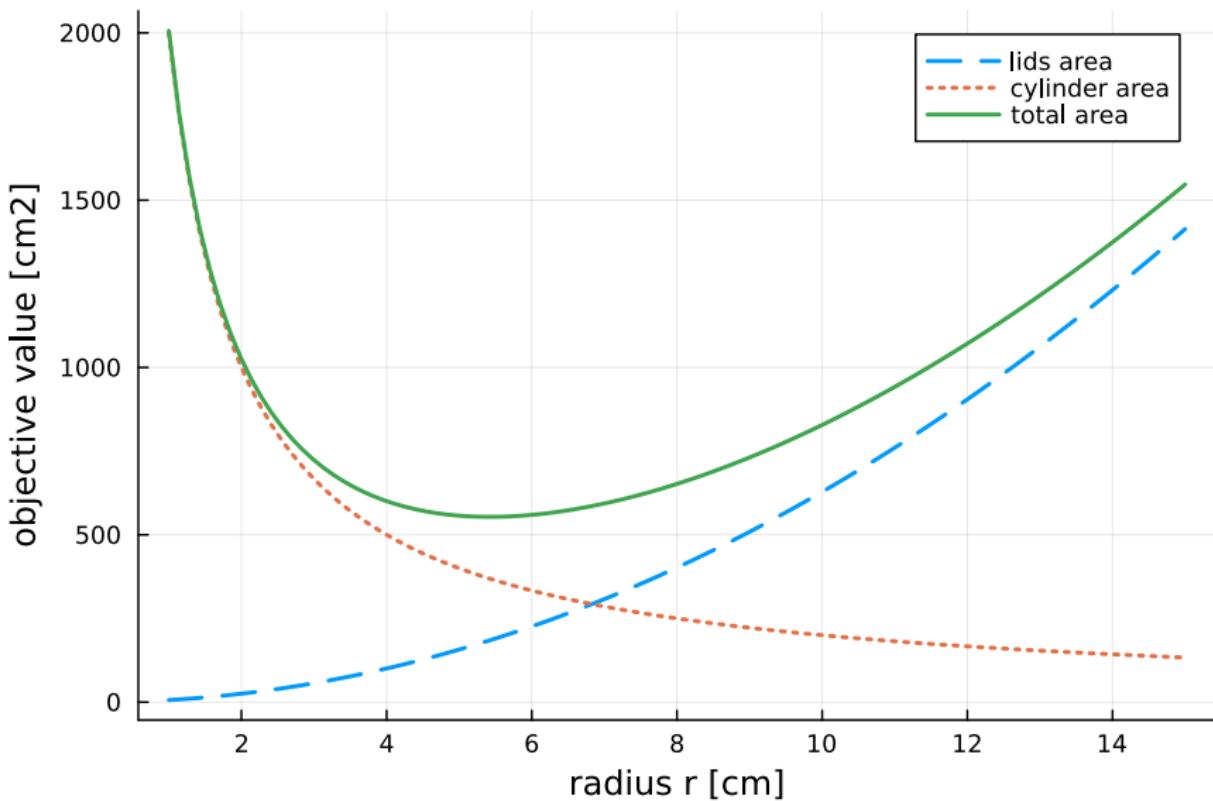
$$h(r) = \frac{V_0}{\pi r^2} \quad (h \geq 0 \quad \text{OK!})$$

By substituting this into the expression for  $A(r, h)$ , we then obtain

$$\begin{aligned} A(r, h) &= 2\pi rh + 2\pi r^2 \\ &= 2\pi r \frac{V_0}{\pi r^2} + 2\pi r^2 \\ &= \frac{2V_0}{r} + 2\pi r^2 \\ &:= \tilde{A}(r) \end{aligned}$$

The volume constraint is directly taken into account in the new objective function  $\tilde{A}(r)$ . We consider the constraint  $r \geq 0$  when searching for the minimum of  $\tilde{A}(r)$  over  $\mathbb{R}^+$ .

# Objective function of the problem ( $V_0 = 1\text{L}$ )



## Explicit solution

The derivative of the objective function vanishes at its minimum.

$$\frac{d\tilde{A}(r)}{dr} = 4\pi r - \frac{2V_0}{r^2}$$

We solve

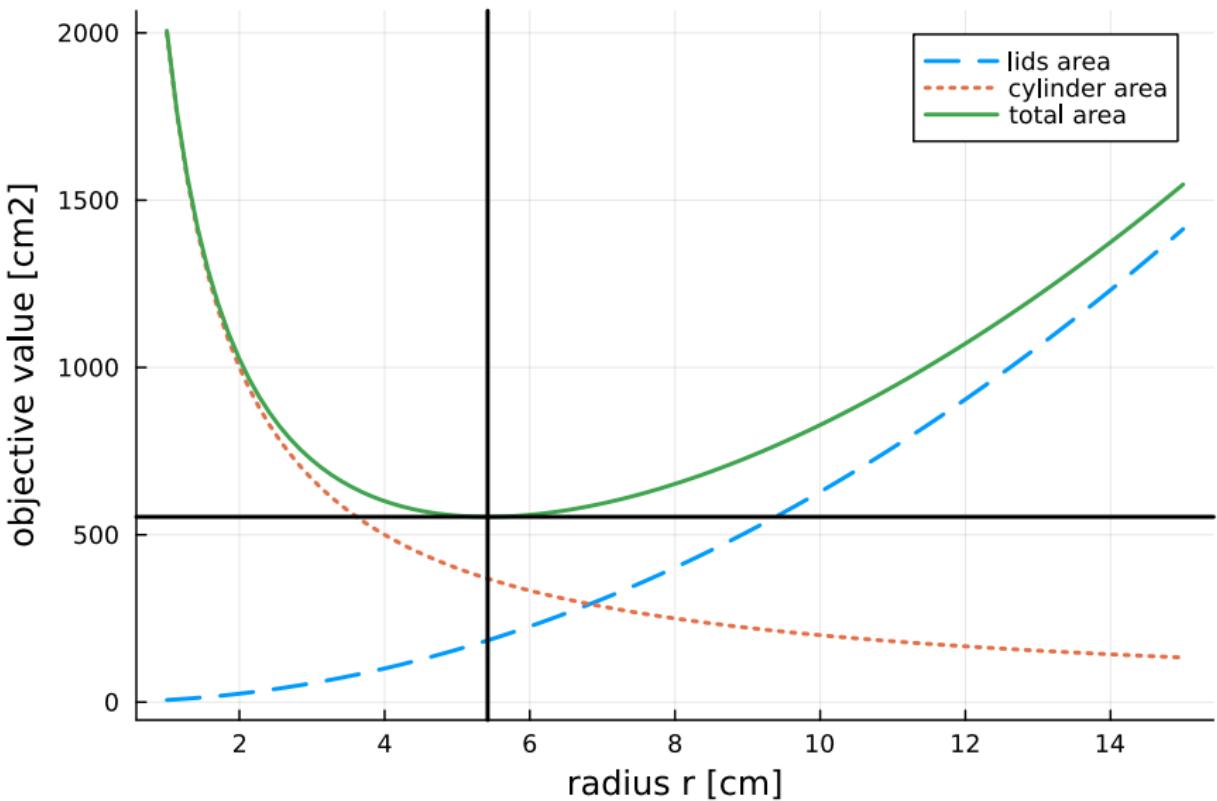
$$\begin{aligned}\left. \frac{d\tilde{A}(r)}{dr} \right|_{r=\hat{r}} &= 0 \Leftrightarrow 4\pi\hat{r} = \frac{2V_0}{\hat{r}^2} \\ &\Leftrightarrow \hat{r} = \left[ \frac{V_0}{2\pi} \right]^{1/3}\end{aligned}$$

In summary, the solutions to the optimization problem are

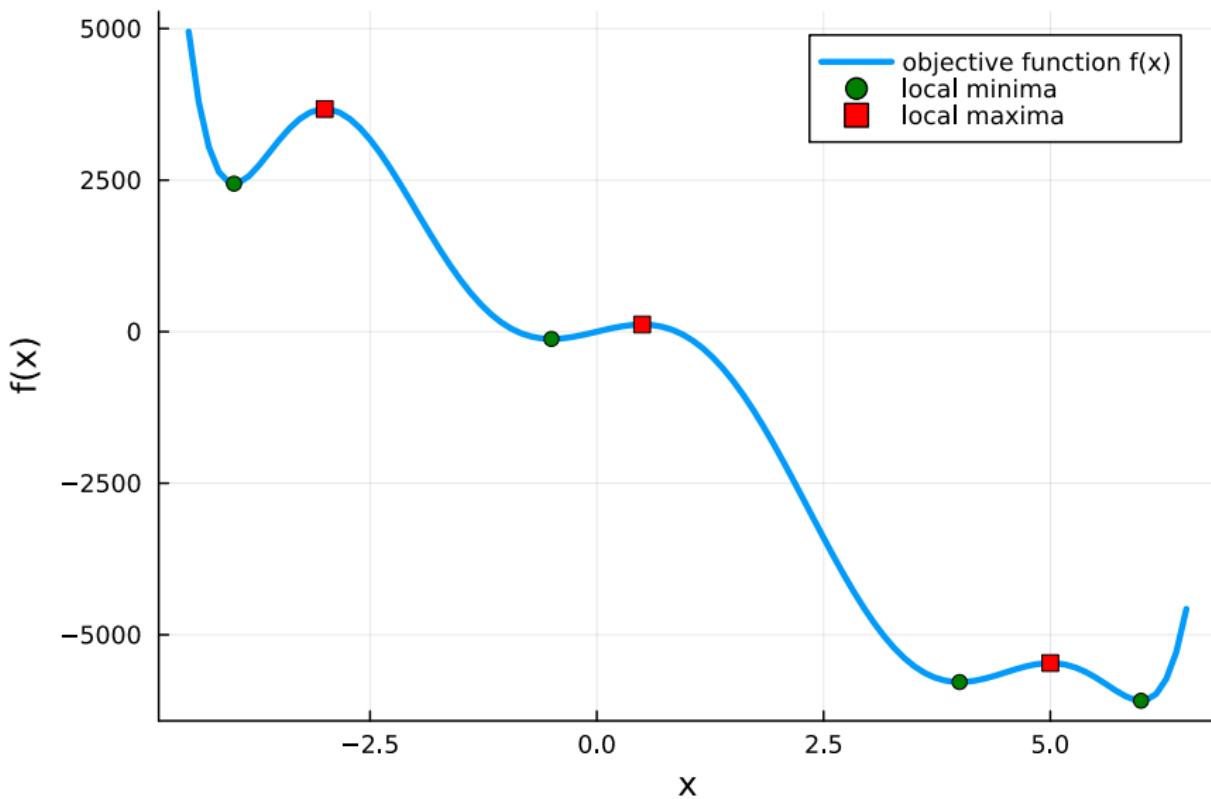
$$\hat{r} = \left[ \frac{V_0}{2\pi} \right]^{1/3} \quad \hat{h} = \frac{V_0}{\pi\hat{r}^2} = 2\hat{r}$$

In this simple case, the solution is explicit and does not require a resolution algorithm → this is not the case in general.

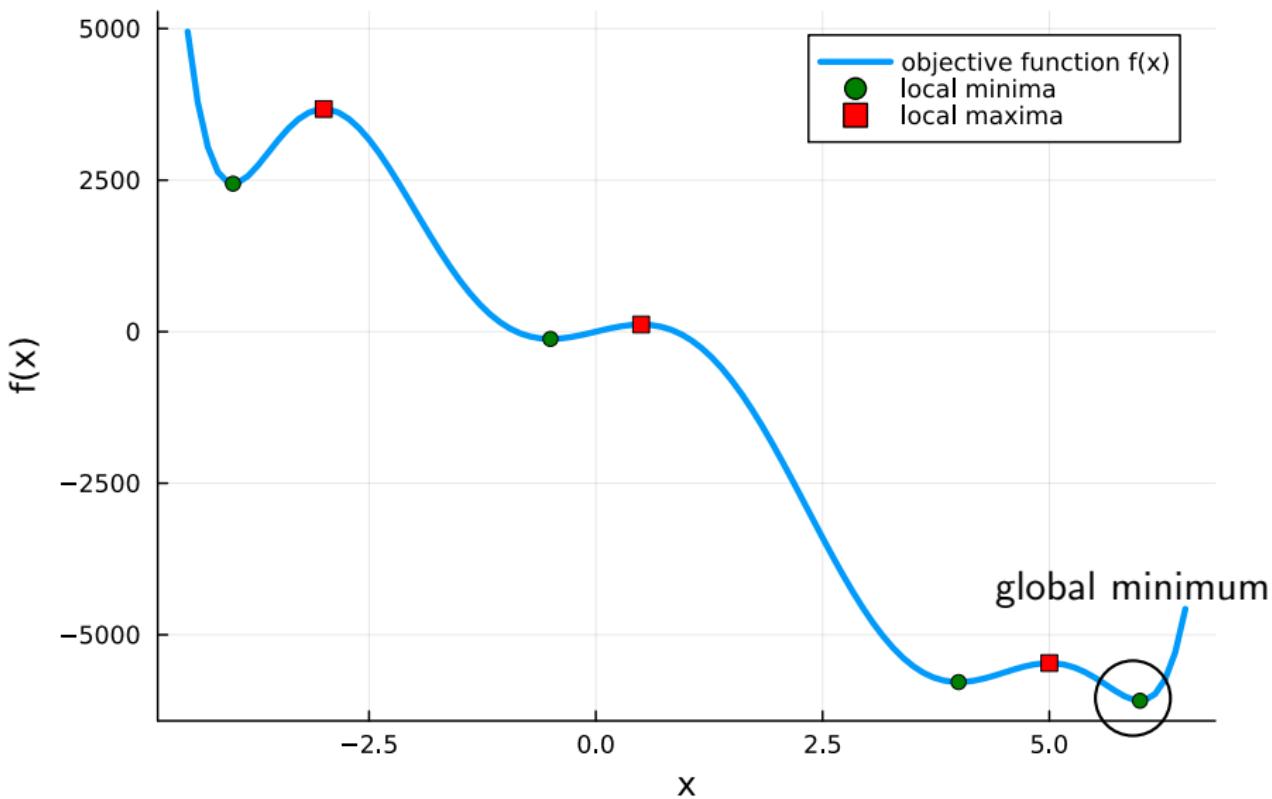
# Objective function of the problem ( $V_0 = 1\text{L}$ ) and solution



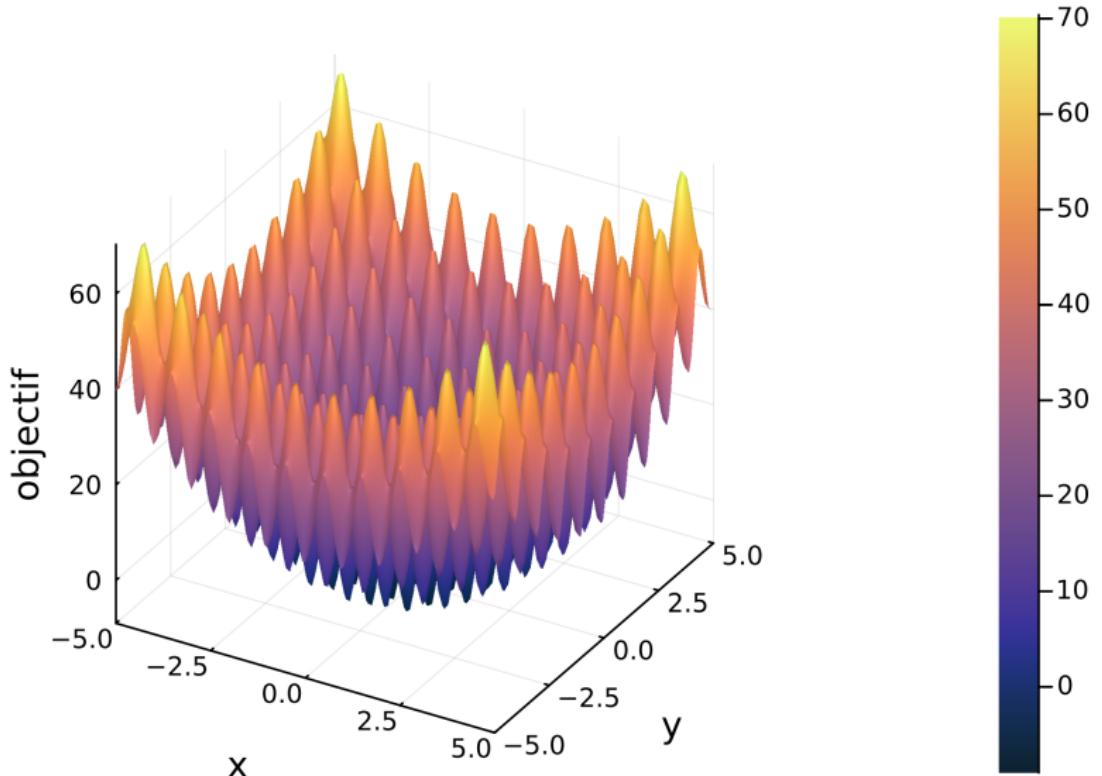
It's not always that simple... example in 1D



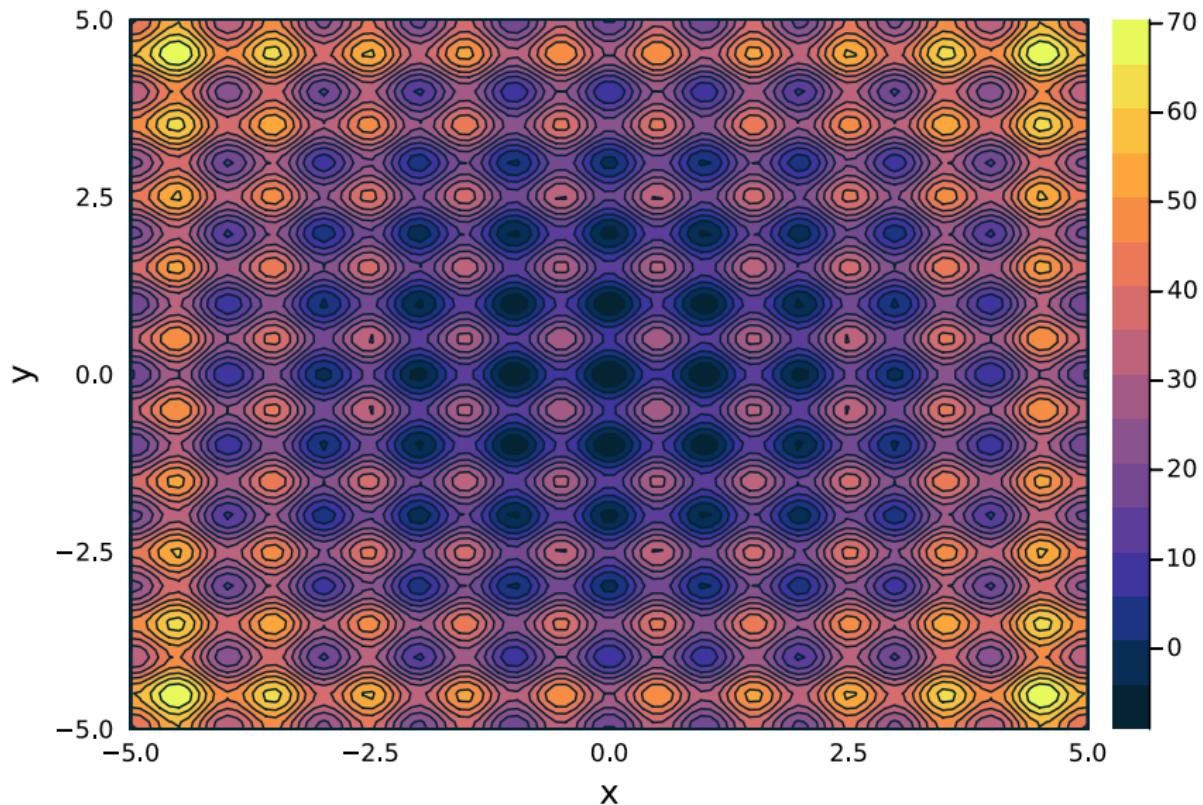
It's not always that simple... example in 1D



It's not always that simple... example in 2D



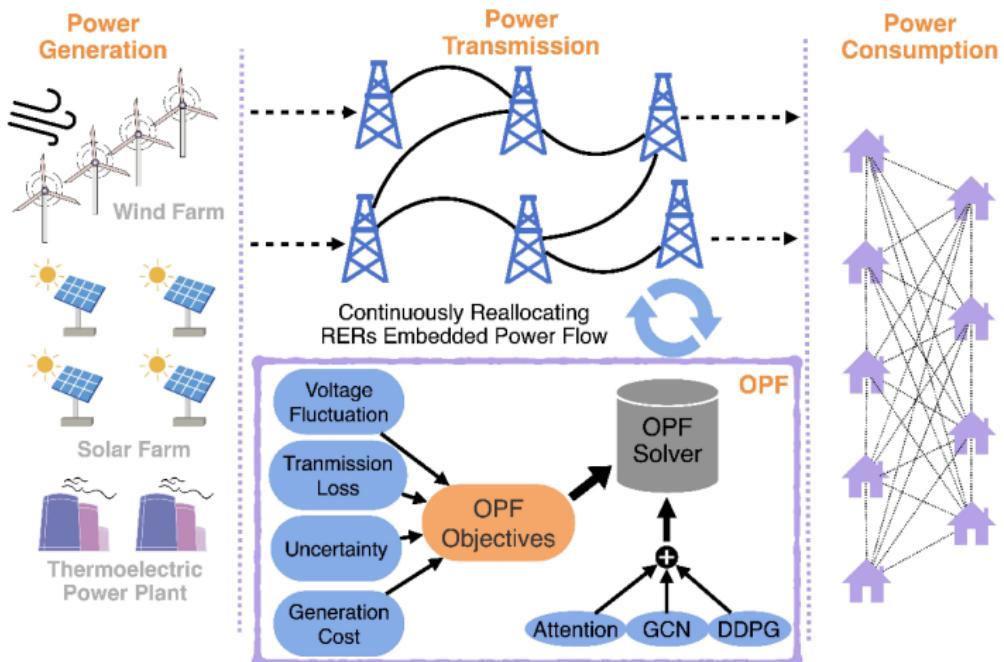
It's not always that simple... example in 2D



# Outline

- ① A first example
- ② More examples
- ③ Optimization: vocabulary and nomenclature
- ④ Some standard optimization problems
- ⑤ Recap

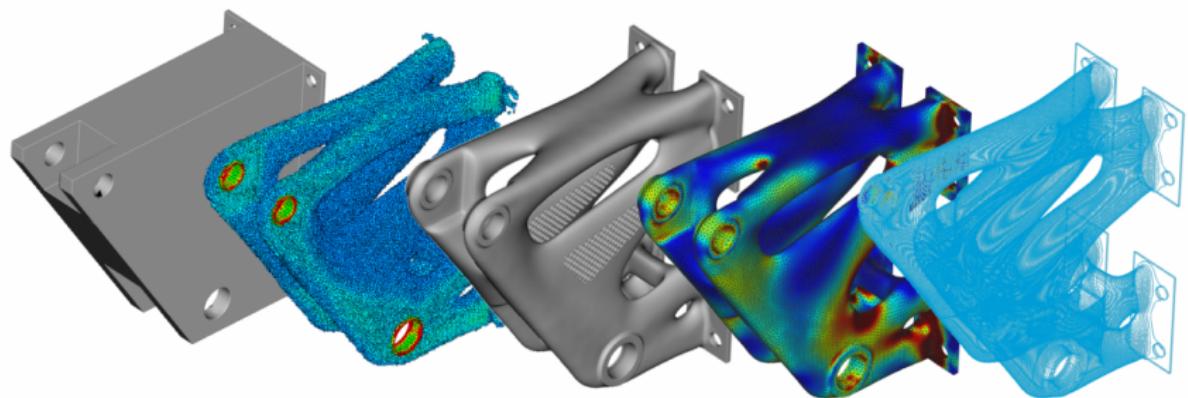
# Energy chain optimization



often a multi-objective problem: maximize system efficiency + minimize costs, etc.

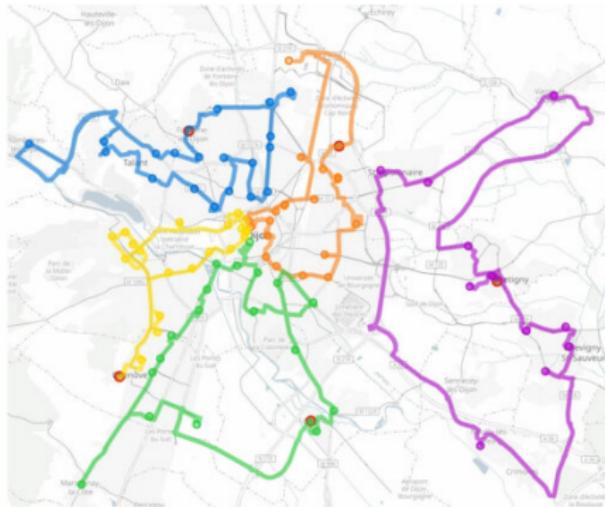
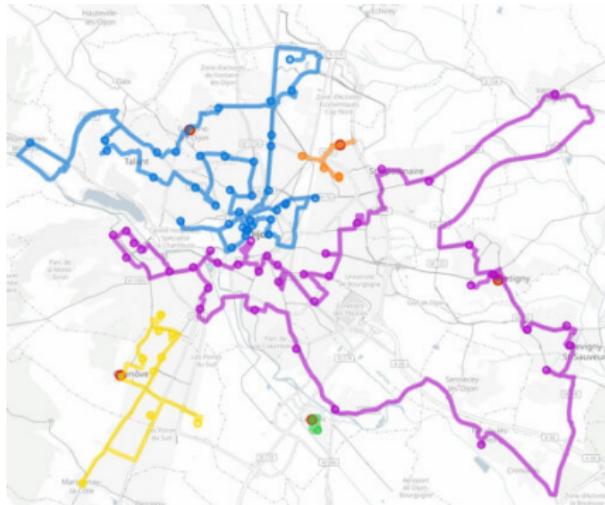
variables of different types: discrete + continuous

# Shape optimization



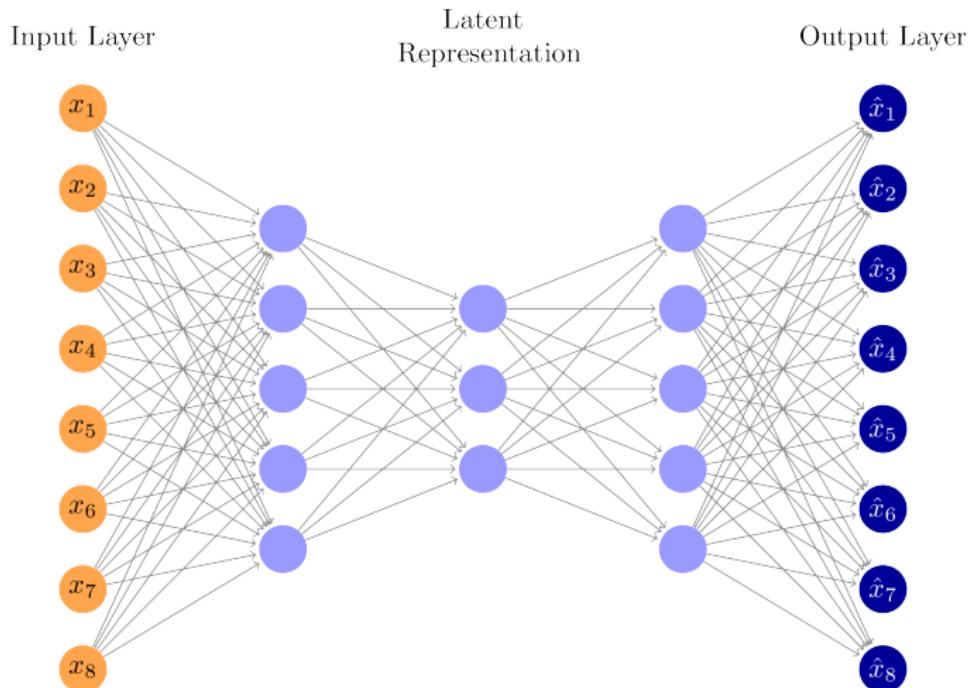
objective: strongly non-linear, based on physics/mechanics  
variables: several billion positions of 3D mesh points

# Route or path optimization



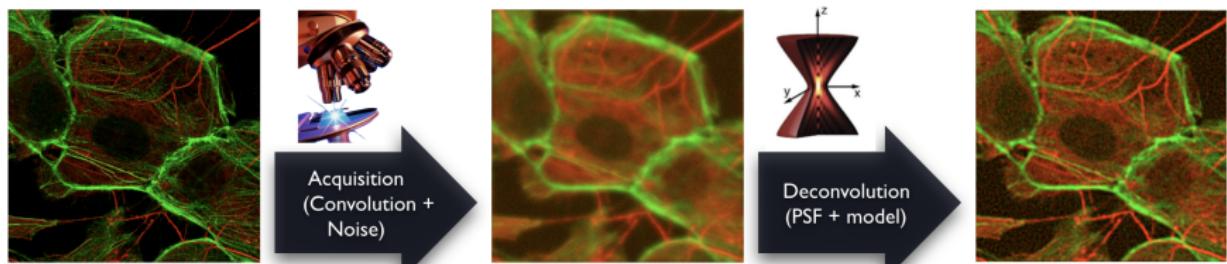
objective: minimize the distance/time of each path/journey/yield/etc.,  
often multi-objective  
generally discrete variables

# Learning and neural networks



objective: minimize the error between input and output  
 $\sim 10^6$  to  $10^9$  optimization variables

# Image processing: deconvolution and denoising



Inverse problem: reconstruct an image  $X$  from the measurements

$$Y = \text{PSF} * X + \text{noise}$$

Objective: minimize the error between the measurements and the direct model

image  $X$  with several megapixels  $\rightarrow$  as many optimization variables

# Outline

- ① A first example
- ② More examples
- ③ Optimization: vocabulary and nomenclature
- ④ Some standard optimization problems
- ⑤ Recap

# General form of an optimization problem

$$\min_{\mathbf{x} \in \mathbb{X}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Vocabulary

- $\mathbf{x} = (x_1, x_2, \dots, x_N)^\top \in \mathbb{X}^N$  is the **vector of variables** or **parameters** of the optimization problem

# General form of an optimization problem

$$\min_{\mathbf{x} \in \mathbb{X}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Vocabulary

- $\mathbf{x} = (x_1, x_2, \dots, x_N)^\top \in \mathbb{X}^N$  is the **vector of variables** or **parameters** of the optimization problem
- $N$  is the number of variables or the **dimension** of the problem

# General form of an optimization problem

$$\min_{\mathbf{x} \in \mathbb{X}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Vocabulary

- $\mathbf{x} = (x_1, x_2, \dots, x_N)^\top \in \mathbb{X}^N$  is the **vector of variables** or **parameters** of the optimization problem
- $N$  is the number of variables or the **dimension** of the problem
- the function  $f : \mathbb{X}^N \rightarrow \mathbb{R}$  is the **objective function**, also called the **cost function** or **criterion**

# General form of an optimization problem

$$\min_{\mathbf{x} \in \mathbb{X}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Vocabulary

- $\mathbf{x} = (x_1, x_2, \dots, x_N)^\top \in \mathbb{X}^N$  is the **vector of variables** or **parameters** of the optimization problem
- $N$  is the number of variables or the **dimension** of the problem
- the function  $f : \mathbb{X}^N \rightarrow \mathbb{R}$  is the **objective function**, also called the **cost function** or **criterion**
- the set  $\Omega \subseteq \mathbb{X}^N$  is the **constraint** or **feasible set**. Often,  $\Omega$  takes the form

$$\Omega = \{\mathbf{x} \mid \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$$

i.e., it is defined in terms of *functional constraints*.

# General form of an optimization problem

$$\min_{\mathbf{x} \in \mathbb{X}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Nomenclature

- The space  $\mathbb{X}$  can be:
  - finite or discrete: discrete (or combinatorial) optimization
  - continuous  $\mathbb{X} = \mathbb{R}$ : continuous optimization

# General form of an optimization problem

$$\min_{\mathbf{x} \in \mathbb{X}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Nomenclature

- The space  $\mathbb{X}$  can be:
  - finite or discrete: discrete (or combinatorial) optimization
  - continuous  $\mathbb{X} = \mathbb{R}$ : continuous optimization
- if no constraints,  $\Omega = \mathbb{X}^N$ : **unconstrained optimization**

# General form of an optimization problem

$$\min_{\mathbf{x} \in \mathbb{X}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Nomenclature

- The space  $\mathbb{X}$  can be:
  - finite or discrete: discrete (or combinatorial) optimization
  - continuous  $\mathbb{X} = \mathbb{R}$ : continuous optimization
- if no constraints,  $\Omega = \mathbb{X}^N$ : **unconstrained optimization**
- with constraints  $\Omega \subset \mathbb{X}^N$ : **constrained optimization**

# General form of an optimization problem

$$\min_{\mathbf{x} \in \mathbb{X}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Nomenclature

- The space  $\mathbb{X}$  can be:
  - finite or discrete: discrete (or combinatorial) optimization
  - continuous  $\mathbb{X} = \mathbb{R}$ : continuous optimization
- if no constraints,  $\Omega = \mathbb{X}^N$ : **unconstrained optimization**
- with constraints  $\Omega \subset \mathbb{X}^N$ : **constrained optimization**
- the nature of  $f$  defines specific problems:
  - $f$  linear: linear optimization
  - $f$  quadratic: quadratic optimization
  - $f$  convex and convex constraints: convex optimization
  - $f$  non-differentiable, non-linear, etc.

# General form of an optimization problem

$$\min_{\mathbf{x} \in \mathbb{X}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Nomenclature

- The space  $\mathbb{X}$  can be:
  - finite or discrete: discrete (or combinatorial) optimization
  - continuous  $\mathbb{X} = \mathbb{R}$ : continuous optimization
- if no constraints,  $\Omega = \mathbb{X}^N$ : **unconstrained optimization**
- with constraints  $\Omega \subset \mathbb{X}^N$ : **constrained optimization**
- the nature of  $f$  defines specific problems:
  - $f$  linear: linear optimization
  - $f$  quadratic: quadratic optimization
  - $f$  convex and convex constraints: convex optimization
  - $f$  non-differentiable, non-linear, etc.

each category has its own resolution algorithms

# Scope of this course

continuous optimization with and without constraints

$$\min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

typical assumption:  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  is  $\mathcal{C}^1$  (or at least, differentiable)

# Scope of this course

continuous optimization with and without constraints

$$\min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

typical assumption:  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  is  $\mathcal{C}^1$  (or at least, differentiable)

## Note: minimization vs. maximization

There are also maximization problems (e.g., maximizing the return of a stock portfolio)

$$\max_{\mathbf{x} \in \mathbb{R}^N} g(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

This problem is equivalent to the previous problem for  $f(\mathbf{x}) = -g(\mathbf{x})$ .  
→ In practice, we always reduce the problem to a minimization problem.

# Typology of solutions

$$\min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

solving this optimization problem means finding  $\mathbf{x}^* \in \Omega$  (i.e. satisfying the constraints) that minimizes the function  $f$ .

An optimization problem can have:

- no solution
- a unique solution
- a finite number of solutions
- an infinite number of solutions

e.g.  $\min_{x \in \mathbb{R}} f(x) := -x^2$

when it exists, we call a solution  $\mathbf{x}^*$  a **minimizer** of  $f$  over  $\Omega$

# Global and local minimizers

$$\min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Definition (Global minimizer)

A point  $\mathbf{x}^* \in \Omega$  is a *global minimizer* of  $f$  over  $\Omega$  if  $f(\mathbf{x}) \geq f(\mathbf{x}^*)$  for all  $\mathbf{x} \in \Omega \setminus \{\mathbf{x}^*\}$ .

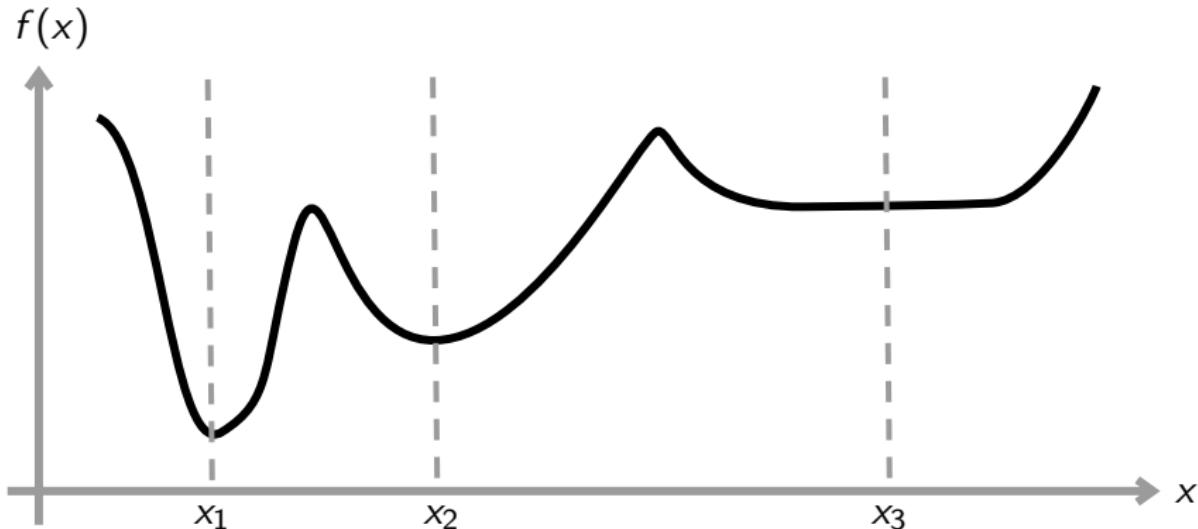
## Definition (Local minimizer)

A point  $\mathbf{x}^* \in \Omega$  is a *local minimizer* of  $f$  over  $\Omega$  if there exists  $\varepsilon > 0$  such that  $f(\mathbf{x}) \geq f(\mathbf{x}^*)$  for all  $\mathbf{x} \in \Omega \setminus \{\mathbf{x}^*\}$  and  $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$ .

## Remarks

- a global minimizer is a local minimizer, but the converse is not in general
- Replace “ $\geq$ ” by “ $>$ ” in definitions: strict (local, global) minimizer
- if  $\mathbf{x}^*$  is a (local, global) minimizer, then the value  $f(\mathbf{x}^*)$  is a (local, global) **minimum**

## Example: unconstrained optimization in 1D



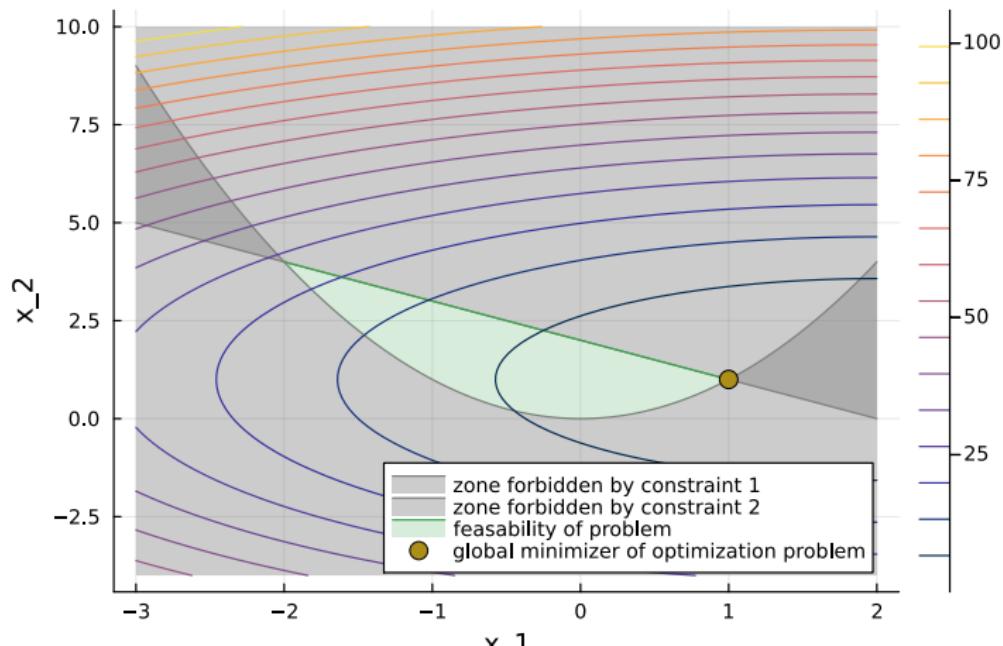
- $x_1$  is a strict global minimizer
- $x_2$  is a strict local minimizer
- $x_3$  is a local minimizer

## Example: constrained optimization in 2D

$$\min_{[x_1, x_2]^\top \in \mathbb{R}^2} (x_1 - 2)^2 + (x_2 - 1)^2 \text{ subject to } \begin{cases} x_1^2 - x_2 \leq 0 \\ x_1 + x_2 \leq 2 \end{cases}$$

## Example: constrained optimization in 2D

$$\min_{[x_1, x_2]^\top \in \mathbb{R}^2} (x_1 - 2)^2 + (x_2 - 1)^2 \text{ subject to } \begin{cases} x_1^2 - x_2 \leq 0 \\ x_1 + x_2 \leq 2 \end{cases}$$



# About notations [disgression]

Many names in the literature

- mathematical programming,
- mathematical optimization,
- numerical optimization,
- optimization

they all mean the field of study of *optimization problems*

## About notations [disgression]

In the literature, you'll find many different equivalent notations for the same optimization problem

$$\min_{\mathbf{x} \in \mathbb{X}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

can be simply written as

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x})$$

or as

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{x} \in \Omega \end{aligned}$$

sometimes, you'll find the notation

$$\arg \min_{\mathbf{x} \in \Omega} f(\mathbf{x})$$

to highlight our interest in finding *minimizers* of the problem.

# Outline

- ① A first example
- ② More examples
- ③ Optimization: vocabulary and nomenclature
- ④ Some standard optimization problems
- ⑤ Recap

# Some standard optimization problems

Several optimization problems are so frequently encountered  
that they are given specific names

being able to recognize to which class belongs a given problem is  
important as resolution methods are usually problem dependent

Some important examples are

- Linear programming
- Quadratic programming
- Convex programming
- Multi-objective programming
- etc.

For each of these cases, we will review the standard form and give some examples.

# Linear programming

Affine objective and constraints: this is a Linear Program (LP).

## LP general form

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} + \mathbf{d} \\ & \text{subject to} && \mathbf{Ax} = \mathbf{b}, \\ & && \mathbf{Gx} \leq \mathbf{h} \end{aligned}$$

## Comments

- the **d** term does not affect the optimization problem – only the **linear** term  $\mathbf{c}^\top \mathbf{x}$  affects the objective function;
- the **feasible set** is a polyhedron;
- LPs are frequently encountered in *operation research*;
- Solving LPs requires specific algorithms (outside of the scope of this course), such as Dantzig's simplex algorithm.

## LP example: diet problem

Example from Boyd and Vanderberghe.

A healthy diet contains  $m$  different nutrients in quantities at least equal to  $b_1, \dots, b_M$ . We can compose such a diet by choosing quantities of  $N$  different foods. One unit of food  $j$  contains  $a_{ij}$  of nutrient  $i$  and has a cost of  $c_j$ .

**Question** write the optimization problem corresponding to choosing the cheapest healthy diet.

## LP example: diet problem

Example from Boyd and Vanderberghe.

A healthy diet contains  $m$  different nutrients in quantities at least equal to  $b_1, \dots, b_M$ . We can compose such a diet by choosing quantities of  $N$  different foods. One unit of food  $j$  contains  $a_{ij}$  of nutrient  $i$  and has a cost of  $c_j$ .

**Question** write the optimization problem corresponding to choosing the cheapest healthy diet.

**Solution** The corresponding optimization problem is

$$\begin{aligned} & \text{minimize} && \mathbf{c}^\top \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \geq \mathbf{b}, \\ & && \mathbf{x} \geq \mathbf{0} \end{aligned}$$

where  $\mathbf{x} \in \mathbb{R}^N$  contains the (non-negative) quantities of each food.

# Quadratic programming

When the objective function is quadratic and the constraints are affine, we obtain a Quadratic Program (QP).

QP general form

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \mathbf{x}^T \mathbf{P} \mathbf{x} + \mathbf{q}^T \mathbf{x} + \mathbf{r} \\ & \text{subject to} && \mathbf{A} \mathbf{x} = \mathbf{b}, \\ & && \mathbf{G} \mathbf{x} \leq \mathbf{h} \end{aligned}$$

where  $\mathbf{P}$  is symmetric (not necessarily positive semidefinite).

## Comments

- Like in LPs, the feasible set is a polyhedron;
- QPs are fundamental for many fields, ranging from signal processing, parameter estimation, regression, machine learning, econometrics, etc.
- have a direct connection with least-squares problems (see session 3)

## QP example: Gaussian maximum likelihood estimation (I)

We want to estimate the mean  $\theta \in \mathbb{R}^N$  of a Gaussian random variable  $\mathcal{N}(\theta, \Sigma)$ , with known covariance matrix  $\Sigma$  from  $M$  i.i.d. measurements  $\mathbf{y}_1, \dots, \mathbf{y}_M \sim \mathcal{N}(\theta, \Sigma)$ . Recall that

$$p(\mathbf{y}_i; \theta) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{y}_i - \theta)^\top \Sigma^{-1} (\mathbf{y}_i - \theta)\right)$$

### Maximum likelihood (ML) principle

$$\text{find } \theta^* = \arg \max_{\theta \in \mathbb{R}^N} \mathcal{L}(\theta; \mathbf{y}_1, \dots, \mathbf{y}_M) := p(\mathbf{y}_1, \dots, \mathbf{y}_M; \theta)$$

The function  $\mathcal{L}(\theta; \mathbf{y}_1, \dots, \mathbf{y}_M)$  is called the *likelihood function*.

The goal of ML estimation is to find the parameter  $\theta$  for which the data  $\mathbf{y}_1, \dots, \mathbf{y}_M$  has the highest joint probability.

**Exercise:** write the corresponding optimization problem!

## QP example: Gaussian maximum likelihood estimation (II)

### Solution

First, since observations are i.i.d.,  $p(\mathbf{y}_1, \dots, \mathbf{y}_M; \boldsymbol{\theta}) = \prod_{i=1}^M p(\mathbf{y}_i; \boldsymbol{\theta})$ .

Second, observe that the ML principle is equivalent to

$$\text{find } \boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta} \in \mathbb{R}^N} \mathcal{L}(\boldsymbol{\theta}; \mathbf{y}_1, \dots, \mathbf{y}_M) = \arg \max_{\boldsymbol{\theta} \in \mathbb{R}^N} \log \mathcal{L}(\boldsymbol{\theta}; \mathbf{y}_1, \dots, \mathbf{y}_M)$$

since the likelihood is positive (by definition) and log is monotone.

Thus,

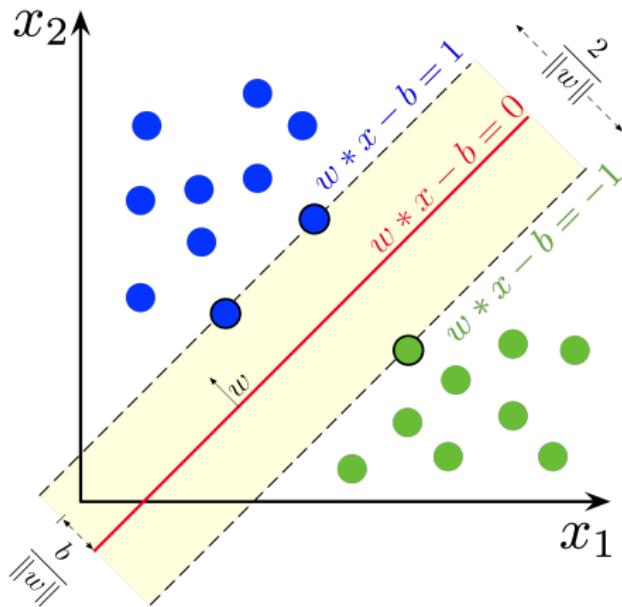
$$\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta} \in \mathbb{R}^N} -\frac{1}{2} \sum_{i=1}^M (\mathbf{y}_i - \boldsymbol{\theta})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y}_i - \boldsymbol{\theta})$$

which can be rewritten as the (unconstrained) QP

$$\text{minimize} \quad \frac{1}{2} \boldsymbol{\theta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\theta} - \frac{1}{M} \left( \sum_{i=1}^M \mathbf{y}_i \right)^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\theta}$$

# QP example: support vector machines (SVMs)

SVMs are a **supervised learning** tool for **classification** and **regression**.



Idea: find the *maximum-margin* hyperplane that divides the data into two classes → one obtains a constrained QP.

More on this in **TPs** !

# Convex programming

LPs and QPs (under some conditions) are **convex problems**.

In the next session, we will precisely define this notion and see why it is crucial for many optimization problems.

**General form of a convex problem**

$$\text{minimize } f(\mathbf{x})$$

$$\text{subject to } g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, M$$

$$\mathbf{a}_i^\top \mathbf{x} = b_i, \quad i = 1, \dots, P$$

where  $f$  and  $g_1, \dots, g_M$  are **convex** functions.

When we will discuss constrained optimization problems, we will solely focus on **convex optimization** problems.

## Multi-objective optimization (I)

Sometimes we are interested in optimization problems that involve multiple objective functions, say  $F_1, F_2, \dots, F_Q$ . Multi-objective problems can be formulated as

$$\min_{\mathbf{x} \in \Omega} [F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_Q(\mathbf{x})]$$

where  $\Omega \subset \mathbb{R}^N$  is the feasible set.

A feasible point  $\mathbf{x} \in \Omega$  is said to *Pareto dominate* another solution  $\mathbf{y} \in \Omega$  if

- $F_i(\mathbf{x}) \leq F_i(\mathbf{y})$  for all  $i = 1, \dots, Q$
- $F_i(\mathbf{x}) < F_i(\mathbf{y})$  for at least one  $i$ .

A point  $\mathbf{x}^{po}$  is said to be *Pareto optimal* if there does not exist another solution that dominates it. The set of such points is called the *Pareto front* or *Pareto boundary*.

## Multi-objective optimization (II)

A typical transformation of the multi-objective optimization problem

$$\min_{\mathbf{x} \in \Omega} [F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_Q(\mathbf{x})]$$

is the **scalarization** procedure

$$\min_{\mathbf{x} \in \Omega} \lambda_1 F_1(\mathbf{x}) + \lambda_2 F_2(\mathbf{x}) + \dots + \lambda_Q F_Q(\mathbf{x})$$

where  $\lambda_1, \lambda_2, \dots, \lambda_Q > 0$  are weights controlling the trade-offs between the  $Q$  objective terms.

Under **certain conditions**, solutions of the scalarized problem give Pareto optimal points (but not necessarily all such points).

The notion of *regularization* can be interpreted in terms of multi-objective optimization. More in TPs!

# Outline

- ① A first example
- ② More examples
- ③ Optimization: vocabulary and nomenclature
- ④ Some standard optimization problems
- ⑤ Recap

# Recap and take-home message

## General form

$$\min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) \text{ subject to } \mathbf{x} \in \Omega$$

## Which questions do we ask in optimization?

- Existence and uniqueness of a solution  $\mathbf{x}^*$
- Characterization of solutions: necessary or sufficient conditions for  $\mathbf{x}^*$  to be a solution
- Design an efficient way to compute  $\mathbf{x}^*$  numerically.
  - by a closed-form solution (not very often)
  - usually, by an algorithm which produces a sequence

$$\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}, \dots \xrightarrow{n \rightarrow \infty} \mathbf{x}^*$$

- Characterize the properties of this algorithm: rate of convergence, numerical complexity, etc.