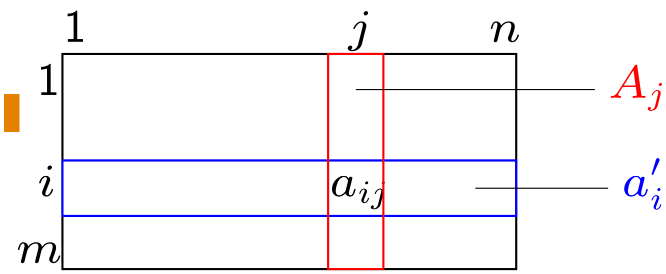
Operations Research (Master's Degree Course)

2. Mathematical Programming

Silvano Martello
DEI "Guglielmo Marconi", Università di Bologna, Italy

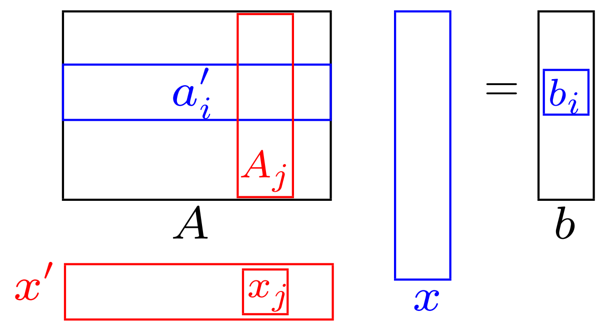
Notation

- R (or R^1): set of real numbers; R^n: n-dimensional vector space.
- $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$: column vector of n elements ($\equiv$ point in R^n);
- $x' = [x_1 \ldots x_n] = (x_1, \ldots, x_n) =$ row vector;
- $A = [a_{ij}] = m \times n$ matrix:



- $Ax = b \iff a'_i x = b_i \quad (i = 1, \ldots, m)$;

$$\iff \sum_{j=1}^{n} x_j A_j = b.$$



- $\det(A)$: determinant of A;
- $S = \{s_1, s_2, \ldots\}$: set of elements $s_1, s_2, \ldots$;
- $S = \{x : \mathcal{P}(x)\}$: set of those x for which property $\mathcal{P}$ holds;
- $|S|$: number of elements in S.

General optimization problem

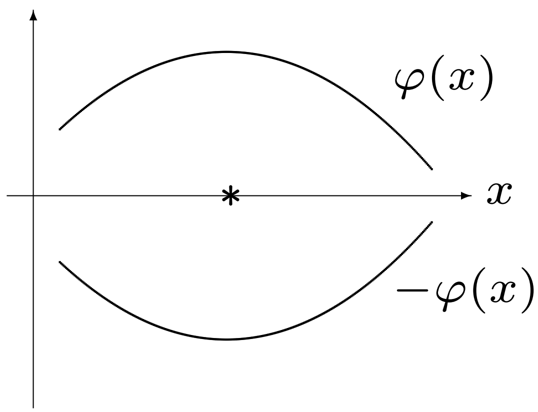
- $x = (x_1, x_2, \ldots, x_n) \in R^n$ = vector of decision variables = point in R^n.
- $F \subseteq R^n$ = set of the **feasible solutions**
- $\varphi : F \to R$ = **objective function (cost function)**
- **Optimization problem:** $\min_{x \in F} \varphi(x)$

  find a point (vector) $x^* \in F$ (**global optimum**) such that:

$$\varphi(x^*) \leq \varphi(x) \; \forall \, x \in F$$

- If the objective function φ has to be maximized **(profit function)**
  1. minimize $-\varphi$;
  2. invert the sign of the solution value, i.e.:

     $\max \varphi(x) = -\min(-\varphi(x))$



Classifying optimization problems

- The **Feasible region** F is normally defined by equations and inequalities:

$$\begin{aligned} \min \quad & \varphi(x) \\ & h_j(x) = 0 \quad (j = 1, \dots, p) \\ & g_i(x) \geq 0 \quad (i = 1, \dots, q) \end{aligned}$$

1. If φ, h_j and g_i are general functions $\Rightarrow$ **Non Linear Programming**: we only know **non efficient** algorithms which can find the global optimum for small-size problem instances, or a local optimum for larger instances, but can also fail in finding any feasible solution.

We will see that

2. If φ is convex, g_i is concave $\forall i$, and h_j is linear $\forall j$ $\Rightarrow$ **Convex Programming**:

we know algorithms which can find a **local optimum** for small- or medium-size problem instances, **but**

a local optimum is always a **global optimum**.

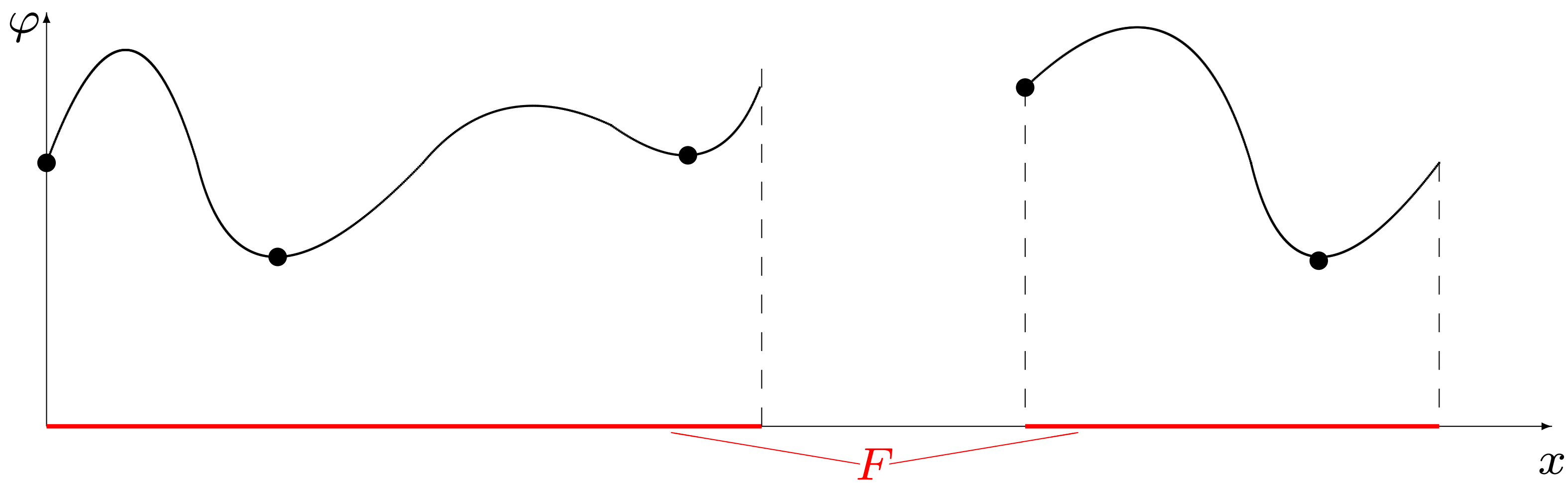
3. If φ, h_j and g_i are all linear $\Rightarrow$ **Linear Programming**:

the **simplex algorithm** (**very efficient**) easily finds a **global optimum** even for very large problem instances.

Non Linear Programming

- In a general optimization problem $\min_{x \in F} \varphi(x)$

    1. φ is a general function, and F is a general set. Hence:
    2. F can be empty (no solution exists) or non-continuous;
    3. local optima can exist ($\bullet$):



- We don't know efficient algorithms to exactly solve this problem

  (algorithms do not have a "sufficiently complete vision" of F and φ);
- we know algorithms which can find the optimal solution, within reasonable times, for small size instances or an approximate (sub-optimal) solution for larger instances.

Optimization problems

- Another definition of **optimization problem** we will use: (F, d), with

  F= set of feasible points (solutions); d: $F \longrightarrow R^1$ (cost function).

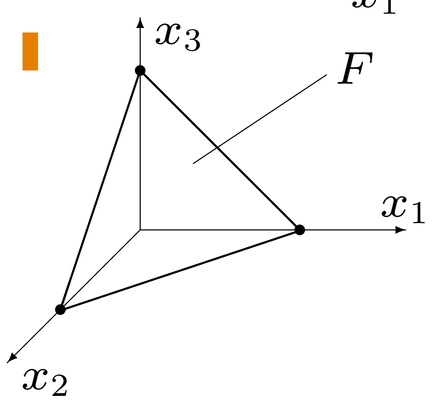
- Problem: find $f \in F$ (global optimum) such that $d(f) \leq d(y) \quad \forall y \in F$.

- Example: Linear Programming:

$$\left.\begin{array}{ll} \text{min} & c'x \\ & Ax = b \\ & x \geq 0 \end{array}\right\} \Leftrightarrow \left\{\begin{array}{l} (F, d) \\ F{=}\{x \in R^n : Ax = b,\ x \geq 0\} \\ d : x \rightarrow c'x \end{array}\right.$$

- Numerical example: m=1, n=3, A=[1 1 1] , b=[2]:

$$\begin{array}{lccccccl} \min & c_1x_1 & + & c_2x_2 & + & c_3x_3 & & \\ \text{s.t.} & x_1 & + & x_2 & + & x_3 & = 2 \\ & x_1 & , & x_2 & , & x_3 & \geq 0 \end{array}$$



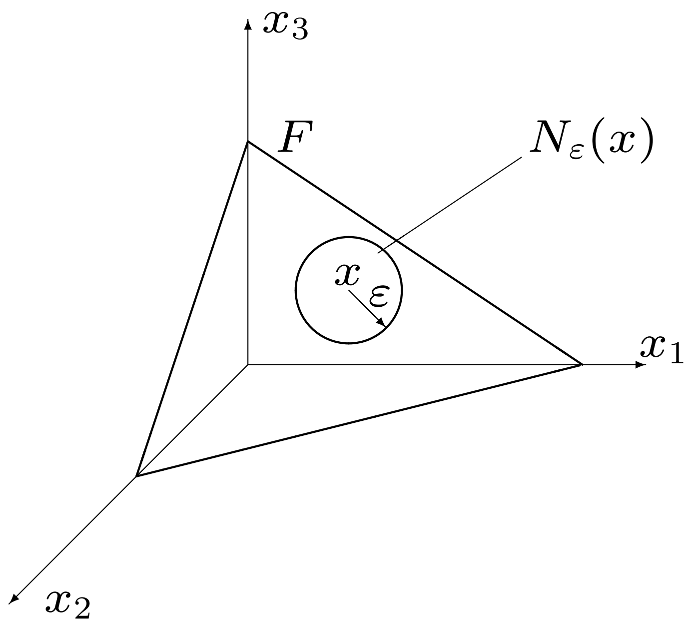
Neighborhoods

- Given a set F, we define:

$$2^F = \text{set of all subsets of } F.$$

- Given a problem (F, d), a **Neighborhood** is a function $N : F \longrightarrow 2^F$ (very general definition).
- Example: (LP) $F = \{x \in R^n : Ax = b, \, x \geq 0\}$;

  for a prefixed $\varepsilon > 0$, possible neighborhood of $x \in F$: $N_\varepsilon(x) = \{y \in F : \| y - x \| \leq \varepsilon \}$ (**Euclidean neighborhood**).



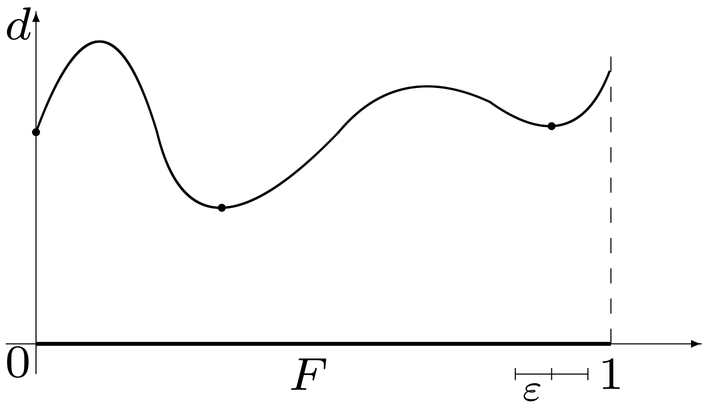
Local and global optima

- Given a problem (F, d), and a neighborhood N,

  $f \in F$ is **locally optimum** with respect to N if:

$$d(f) \leq d(p) \quad \forall p \in N(f).$$

- Example: $F = [0, 1] \subset R^1$, $N_\varepsilon(f) \; = \{x \in F \; : \; |x - f| \leq \varepsilon\}$



- Given(F, d) and N , N is **exact** if:

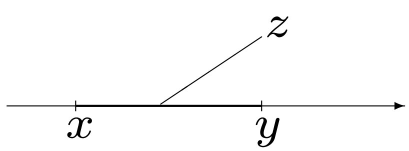
$$(f \in F \text{ locally optimum with respect to } N) \Longrightarrow (f \text{ globally optimum}).$$

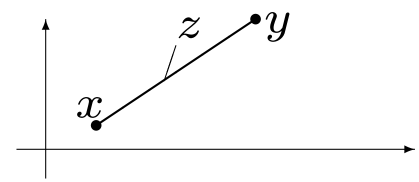
- Example: $N_1(f) \; = \; \{x \in [0, 1] \; : \; |x - f| \leq 1\}$, obviously exact.

Convex sets

- Given $x, y \in R^n$, a **convex combination** of x and y is any $z \in R^n$ defined by

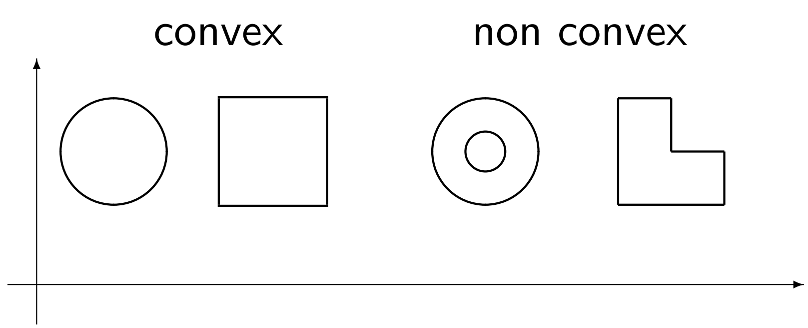
$$z = \lambda x + (1 - \lambda) y \text{ with } \lambda \in R^1, 0 \le \lambda \le 1.$$

- Example: $x, y \in R^1$ 

$x, y \in R^2$ 

- $S \subseteq R^n$ is a **convex set** if

$$\forall\, x, y \in S,\ \forall\, \lambda\ (0 \le \lambda \le 1), z = \lambda x + (1 - \lambda) y \in S.$$

- Examples in R^2: 

- **Property 0** *R^n is convex* (proof immediate from definition).

- **Property 1** *Given convex sets S_i, $\cap S_i$ is convex.*

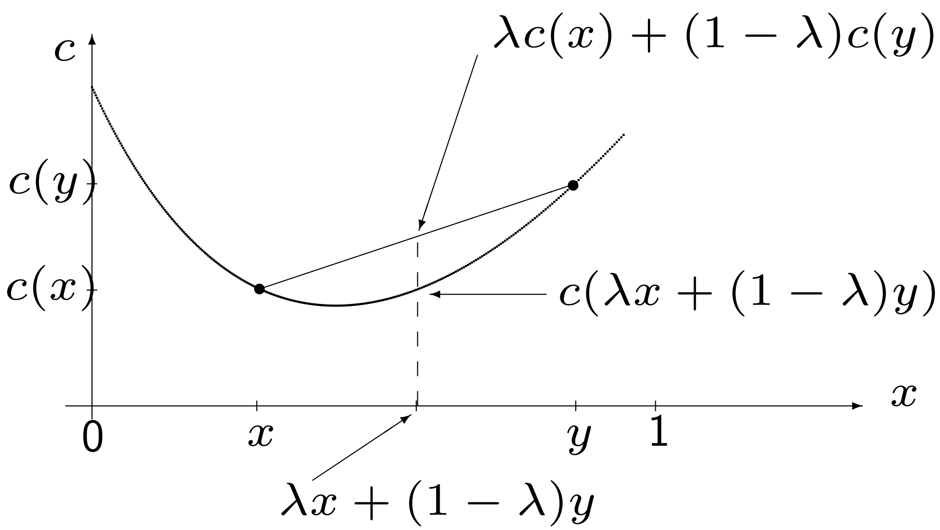
Proof $x, y \in \cap S_i \Rightarrow x, y \in S_i\ \forall\, i \Rightarrow z \in S_i \forall\, i \Rightarrow z \in \cap S_i$. $\square$

Convex functions

- Given $S \subseteq R^n$ convex, $c : S \to R^1$ is **convex in** S if

$$\forall\ x, y \in S,\ \forall \lambda (0 \leq \lambda \leq 1),\quad c(\lambda x + (1-\lambda)\ y) \leq \lambda c(x) + (1-\lambda) c(y).$$

- Example: $S = [0, 1] \subset R^1$:



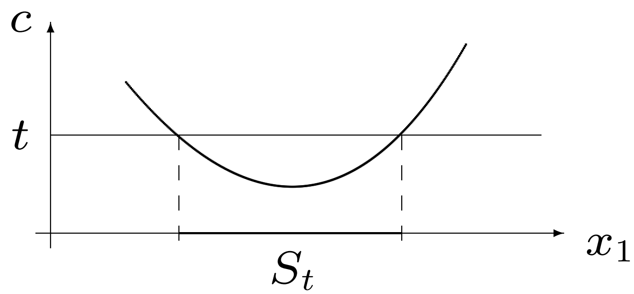
- **Property 2** *Given $c(x)$ convex in S convex, $\forall\ t\ S_t = \{x \in S : c(x) \leq t\}$ is convex.*

  Proof Given $x, y \in S_t$, $\lambda x + (1-\lambda)y \in S$, and

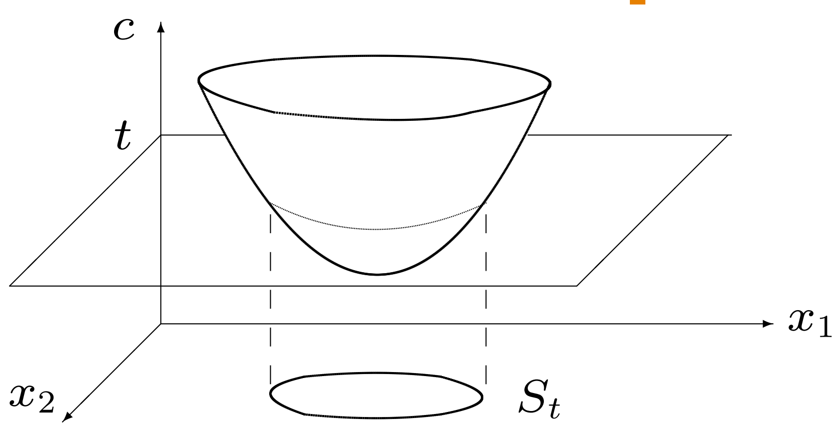
$$c(\lambda x + (1-\lambda)\ y) \leq \lambda c(x) + (1-\lambda)\ c(y) \leq \lambda t + (1-\lambda) t = t \ \Rightarrow\ \lambda x + (1-\lambda) y \in S_t. \square$$

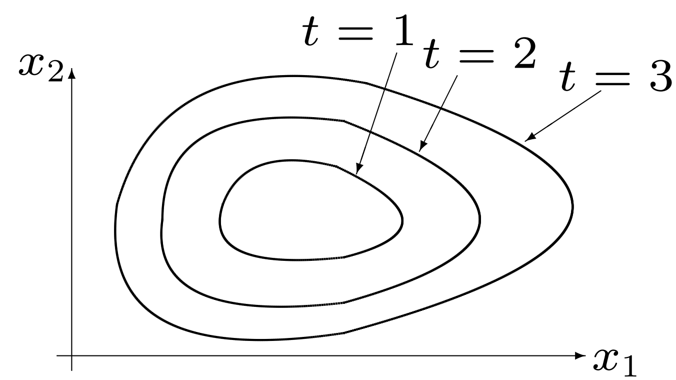
Convex functions (cont'd)

- Example: $S \subseteq R^1$

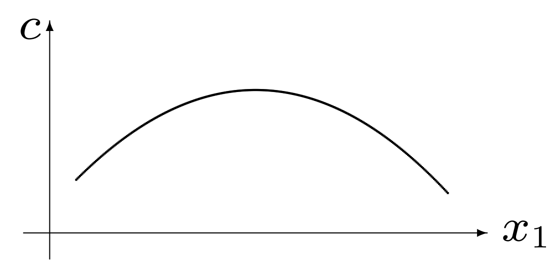


- Example: $S \subseteq \ R^2$





- A function c, defined in S convex, is **concave** if $-c$ is convex in S:



- A **linear function** is both concave and convex.

Convex programming

- Let us consider the problem of minimizing a convex function over a convex set:

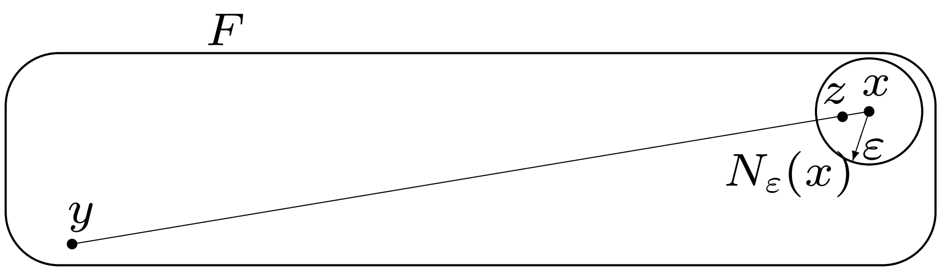
- **Theorem** *Given (F, c) with $F \subseteq R^n$ convex and c convex in F, the neighborhood*

$$N_\varepsilon(x) = \{y \in F : \| x - y \| \leq \varepsilon\}$$

is exact $\forall\ \varepsilon > 0$.

Proof $x =$ local optimum with respect to N_ε; $y \in F$;

take $z = \lambda x + (1 - \lambda) y$ in $N_\varepsilon(x)$ (λ close to 1);



$$c(z) = c(\lambda x + (1 - \lambda) y) \leq \lambda c(x) + (1 - \lambda)\, c(y) \Rightarrow c(y) \geq \frac{c(z) - \lambda c(x)}{1 - \lambda};$$

$$z \in N_\varepsilon(x) \Rightarrow c(z) \geq c(x) \Rightarrow c(y) \geq \frac{c(x) - \lambda c(x)}{1 - \lambda} = c(x). \ \square$$

Convex programming (cont'd)

- (F, c) is a **Convex Programming Problem (CP)** if
  - c is convex;
  - $F \subseteq R^n$ is defined by

    $$g_i(x) \geq 0 \ (i = 1, \dots, q)$$

    with $g_i : R^n \to R^1$ concave $\forall \ i$.
- Relationship with the previous definition:
- A constraint $h_j(x) = 0$ with h_j linear can be replaced by a pair of constraints:

  $$\begin{aligned} h_j(x) &\geq 0 \\ -h_j(x) &\geq 0 \end{aligned}$$

  (both $h_j(x)$ and $-h_j(x)$ are concave)
- **Property** *In a CP, F is convex.*

  Proof $-g_i$ is convex $\forall \ i \Rightarrow F_i = \{x \in R^n : g_i(x) \geq 0\} = \{x \in R^n : -g_i(x) \leq 0\}$ is convex $\forall \ i$ (by **Property 2**);

  $\Rightarrow F = \cap \ F_i$ is convex (by **Property 1**). $\square$ **Hence**
- **In a CP a local optimum with respect to the Euclidean distance is a global optimum.**
- The same holds for linear programming (c linear; F defined by linear functions).