UNIVERSITAT POLITÈCNICA DE CATALUNYA

DAC

## Algorithmic Methods for Mathematical Models (AMMM)

# Intro to Heuristics Methods

**Luis Velasco**

(lvelasco @ ac.upc.edu)

Campus Nord D6-107

# Mathematical programming typology

- Linear Programming (LP)

$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & Ax = b \\ & x \geq 0 \quad x \in \Re^n \end{aligned}$$

- Integer Linear Programming (ILP)

$$\begin{aligned} \min \quad & c^T x \\ s.t. \quad & Ax = b \\ & x \geq 0 \quad x \in Z^n \end{aligned}$$

- Mixed Integer Linear Programming (MILP)

$$\begin{aligned} \min \quad & c^T x + d^T y \\ s.t. \quad & Ax + By = b \\ & x \geq 0 \quad x \in Z^n \\ & y \geq 0 \quad y \in \Re^n \end{aligned}$$

- Nonlinear Programming (NLP)

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & g_i(x) \leq b_i \quad \forall i \\ & x \geq 0 \end{aligned}$$

# Time to solve

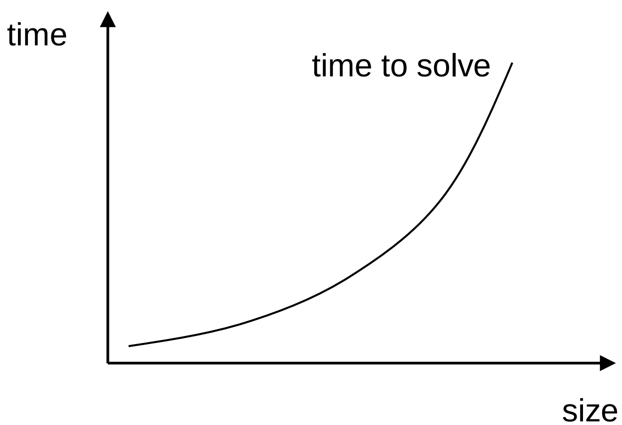
$$
\begin{array}{lll}
\min & c^T x & \\
s.t. & Ax = b & \\
 & x \geq 0 & x \in \mathrm{Z}^n
\end{array}
$$

$A_{[\mathrm{n}\times\mathrm{m}]}$

Size = #vars × #constraints

Example

$$
\begin{array}{l}
\text{minimize} \quad z \\
\text{subject to:} \\
\qquad \sum_{c \in C} x_{tc} = 1 \quad \forall t \in T \\
\qquad \sum_{t \in T} r_t \cdot x_{tc} \leq r_c \quad \forall c \in C \\
\qquad z \geq \frac{1}{r_c} \cdot \sum_{t \in T} r_t \cdot x_{tc} \quad \forall c \in C
\end{array}
$$

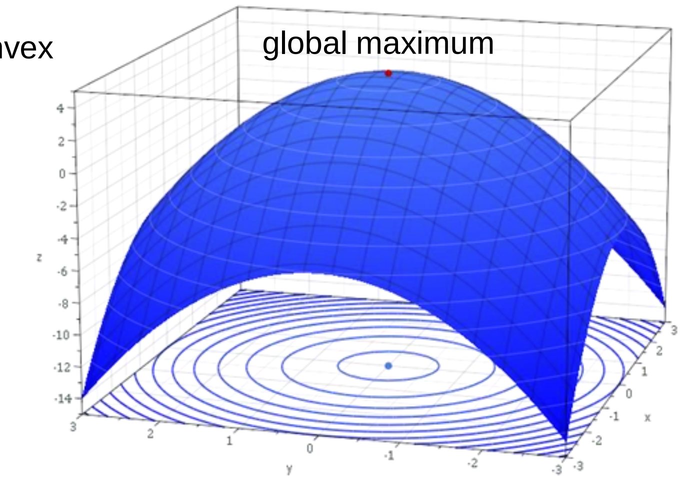


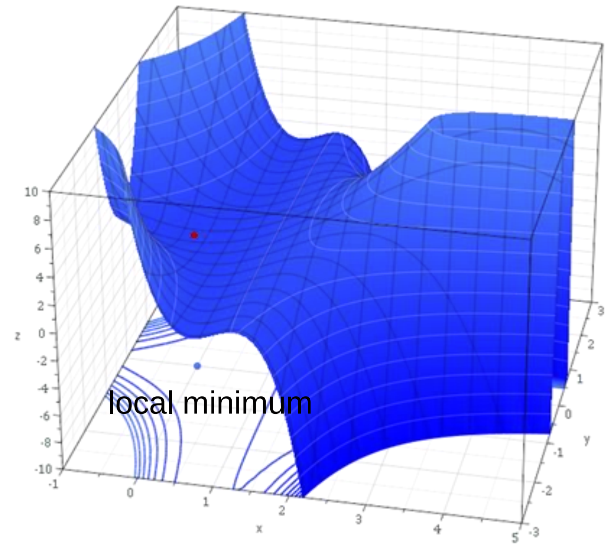

What happens when

time to solve >>> time a solution is needed

# How to find maxima and minima of a function

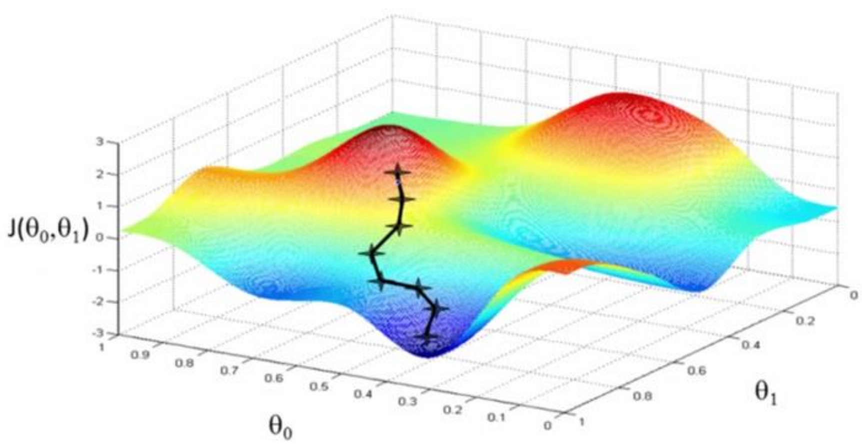
- Local extrema of **differentiable functions** can be found by Fermat's theorem, which states that they must occur at *critical points*.
- One can distinguish whether a critical point is a local maximum or local minimum by using the **first and second derivative tests**.
  - The **first derivative** finds the critical points.
  - The **second derivative** test uses the value of the second derivative at the critical points to determine whether they are a **local** maximum or minimum.

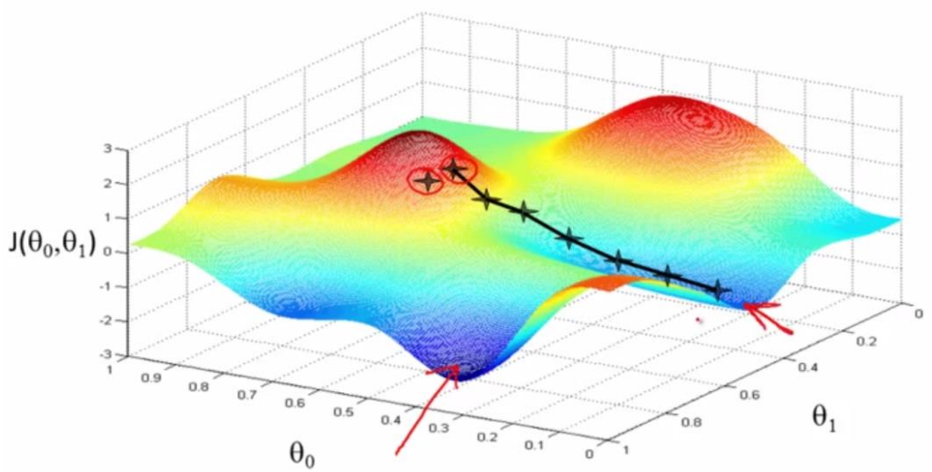


# Gradient Descent

- A first-order iterative optimization algorithm for finding a local minimum of a differentiable function.
- We take steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point.

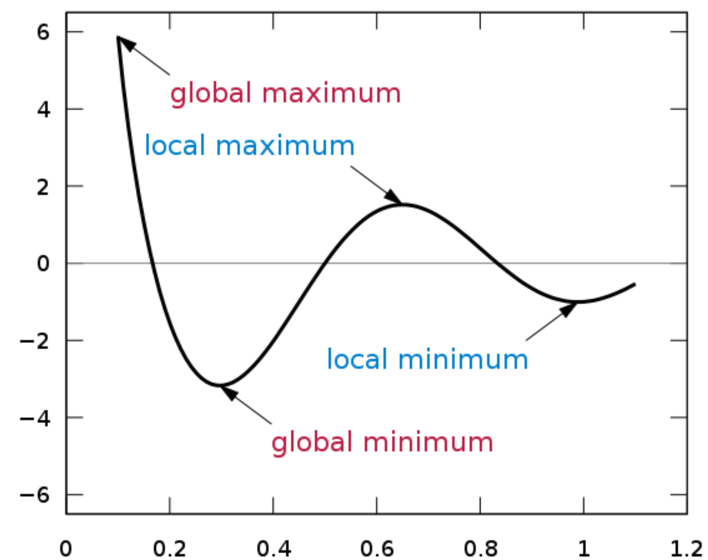



Can be susceptible to **local minima.**

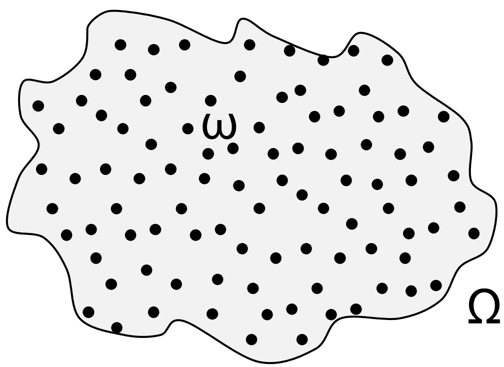
# Heuristics

- Approximate solution techniques that have been used since the beginnings of operations research to tackle difficult **combinatorial problems**.
- With the development of complexity theory in the early 70's, it became clear that, since most of these problems were indeed NP-hard,
  - there was **little hope** of ever finding efficient exact solution procedures for them.
- This emphasized the role of heuristics for solving the problems that were encountered in **real-life applications** and that needed to be tackled, whether or not they were NP-hard.
- Heuristics usually consists of two phases:
  - **Constructive Phase**, where a solution is built.
    - **Greedy algorithms** or any other method based on the **problem structure** can be used.
  - **Local search**, where the solution is improved.

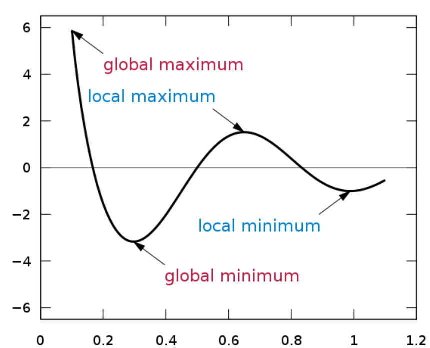


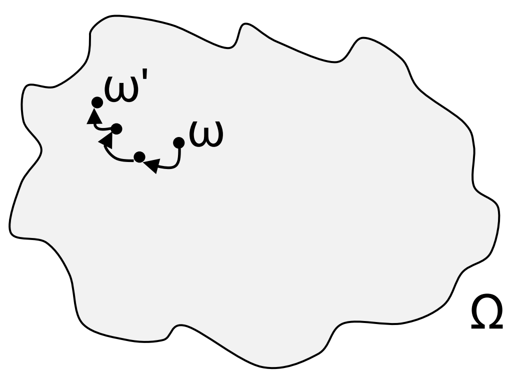
# Heuristics

| | |
|---|---|
| $\Omega$ | Solution space |
| $f: \Omega \rightarrow \mathbb{R}$ | Objective function defined on the solution space |
| goal: | find $\omega^* \in \Omega,\ f(\omega) \geq f(\omega^*)\ \forall \omega \in \Omega$ |




$\omega^* \in \Omega,\ f(\omega) \geq f(\omega^*)\ \forall \omega \in \Omega$




ω' = is a local minimum

# Algorithmic Methods for Mathematical Models (AMMM)

# Intro to Heuristics Methods

**Luis Velasco**

(lvelasco @ ac.upc.edu)

Campus Nord D6-107