UNIVERSITAT POLITÈCNICA DE CATALUNYA

DAC

Algorithmic Methods for Mathematical Models (AMMM)

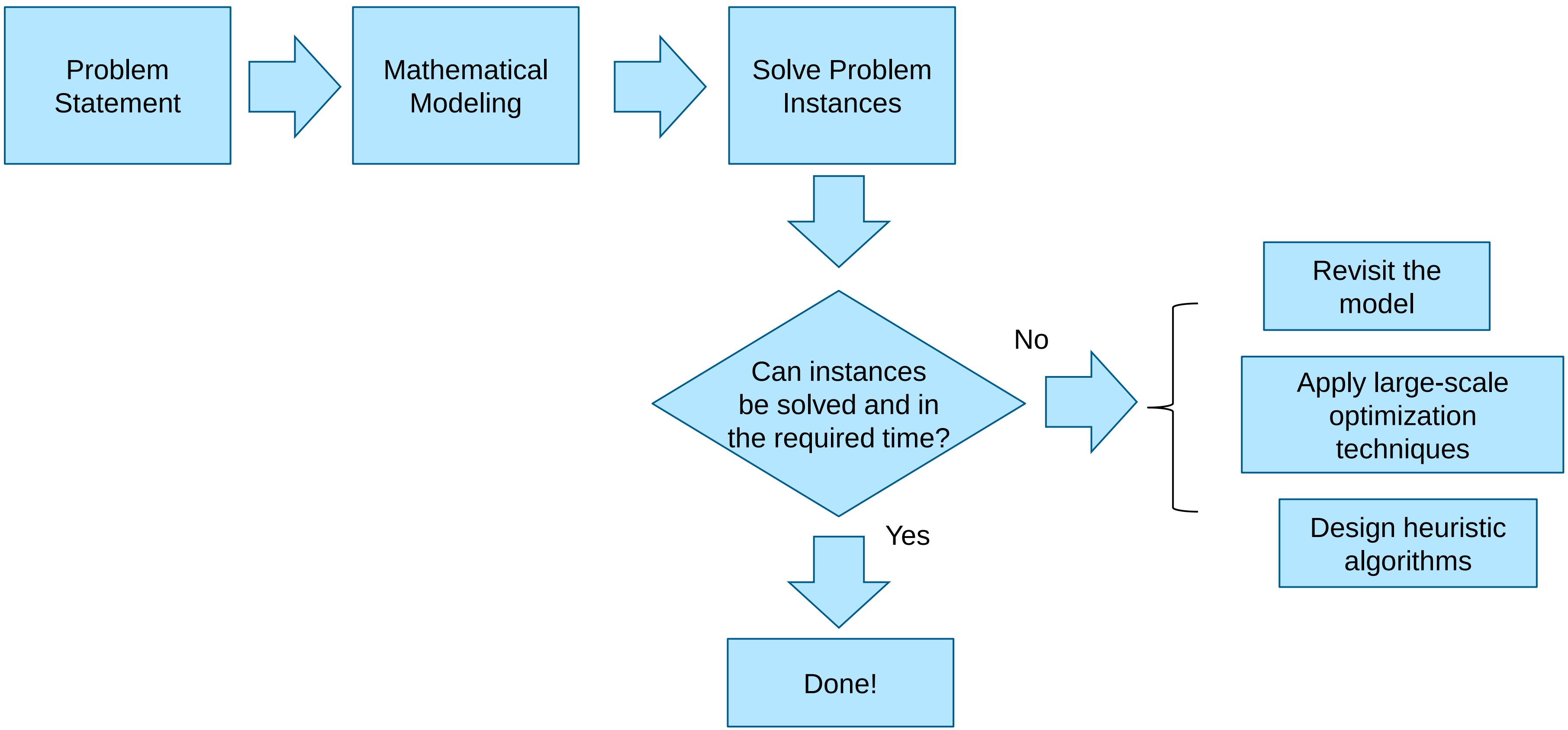
Intro to Heuristics Methods

Luis Velasco

(luis.velasco@upc.edu)

Campus Nord D6-107

General Methodology to Deal with Optimization Problems



Problem Statement

- Given:
  - Sets and Parameters (input data) definition
- Output:
  - What data we want to obtain from the solution
- Minimization or maximization objective
- Constraints of the problem

Mathematical programming typology

- Linear Programming (LP)

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

- Integer Linear Programming (ILP)

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

- Mixed Integer Linear Programming (MILP)

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

- Nonlinear Programming (NLP)

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

Problem Instance

Data + Model = Problem Instance

```
nTasks=5;
nCPUs=3;

rt=[261.27 560.89 310.51 105.80 344.7];
rc=[505.67 503.68 701.78];
```

$$\text{minimize} \quad z$$

subject to:

$$\sum_{c \in C} x_{tc} = 1 \quad \forall t \in T$$

$$\sum_{t \in T} r_t \cdot x_{tc} \le r_c \quad \forall c \in C$$

$$z \ge \frac{1}{r_c} \cdot \sum_{t \in T} r_t \cdot x_{tc} \quad \forall c \in C$$

Size of an Instance

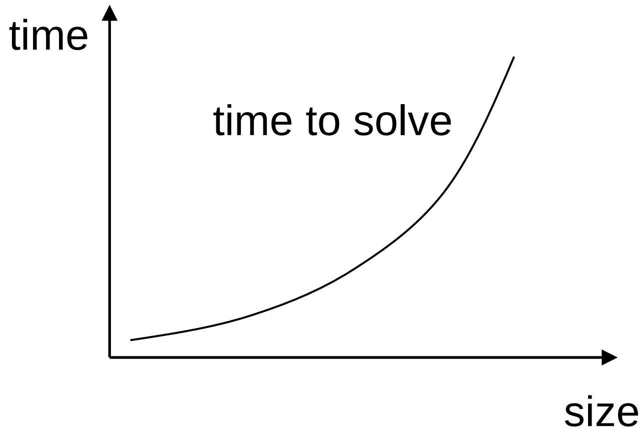
$$
\begin{array}{ll}
\min & c^T x \\
s.t. & Ax = b \\
 & x \geq 0 \quad x \in Z^n
\end{array}
$$

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

Size = #vars × #constraints

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

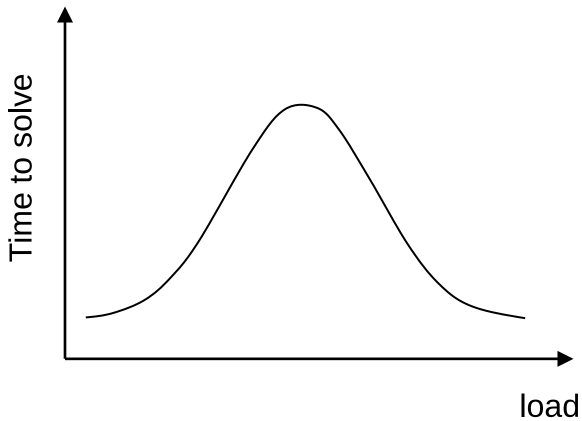
Time to solve

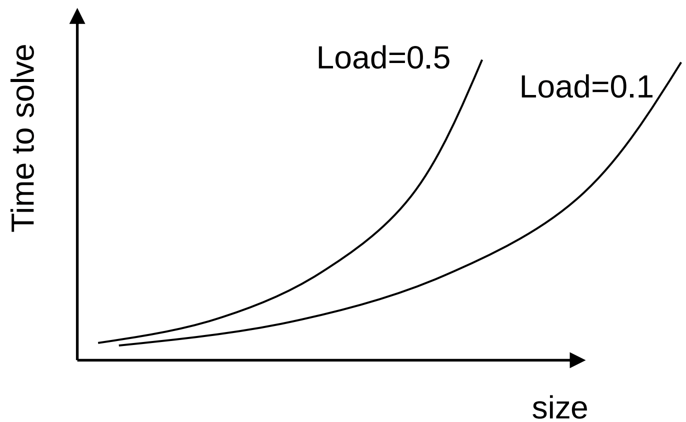


What happens when

time to solve >>> time a solution is needed

Time to solve vs load





$$\text{minimize} \quad z$$

subject to:

$$\sum_{c \in C} x_{tc} = 1 \quad \forall t \in T$$

$$\sum_{t \in T} r_t \cdot x_{tc} \le r_c \quad \forall c \in C$$

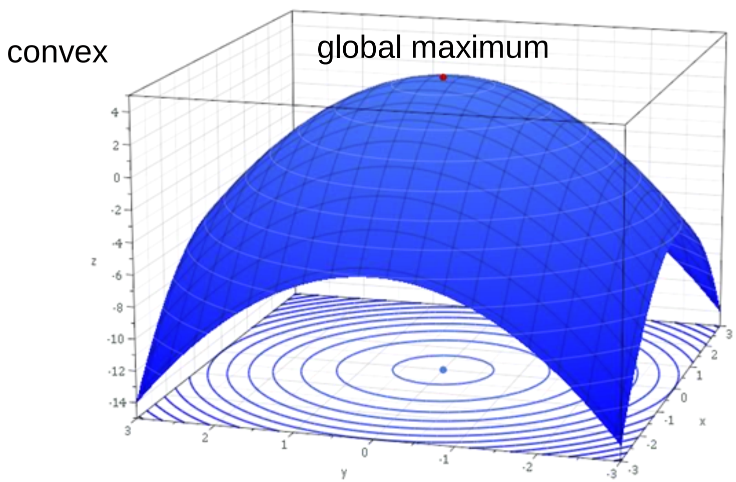
$$z \ge \frac{1}{r_c} \cdot \sum_{t \in T} r_t \cdot x_{tc} \quad \forall c \in C$$

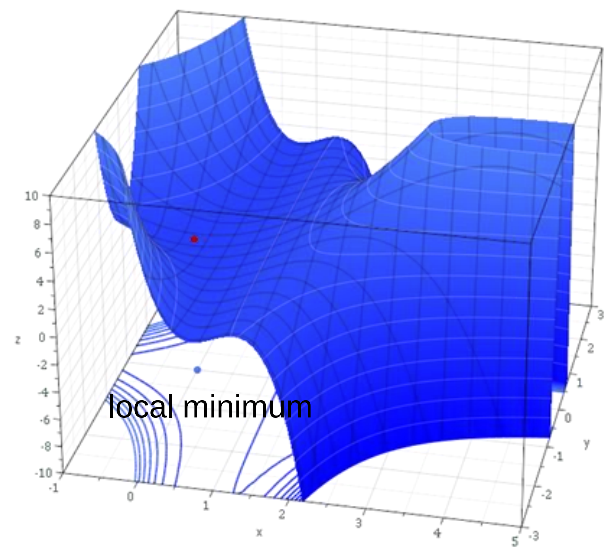
$$load = \frac{\sum_{t \in T} r_t}{\sum_{c \in C} r_c}$$

Heuristics Methods

How to find maxima and minima of a function

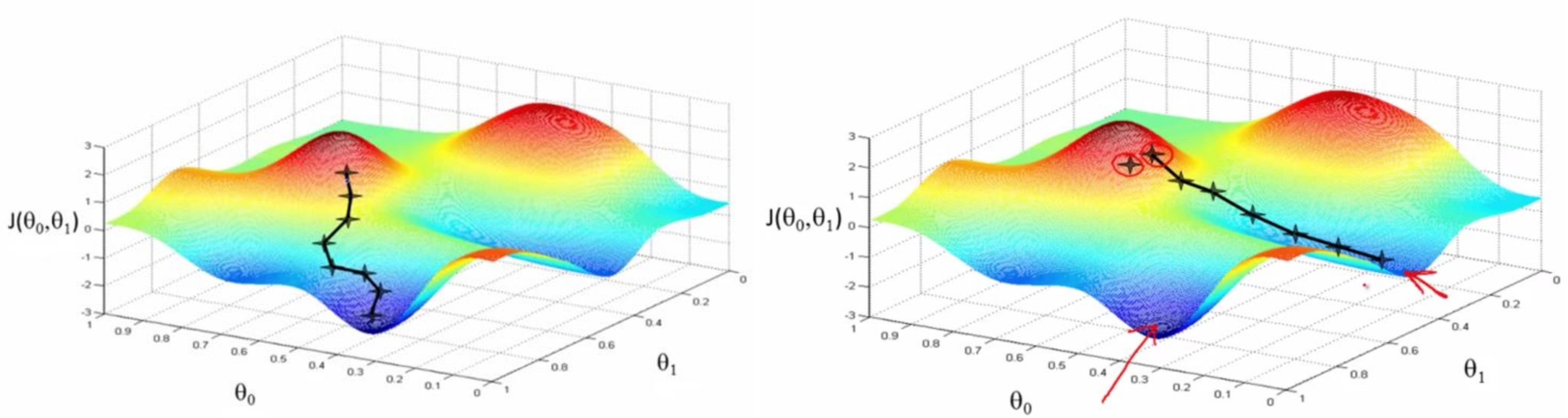
- Local extrema of **differentiable functions** can be found by Fermat's theorem, which states that they must occur at *critical points*.
  - The first derivative finds the critical points.
- One can distinguish whether a critical point is a local maximum or local minimum by using the **second derivative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





Gradient Descent

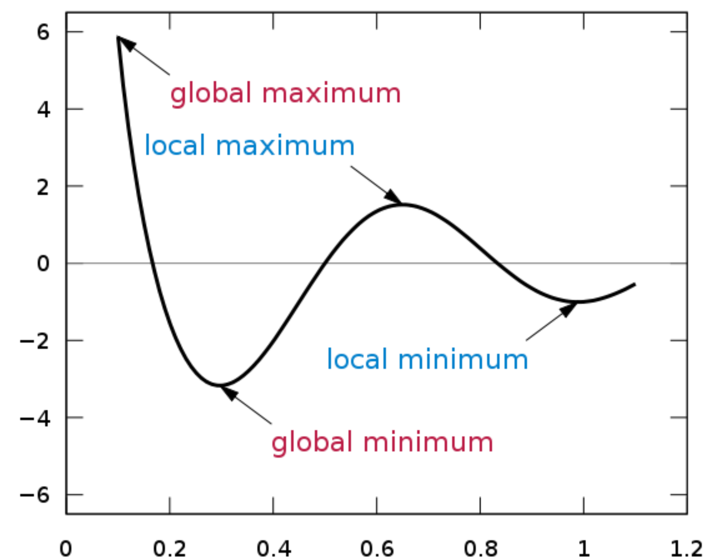
- A first-order iterative optimization algorithm for finding a local minimum of a differentiable function.
- We take steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point.



Can be susceptible to **local minima.**

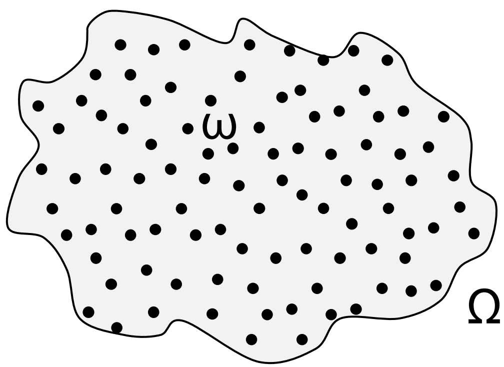
Heuristics

- 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, 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.

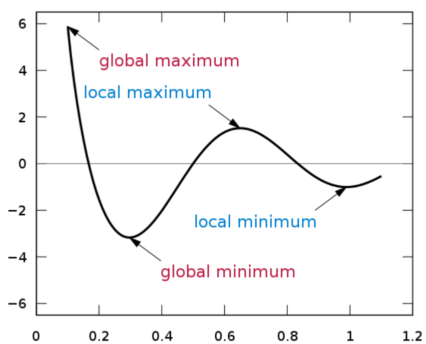


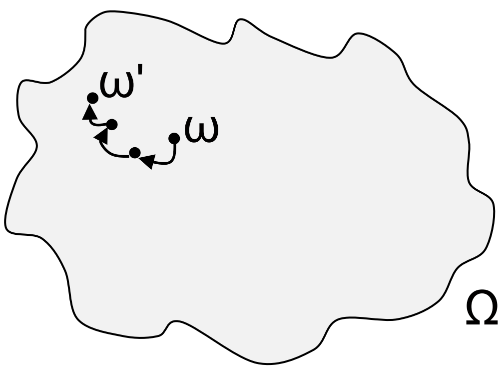
Heuristics

Ω	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

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Algorithmic Methods for Mathematical Models (AMMM)

Intro to Heuristics Methods

Luis Velasco

(lvelasco @ ac.upc.edu)

Campus Nord D6-107