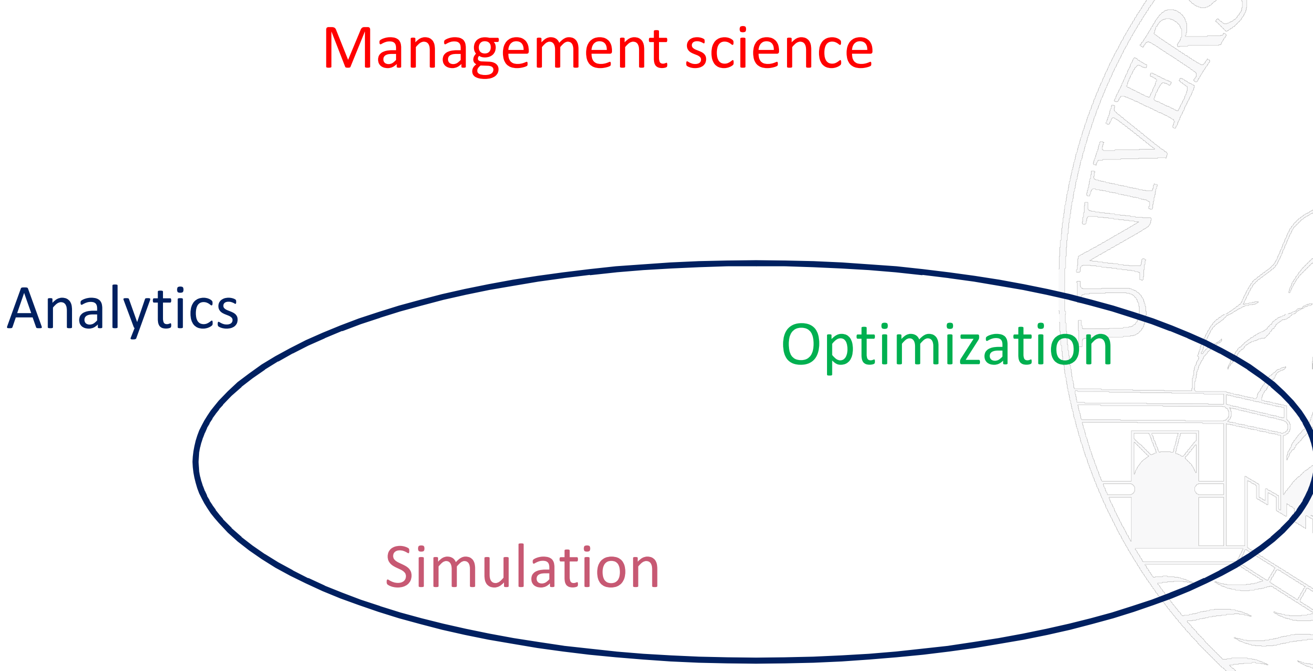
Fundamentals of optimization
Part I

M. Grazia Speranza
University of Brescia

Operations research



Operations research

Management science

Analytics

Optimization

Simulation

What if analysis

UNIVERSITY OF BRESCIA

Operations research

Management science

Analytics

Optimization

Simulation

UNIVERSITY
OF BRESCIA

Optimization

What is it?

Why do we care?

When do we use it?

UNIVERSITY
OF BRESCIA

Optimization

$$\boxed{\begin{array}{c} \min c(x) \\ \\ x \in F \end{array}}$$

Decision variables

One objective (objective function)

Constraints (feasible region)

Optimization

One objective only?

UNIVERSITY
OF BRESCIA

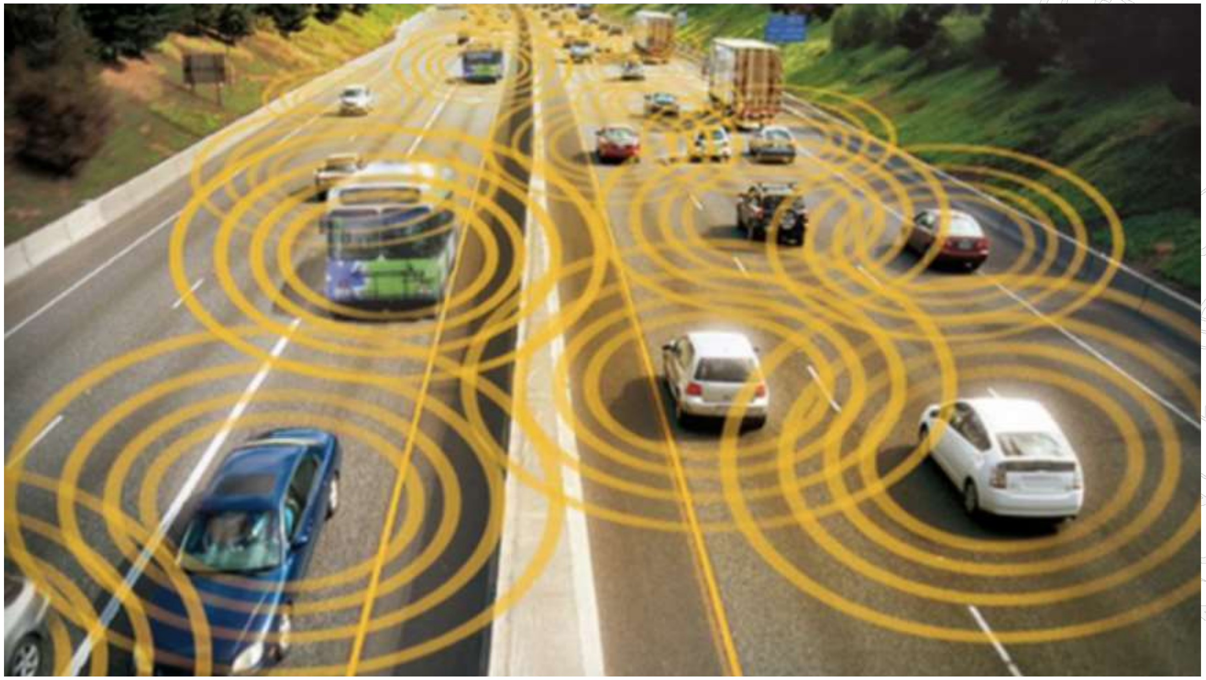
Optimization

One objective only?

Safety

‘Low’ cost

Small traveling time



Fair traffic assignment

Environmentally friendly

UNIVERSITY OF BRESCIA

Optimization

Multiple objectives

a) Combination of objectives into a single one

$$\min f(x) \qquad \max g(x) \quad \rightarrow \quad \max g(x)-p\,f(x)$$

Method not always appropriate

Optimization

Multiple objectives

b) One objective chosen as objective with the others into constraints with thresholds

min $f(x)$	$\rightarrow$	min $f(x)$
max $g(x)$		$g(x) \geq k$

UNIVERSITY OF BRESCIA

Optimization

a)

max g(x)-p f(x)

Choice of p

b)

min f(x)

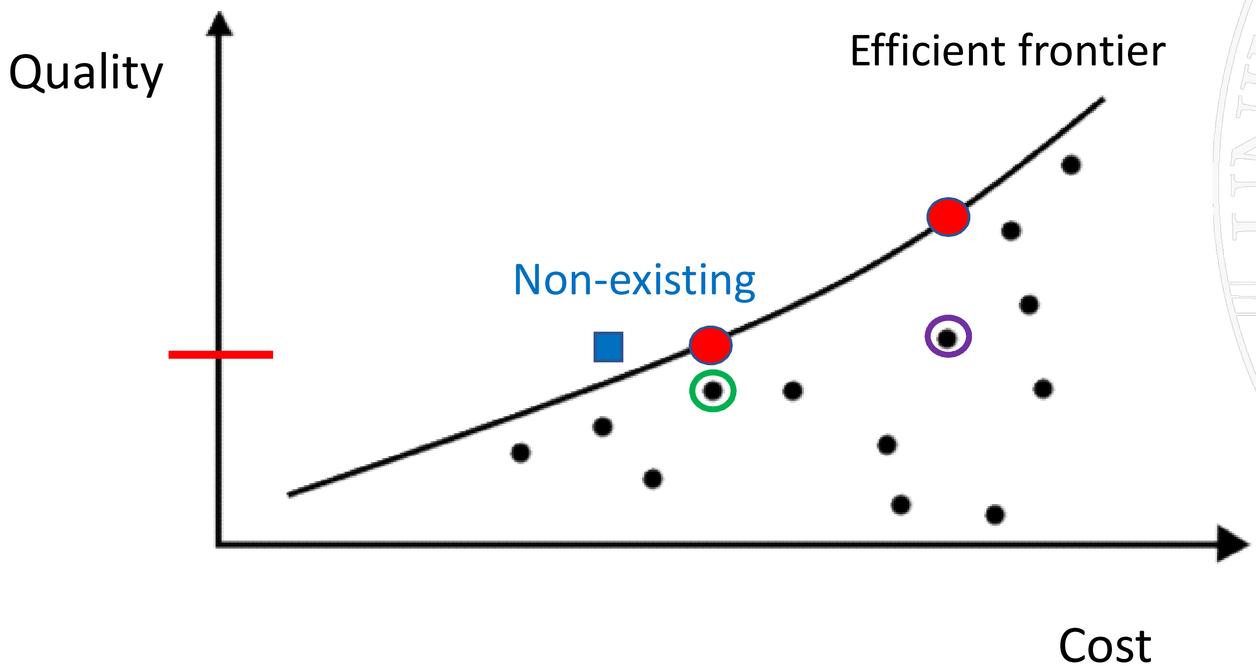
g(x)≥k

Choice of k

Multiple optimization problems, each with one objective

Efficient frontier

Optimization



UNIVERSITY OF BRESCIA

Optimization problem

$$\min c(x)$$

$$x \in F$$

Types of objectives

$$\boxed{min \sum_i \; c_i(x)}$$

Average

$$\boxed{minmax \; c_i(x)}$$

Worst

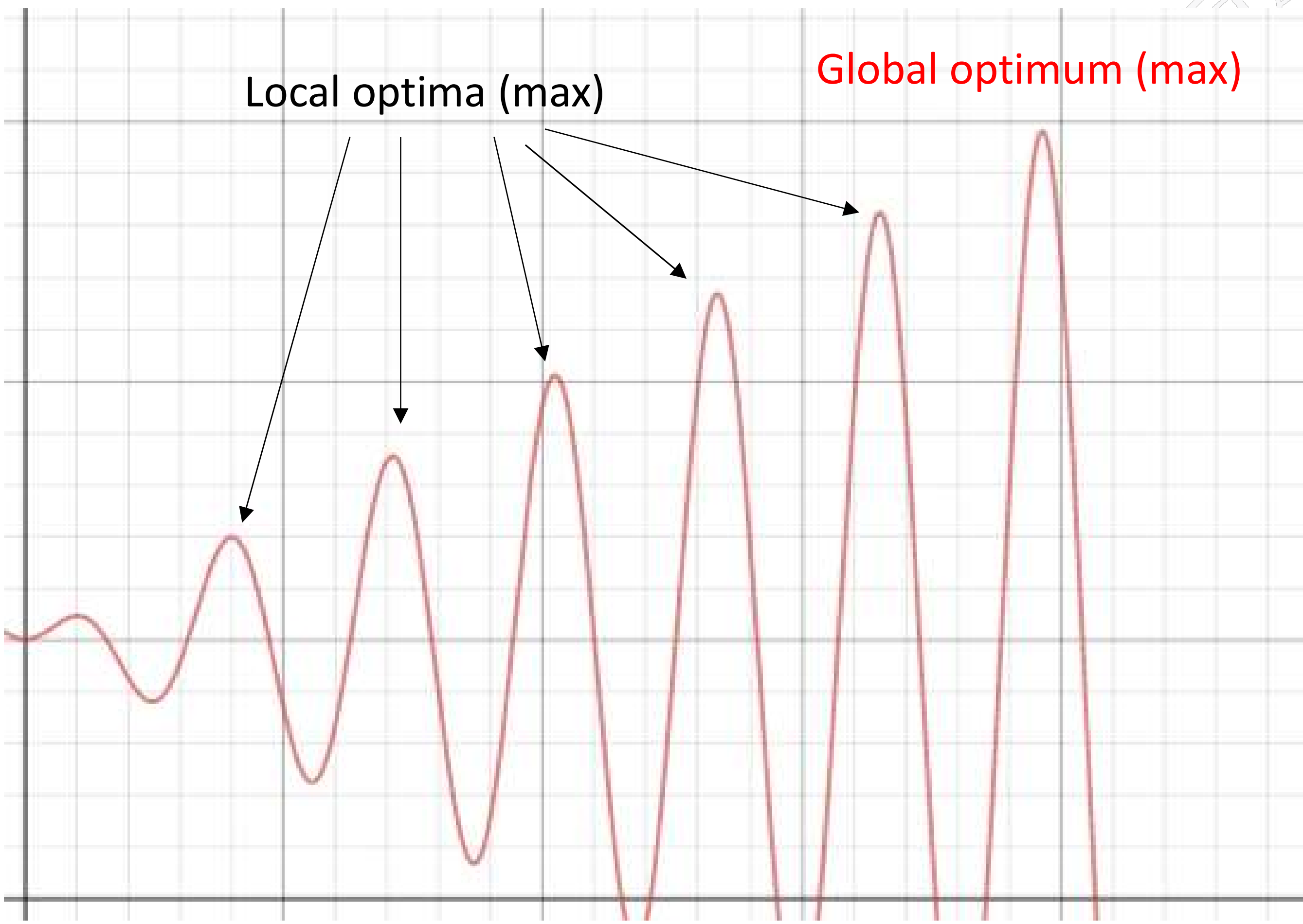
$$min \sum_i \; c_i(x)$$

sum over a given percentage of the worst i

CVaR-like
(can be used on scenarios)

UNIVERSITY OF BRESCIA

Local and global optima



What do we need?

- We know how to analitically find local optima for ‘simple’ optimization problems
- We know how to analitically find the global optimum for some ‘very simple’ optimization problems
- We do not know for all the others – **algorithms needed**

Convex problem

$$\min_{x \in F} c(x)$$

F	Convex set
$c(x)$	Convex function on F

$\Downarrow$

Convex problem

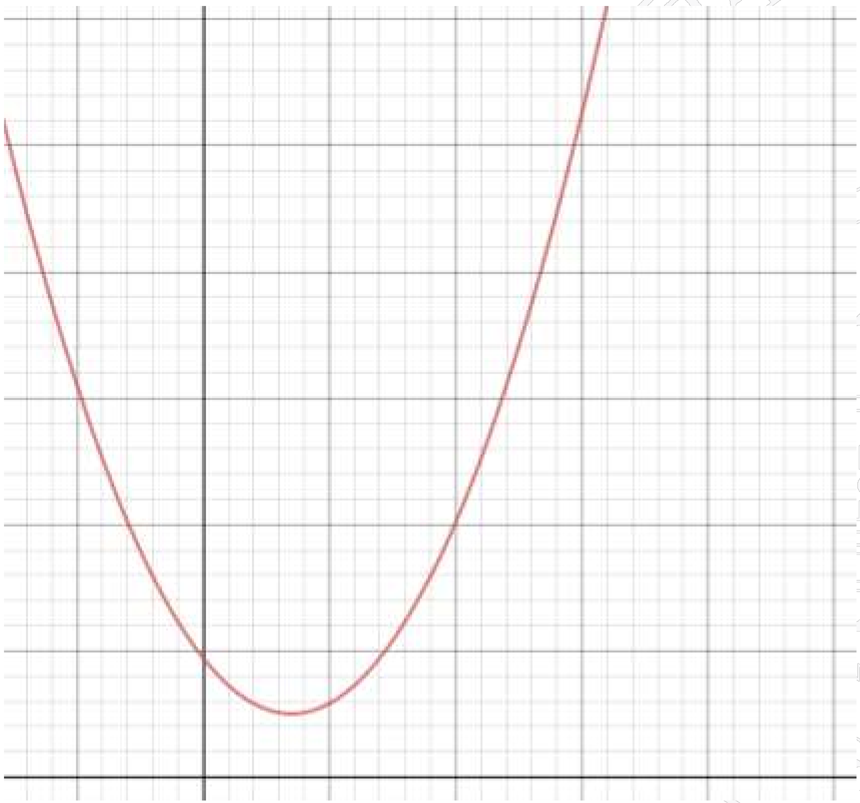
Convex problem

Property:

For a convex problem, any local optimum is a global optimum

↓

Effective and efficient algorithms



Convex problem

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

Theorem:

If $\varphi(x)$ is convex, $g_i(x)$ are concave, $h_j(x)$ are linear, the optimization problem is a convex problem

UNIVERSITY OF BRESCIA

Linear programming

$$\min c'x$$
$$Ax = b$$
$$x \geq 0$$

Special case of convex problem

Model - algorithm

Mathematical programming formulation

⬇

Algorithm

(for the formulation and not for a specific problem)

Linear programming

$$\min c'x$$
$$Ax = b$$
$$x \geq 0$$

LP problems can be solved with efficient algorithms:

Simplex method

Interior point methods (Khachiyan, Karmarkar)

Computational complexity

Computational complexity or simply **complexity** of an algorithm is the amount of resources required to run it

Resources: time and memory

Computational complexity

Worst-case complexity of an algorithm

$n \rightarrow f(n)$

n: size of the input

f(n) = max f(I) over all instances I of size n

Problem: Sorting

Instance I: 6, 35, 7, 15, 27, 12, 18

Algorithm: Bubblesort

Size: 7

f(7) = max f(I) over all instances I of size 7

Computational complexity

f(n) very difficult to obtain

Asymptotic behavior (when *n* tends to the infinity)

The complexity is expressed by using big O notation

Complexity of Bubblesort: O(n^2)

Computational complexity

The complexity of a problem is the infimum of the complexities of the algorithms that solve the problem, including unknown algorithms.

Complexity of problem Sorting: O(n log n)
(thanks to Heapsort)

Computational complexity

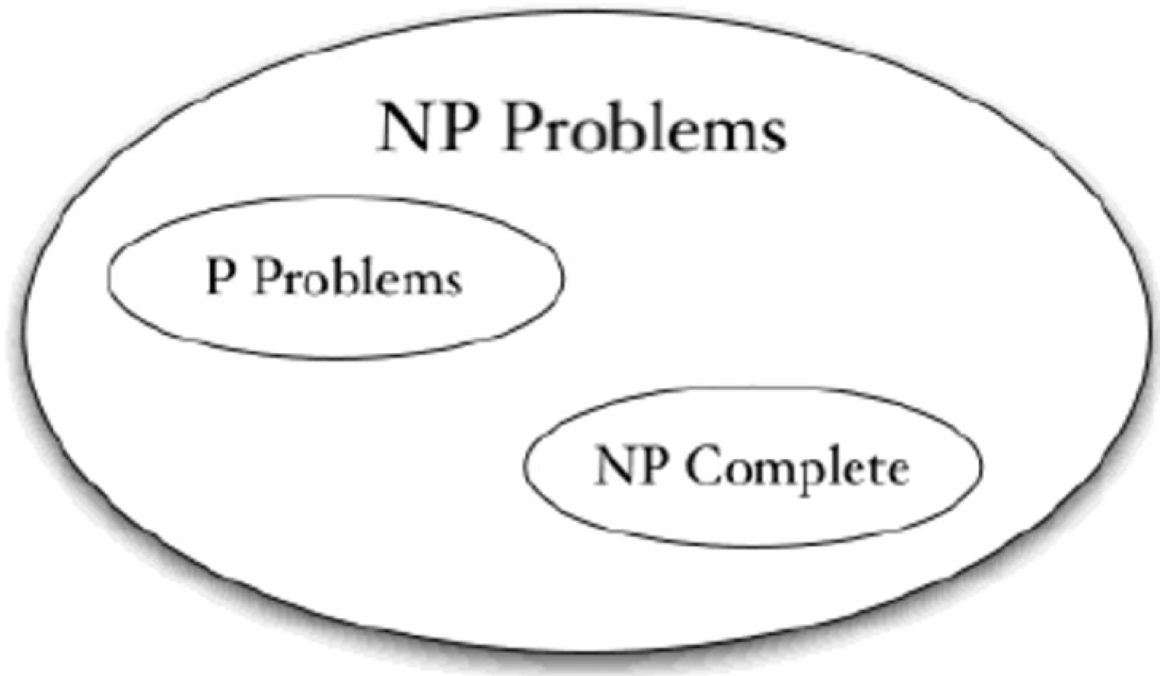
Why do we care about computational complexity?

O(n) O (n log n) O(n^2) O(2^n)

	$n=10$	$n=20$	$n=30$	$n=40$	$n=50$	$n=60$
$O(n)$	$10^{-5''}$	$2 \cdot 10^{-5''}$	$3 \cdot 10^{-5''}$	$4 \cdot 10^{-5''}$	$5 \cdot 10^{-5''}$	$6 \cdot 10^{-5''}$
$O(n^2)$	$10^{-4''}$	$4 \cdot 10^{-4''}$	$9 \cdot 10^{-4''}$	$16 \cdot 10^{-4''}$	$25 \cdot 10^{-4''}$	$36 \cdot 10^{-4''}$
$O(2^n)$	$10^{-3''}$	$1''$	$17.9'$	12.7 *days*	35.7 *years*	366 *centuries*

CPU speed *1000 O(2^n) $\longrightarrow$ size solved increases by 10

Computational complexity: classes



P=NP?

Conjecture: no

Computational complexity: LP

George Dantzig (1914-2005) in 1947 invented the **simplex method**

Worst-case complexity: O($2^{\frac{m}{2}-1}$)

In practice: Rarely time required is greater than O(m log n)

Complexity of LP unknown until 1979

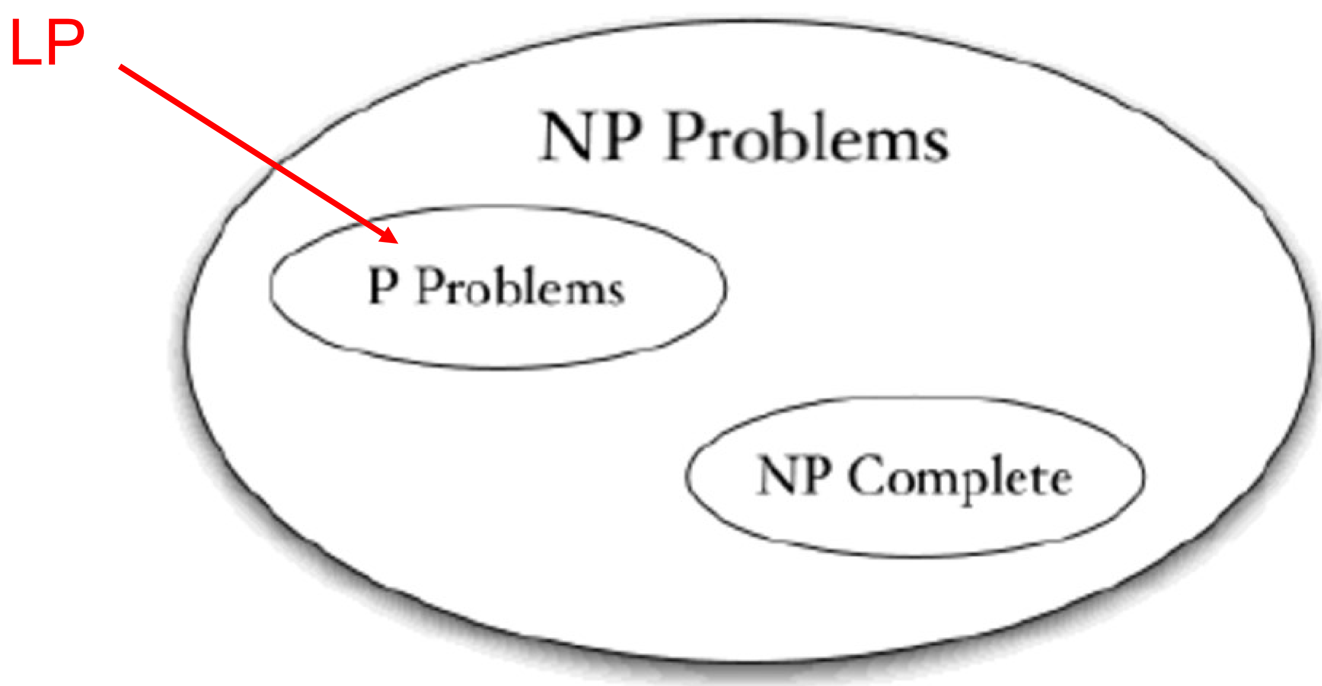
Leonid Khachiyan (1952-2005) in 1979 invented the **ellipsoid method**

Worst-case complexity: O(n^6)

In practice: O(n^6)

UNIVERSITY OF BRESCIA

Computational complexity: LP

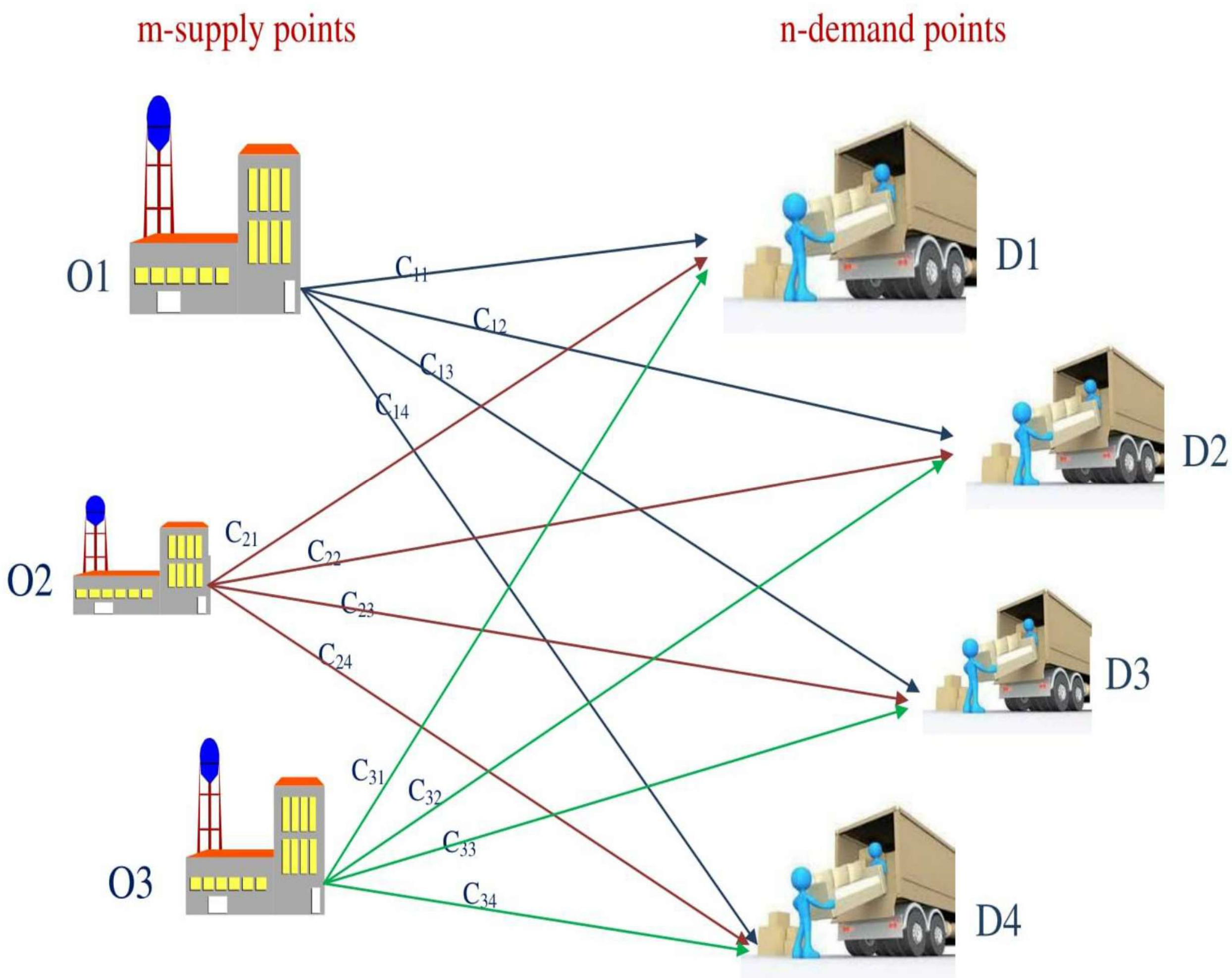


Solution of LP

More and more powerful software available

(CPLEX, Gurobi)
(GLPK, LP-SOLVE)

Planning problems



UNIVERSITY OF BRESCIA

Planning problems

$$\min \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{j=n} x_{ij} = s_i \quad (i = 1, 2, \ldots, m) \qquad \text{(Supply constraints)}$$

$$\sum_{i=1}^{i=m} x_{ij} = d_j \quad (j = 1, 2, \ldots, n) \qquad \text{(Demand constraints)}$$

$$x_{ij} \geq 0 \quad (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$$

Optimization problems

Mixed integer linear programming

Non-linear programming

Global optimization

Non-convex optimization

Stochastic programming

Robust optimization

Mixed integer linear programming

Fixed costs

Scheduling

Location

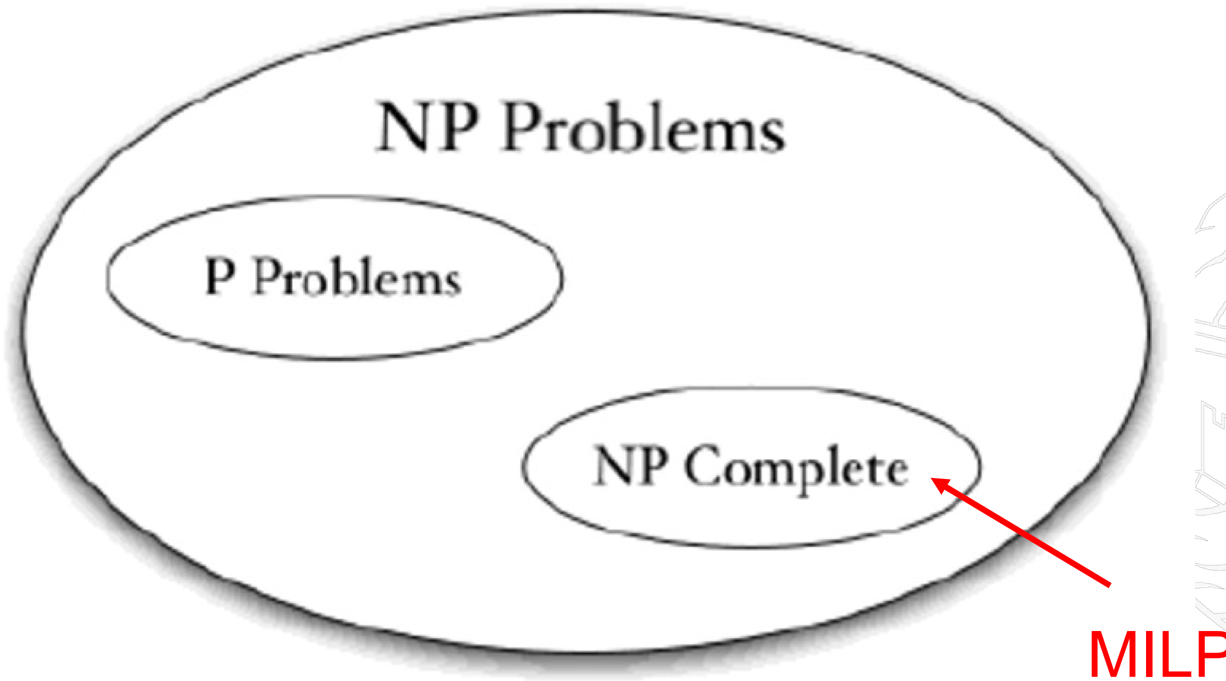
$$\min c'x$$
$$Ax = b$$
$$x \geq 0$$

Integer

Selection

Routing

Mixed integer linear programming



Mixed integer linear programming

Branch-and-bound

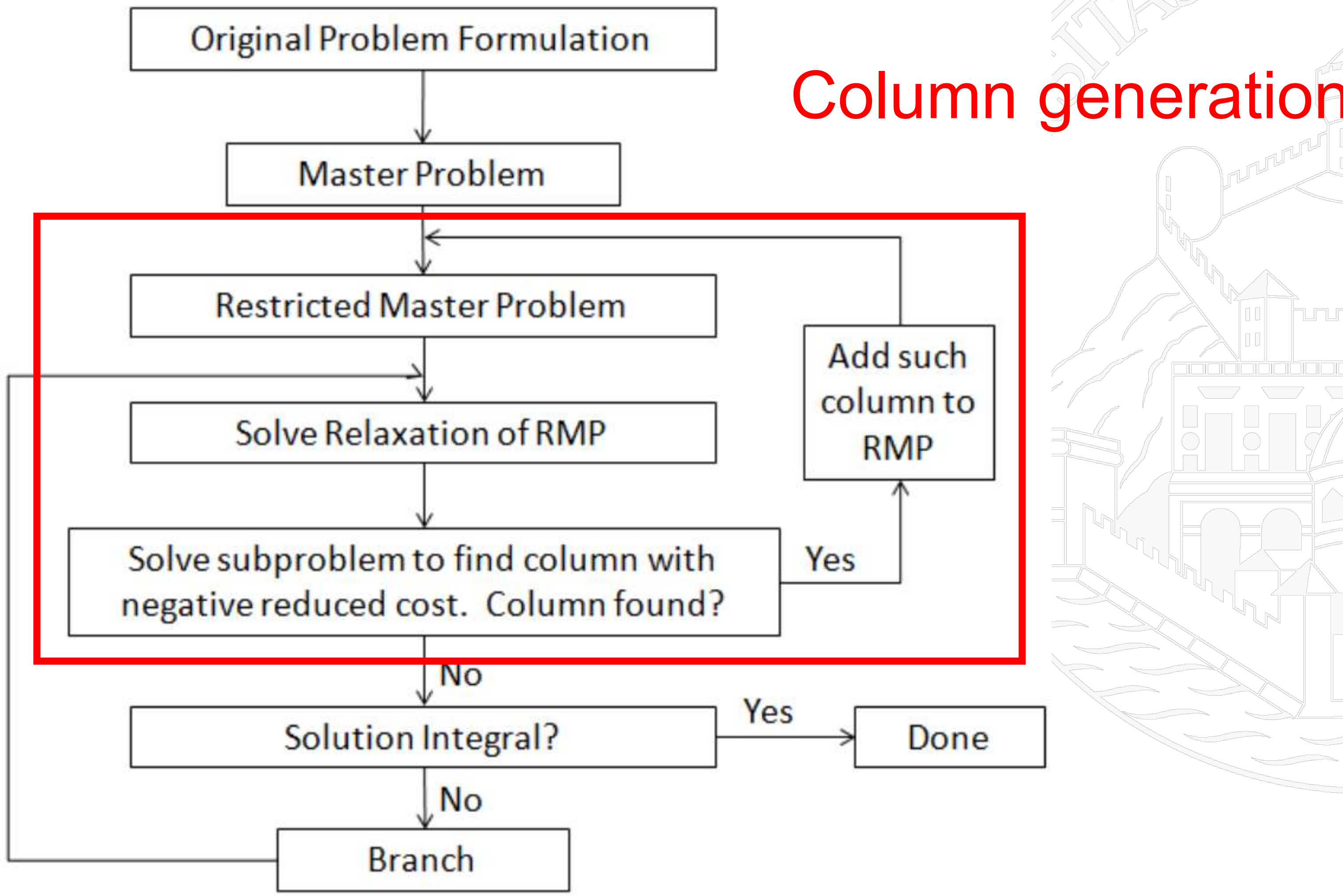
Branch-and-cut

Branch-and-price

$$\min c'x$$
$$Ax = b$$
$$x \geq 0 \;\; binary$$

Exponential number of variables

Branch-and-price



UNIVERSITY OF BRESCIA

MILP

More and more powerful software

UNIVERSITY OF BRESCIA

Mixed integer linear programming

Fixed costs

Scheduling

Location

Specialized algorithms
(exact and heuristic)

Selection

Routing

UNIVERSITY
OF BRESCIA