ORF307 – Optimization

22. The role of optimization

Bartolomeo Stellato — Spring 2022

Ed Forum

- Do we have new global upper/lower bounds for each iteration? Or, do we also take into account the local upper/lower bounds from previous iterations?

Final project

- Longer coding exercise (similar to coding in homeworks)
- Topics on the whole course:
  - Least-squares
  - Linear optimization
  - Integer optimization

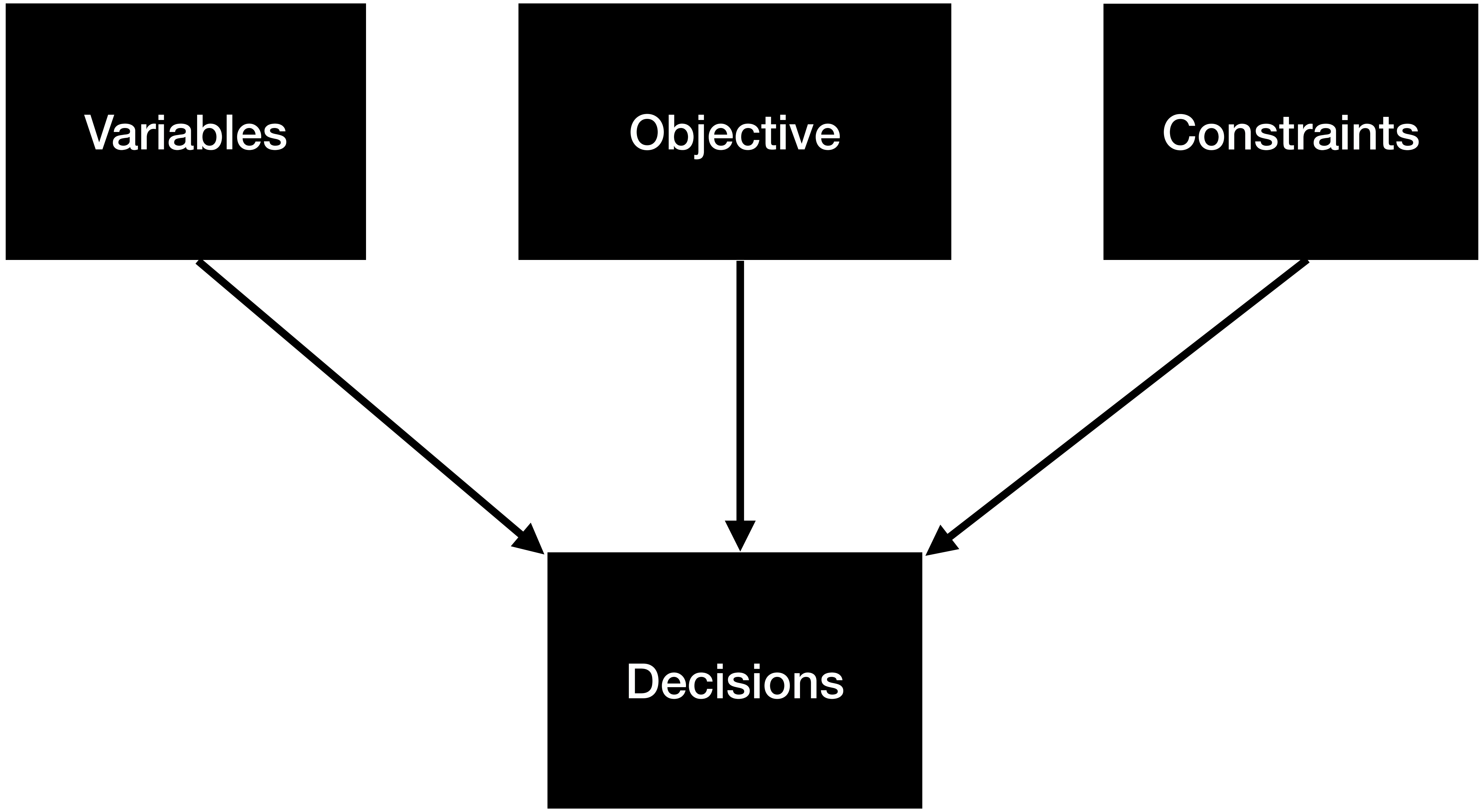
Today’s lecture

The role of optimization

- Geometry of optimization problems
- Solving optimization problems
- What’s left out there?
- The role of optimization

Basic use of optimization

Optimal decisions



Mathematical language

The algorithm computes them for you

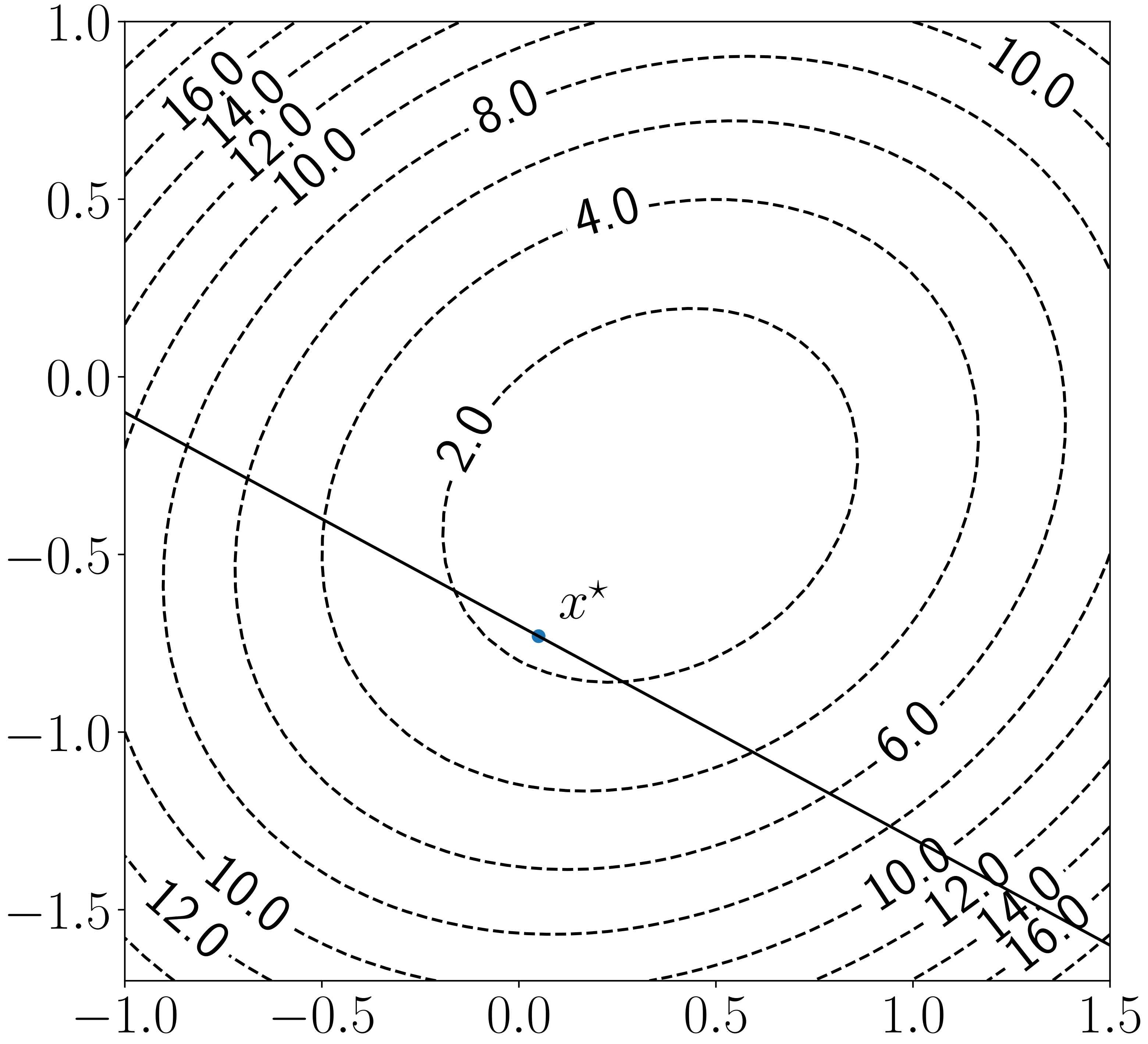
Most optimization problems cannot be solved

Geometry of optimization problems

Least squares

$$\begin{array}{ll} \text{minimize} & \overset{\overset{f(x)}{\downarrow}}{\|Ax-b\|^2} \\ \text{subject to} & Cx = d \end{array}$$

Example



$$\overset{A}{\begin{bmatrix} 2 & 0 \\ -1 & 1 \\ 0 & 2 \end{bmatrix}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \approx \overset{b}{\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}}$$

$$\overset{C}{\begin{bmatrix} 0.6 & 1 \end{bmatrix}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \overset{d}{\begin{bmatrix} -0.7 \end{bmatrix}}$$

$$x^\star = (0.05, -0.73)$$

Optimal point properties

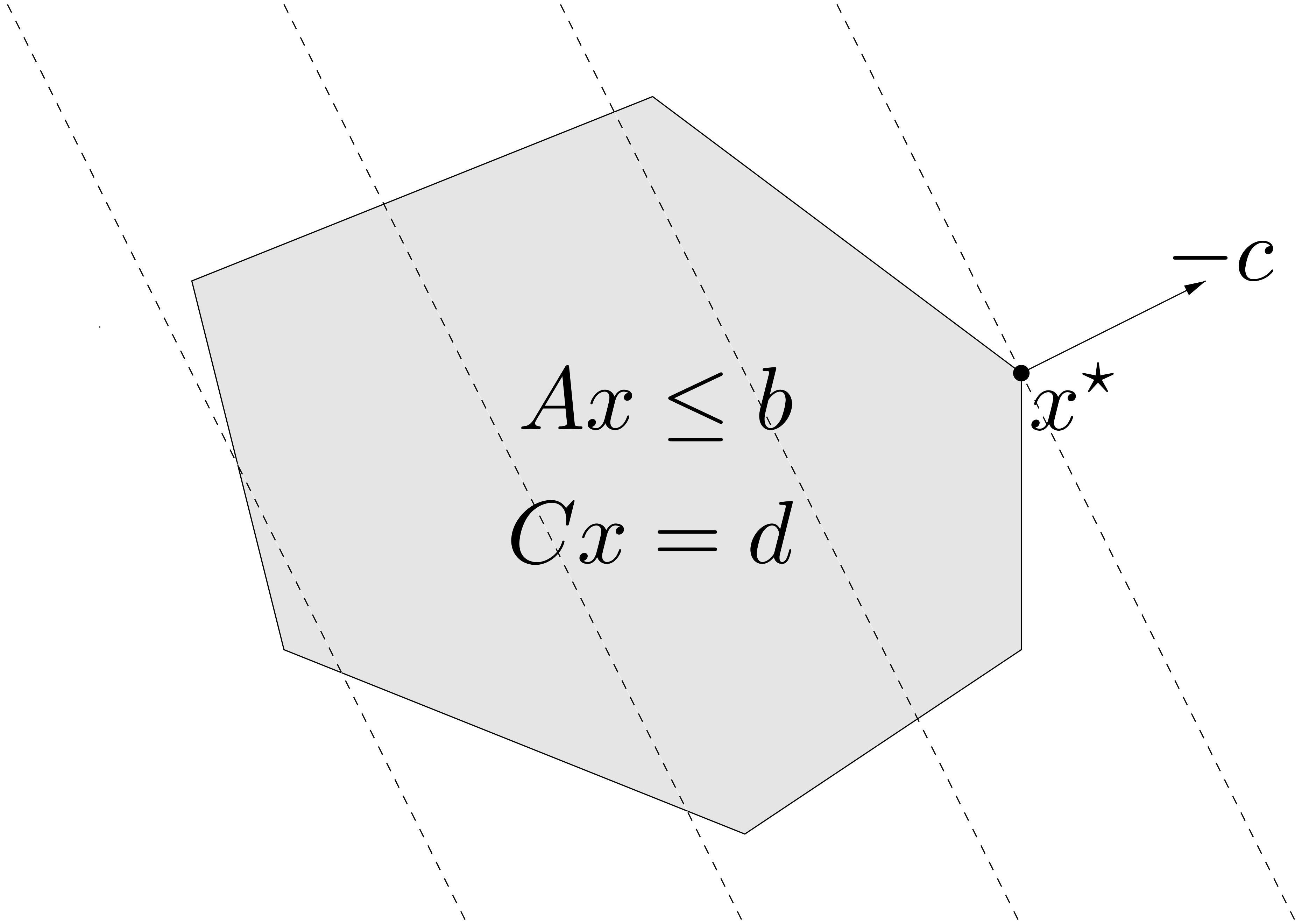
- Minimum point of $2x^T A^T A x - 2(A^T b)^T x$ over subspace $Cx = d$

Linear optimization

$$f(x)$$

$$\downarrow$$

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \\ & Cx = d \end{array}$$



Optimal point properties

- Extreme points are optimal
- Need to search only between extreme points

Duality

Dual function

$g(y)$ →

Properties

- Lower bound $g(y) \leq f(x)$
  (x primal and y dual feasible)
- Always concave
  (minimum of linear functions of x)

Strong duality

$$g(y^\star) = f(x^\star)$$

It holds unless primal and dual infeasible

Optimality conditions

Linear optimization

$$\begin{array}{ll} \text{minimize} & c^T x \longleftarrow f(x) \\ \text{subject to} & Ax \leq b \\ & Cx = d \end{array}$$

Least-squares

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|^2 \longleftarrow f(x) \\ \text{subject to} & Cx = d \end{array}$$

KKT optimality conditions

$$\nabla f(x^\star) + A^T y^\star + C^T z^\star = 0$$
$$y^\star \geq 0$$

dual feasibility

$$Ax^\star \leq b$$
$$Cx^\star = d$$

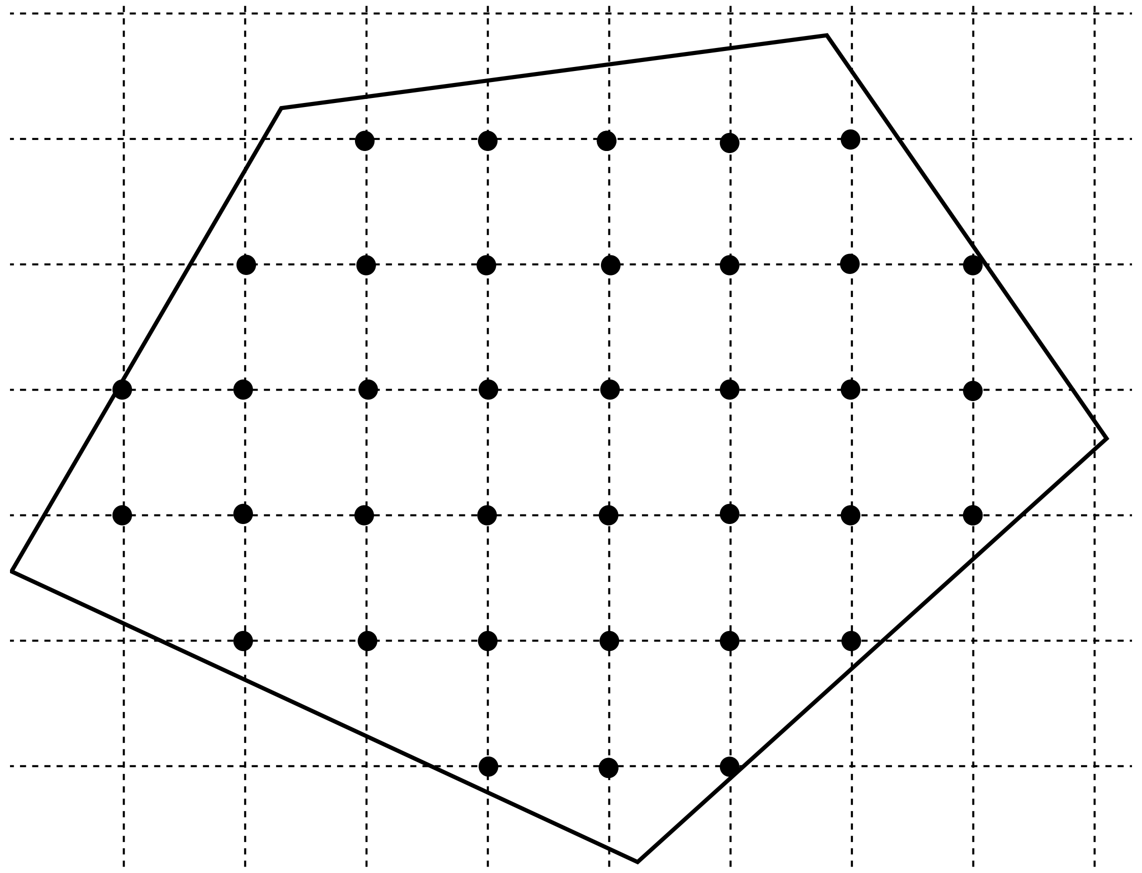
primal feasibility

$$y_i^\star (Ax^\star - b)_i = 0, \quad i = 1, \ldots, m$$

complementary slackness

Integer optimization

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \\ & x_i \in \mathbf{Z}, \quad i \in \mathcal{I} \end{array}$$



Optimal point properties

- Extreme points are not optimal in general
- If all integral variables, then finite set of solutions
- $x_i \in \mathbf{Z} \quad \Rightarrow$ Cannot use KKT optimality conditions

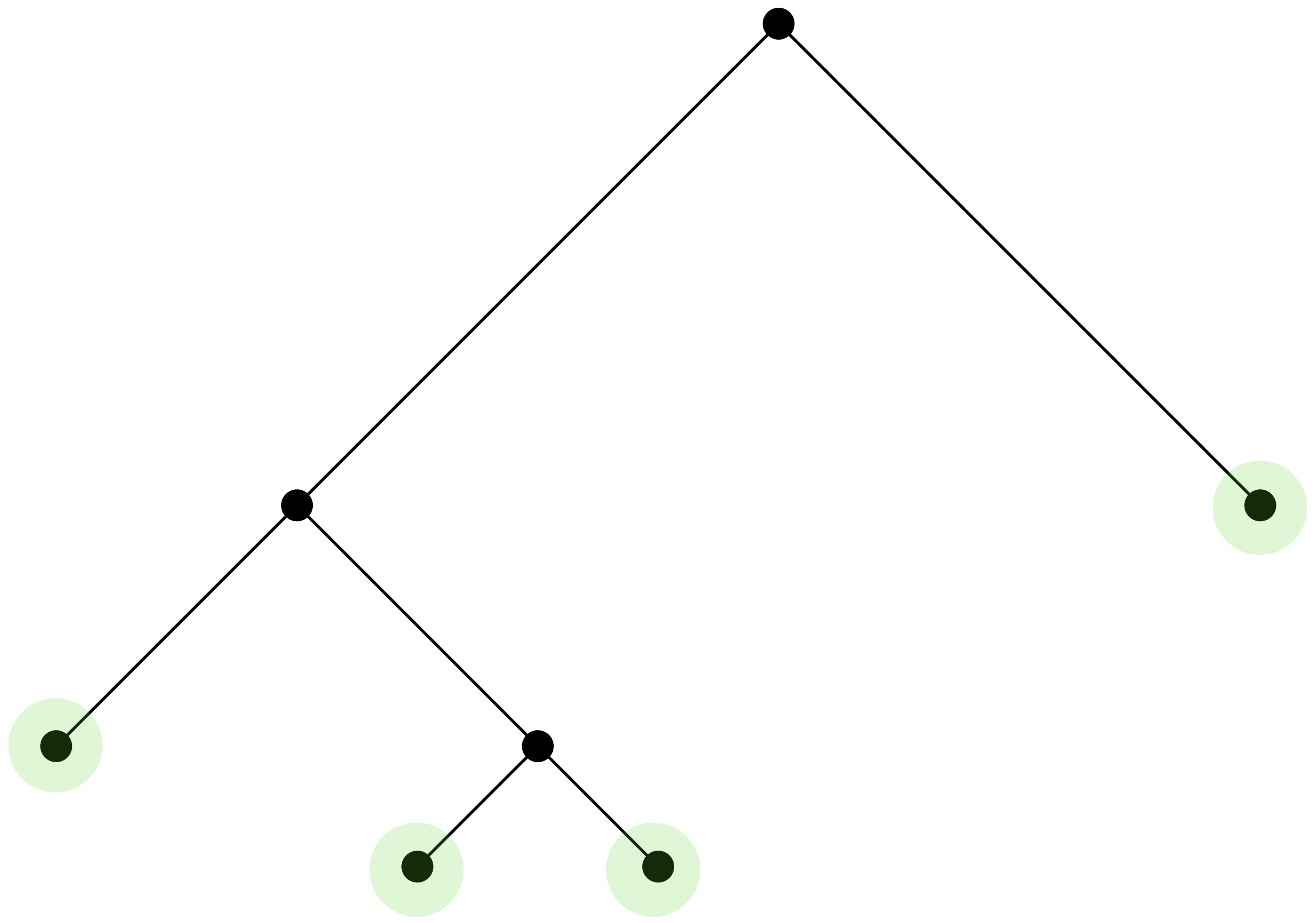
Optimality in integer optimization

certify optimality $\longrightarrow$ $L \leq c^T x^\star \leq U$ $\longleftarrow$ return feasible point "incumbent"

Lower bounds from direct relaxation

- Do not give integer feasible $\bar{x}$
- Different than the optimal objective $c^T x^\star$

Partition = Leaves



Optimality certificate in integer optimization

- Partition S^j
- Bounds $(L_j, U_j) \quad \forall j$

Solving optimization problems

Numerical linear algebra

The core of optimization algorithms is linear systems solution

$$Ax = b$$

Direct method

1. Factor $A = A_1 A_2 \ldots A_k$ in “simple” matrices ($O(n^3)$)
2. Compute $x = A_k^{-1} \ldots A_1^{-1} b$ by solving k “easy” linear systems ($O(n^2)$)

Main benefit

factorization can be reused
with different right-hand sides b

You **never** invert A

Solving least squares

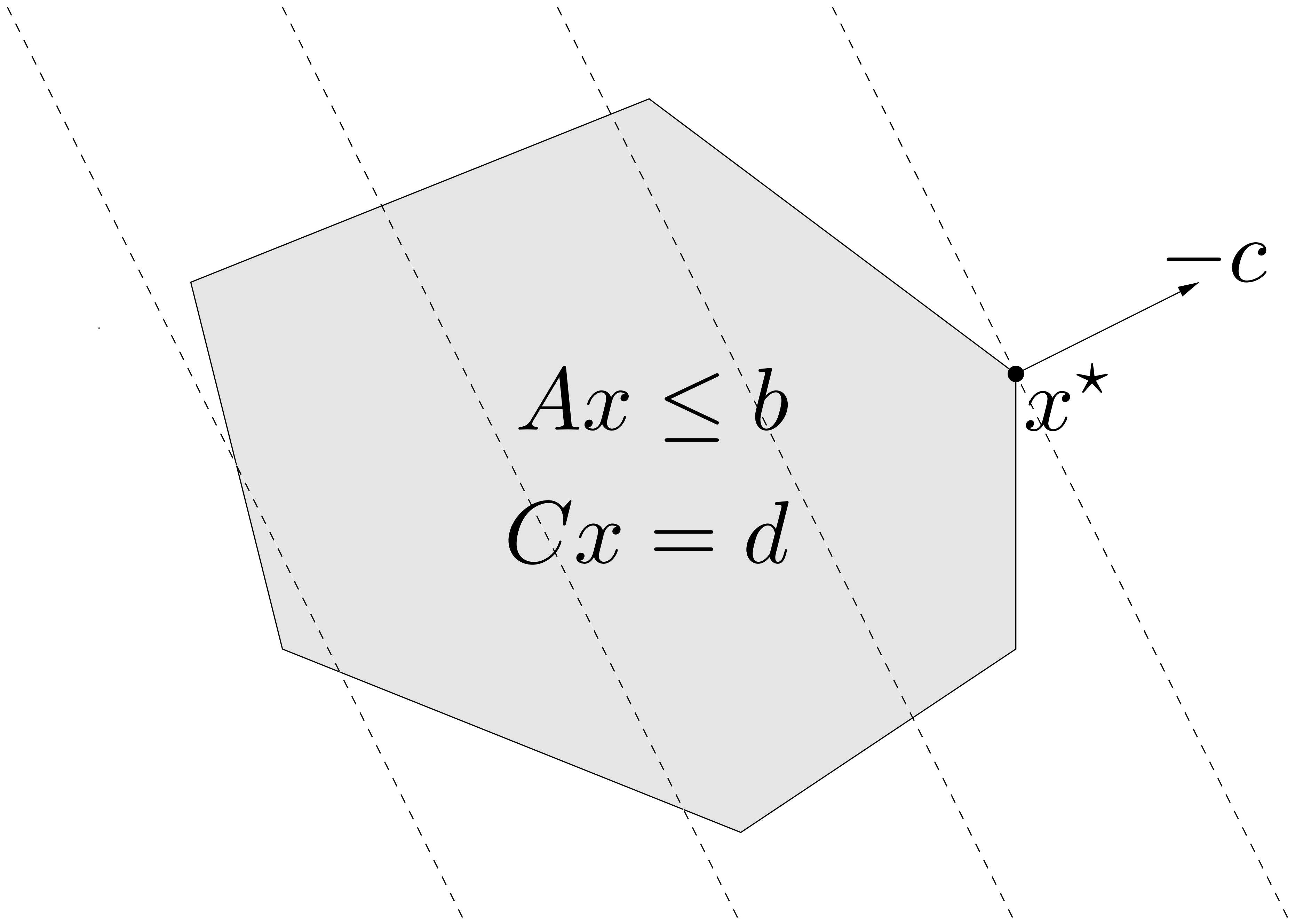
$$\begin{array}{ll} \text{minimize} & \|Ax - b\|^2 \\ \text{subject to} & Cx = d \end{array}$$

KKT linear system solution

$$\begin{bmatrix} 2A^T A & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} x^\star \\ z \end{bmatrix} = \begin{bmatrix} 2A^T b \\ d \end{bmatrix}$$

Solving linear optimization

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \le b \\ & Cx = d \end{array}$$

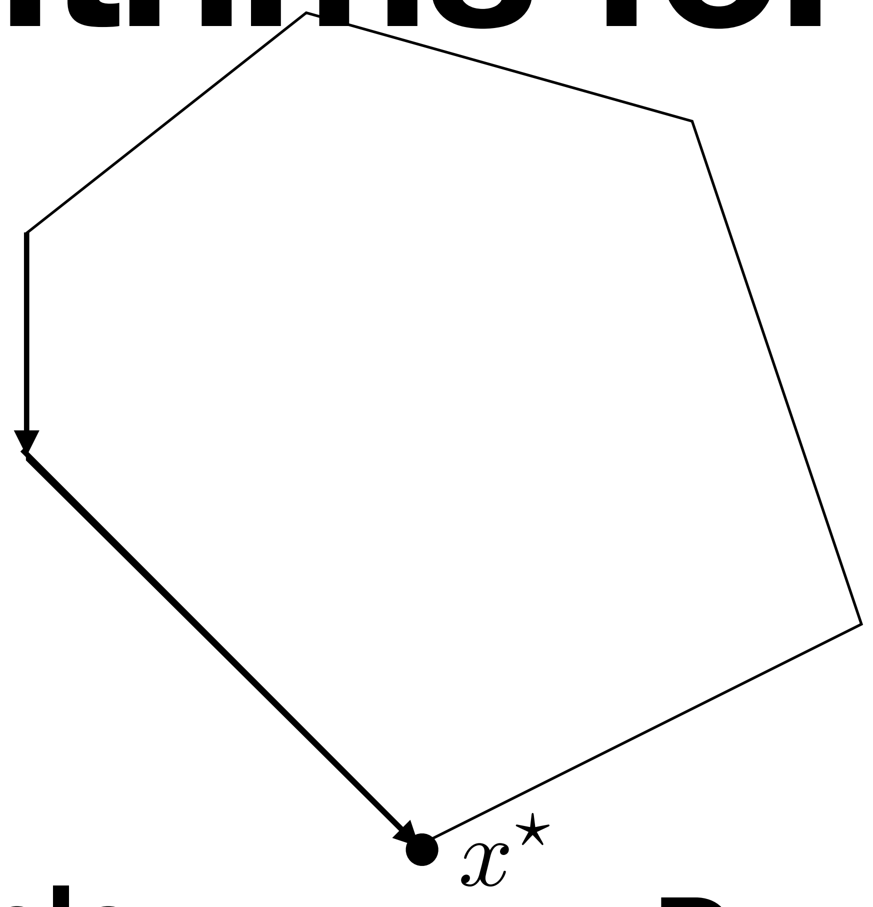


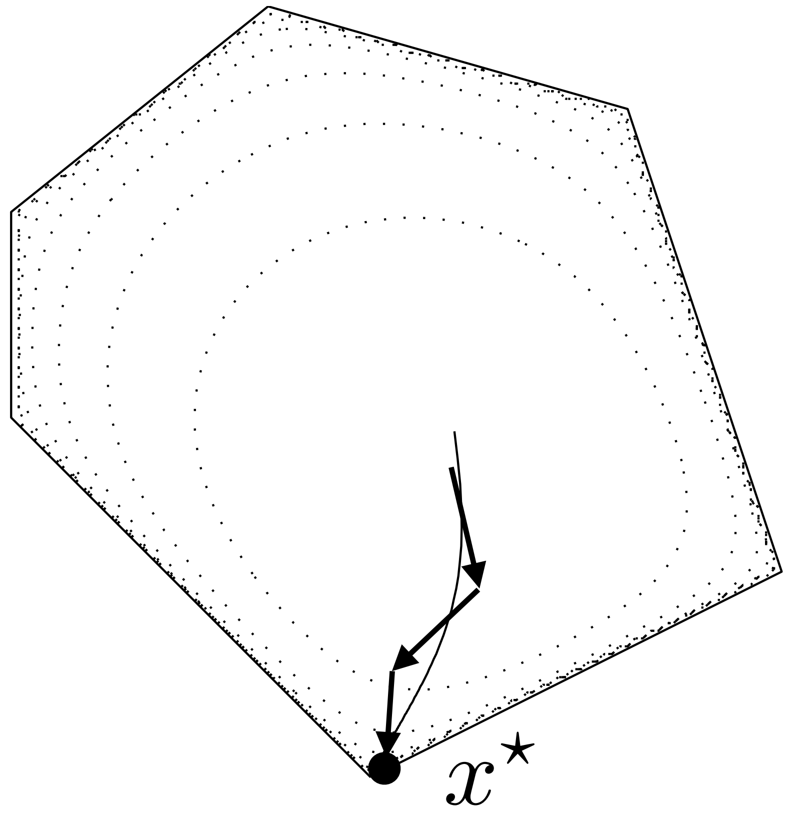
No closed form solution

↓

We need an iterative algorithm

Algorithms for linear optimization





Primal simplex

- Primal feasibility

↓

- Zero duality gap
- Dual feasibility

Dual simplex

- Dual feasibility

↓

- Zero duality gap
- Primal feasibility

Exponential worst-case complexity

Requires feasible point

Can be warm-started

Interior-point methods

- Interior condition

↓

- Primal feasibility
- Dual feasibility
- Zero duality gap

Polynomial worst-case complexity

Allows infeasible start

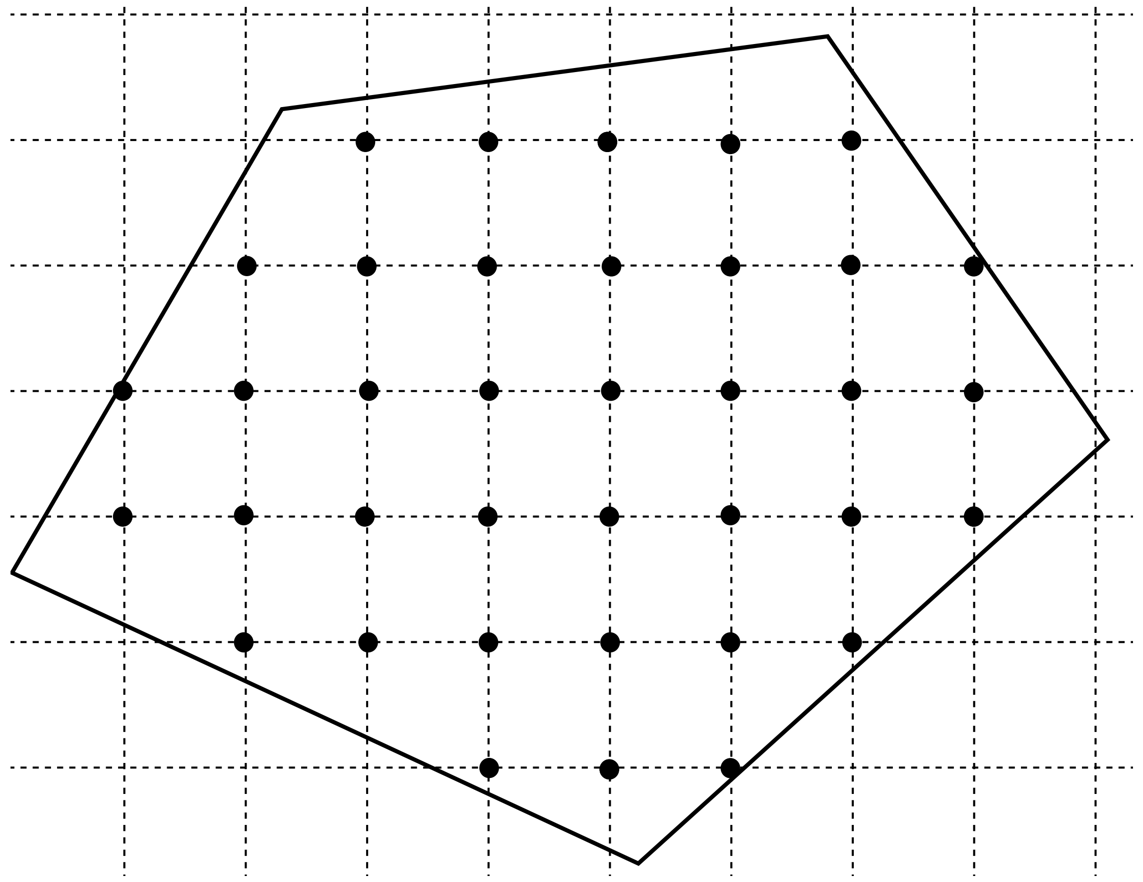
Cannot be warm-started

Linear optimization solvers

- Very **reliable** and **efficient** (many open source)
- Can solve problems in **milliseconds** on small processors
- **Simplex** and **interior-point solvers** are **almost a technology**
- **Used daily** in almost everywhere

Solving mixed-integer optimization

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \\ & x_i \in \mathbf{Z}, \quad i \in \mathcal{I} \end{array}$$

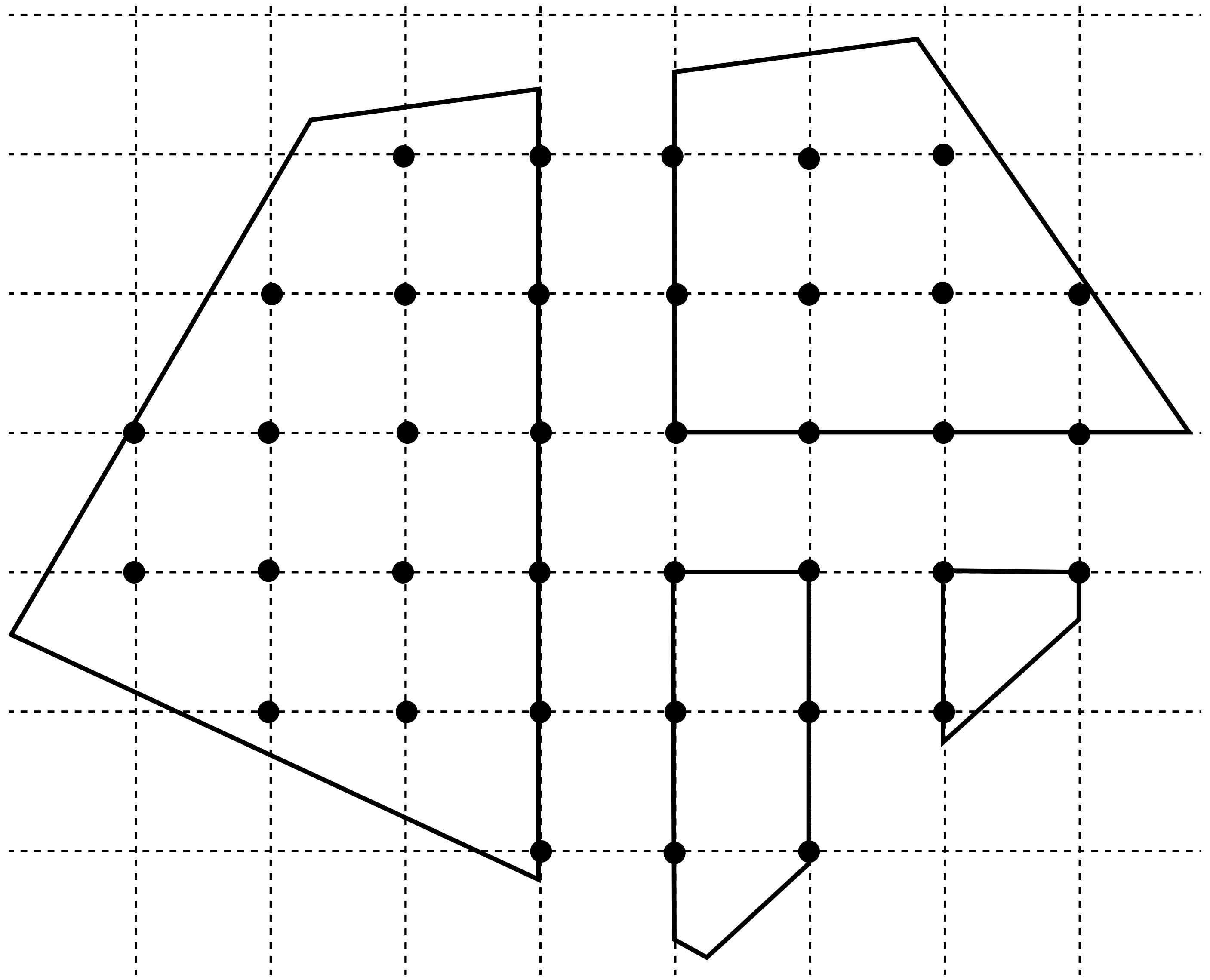


Relaxation does not always give feasible solutions ⟶ **Recursively partition the feasible space**

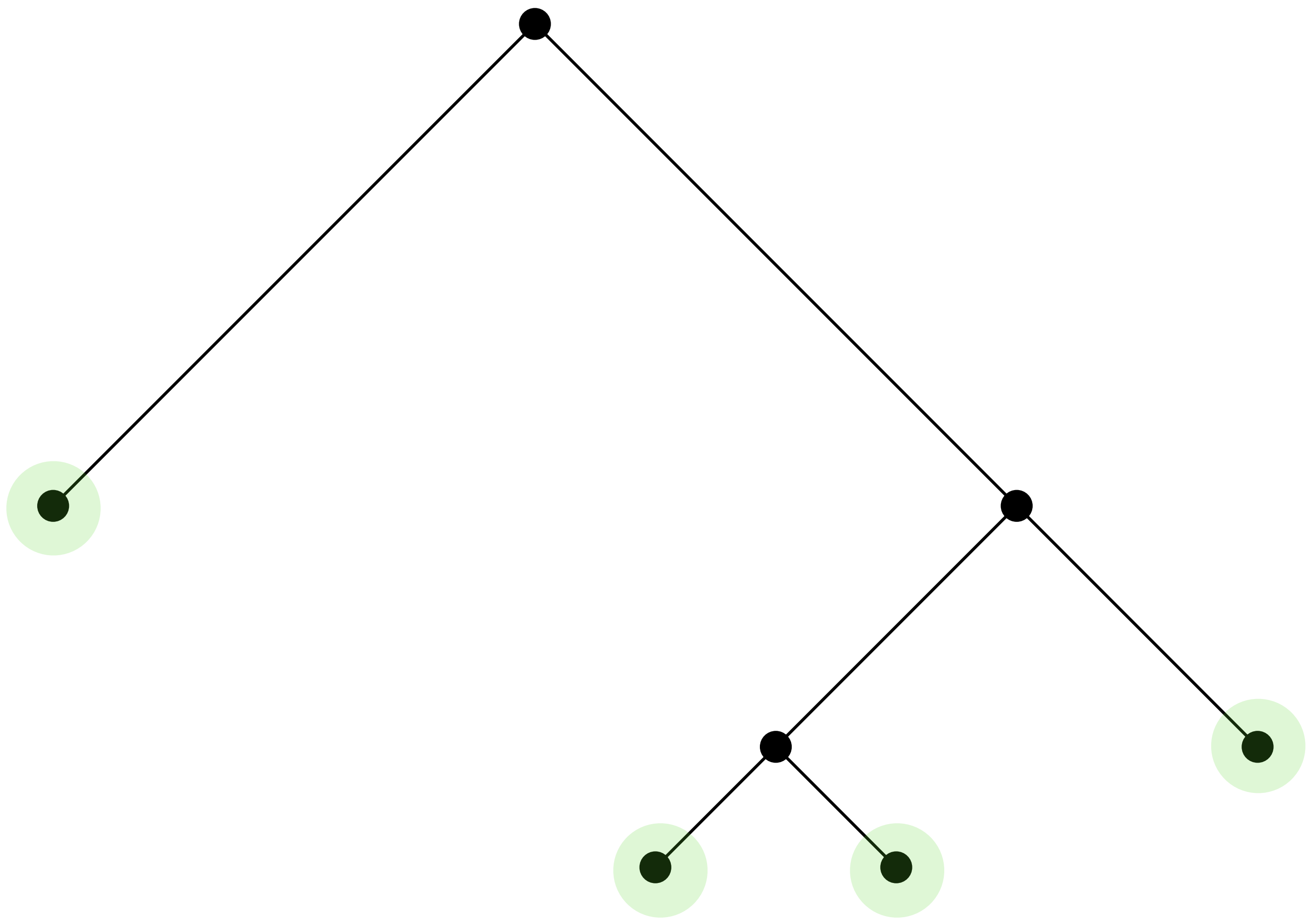
Algorithms for mixed-integer optimization

Branch and bound

Partition



Binary tree



Iteratively **branch** and **bound** until $U - L \leq \epsilon$

Mixed-integer optimization solvers

- Can be **slow** (the only very good ones are commercial)
- Recent **huge progress in hardware and software**
- Still **not a reliable technology**
- **Used daily** in almost everywhere

What’s left out there?

What we did not cover in continuous optimization?

Convex optimization

- Quadratic optimization
- Second-order cone optimization
- Semidefinite optimization
- Convex relaxations of combinatorial problems

→ Covered in
ORF363: Computing and
Optimization

Optimization applications

- Stochastic Optimization and ML in Finance (ORF311)
- Modern Control (MAE434)

What we did not cover in machine learning?

Machine learning

- Analysis of big data (ORF350)
- Fundamentals of machine learning (COS424)

Decision-making under uncertainty

- Optimal learning (ORF418)
- Foundations of Reinforcement Learning (ELE524)

The role of optimization

Optimization as a surrogate for real goal

Very often, optimization is not the actual goal

The goal usually comes from practical implementation (new data, real dynamics, etc.)

Real goal is usually encoded (approximated) in cost/constraints

Optimization problems are just models

> “All models are wrong, some are useful.”
>
> — George Box

Implications

- Problem formulation does not need to be “accurate”
- Objective function and constraints “guide” the optimizer
- The model includes parameters to tune

We often do not need to solve most problems to extreme accuracy

Data fitting

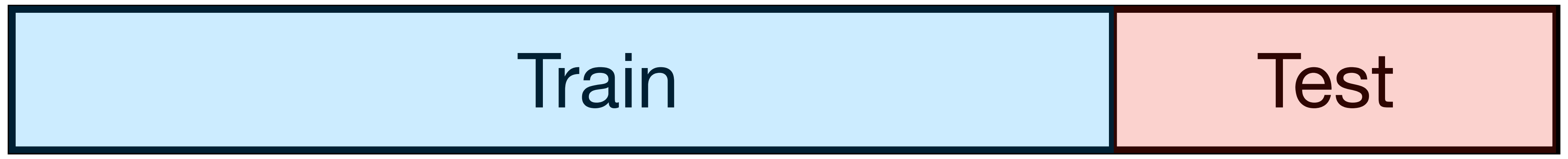
Goal learn model

$$y \approx f(x)$$

from **training data**

$(x^{(i)}, y^{(i)})$ for $i = 1, \ldots, N$

Data



Train

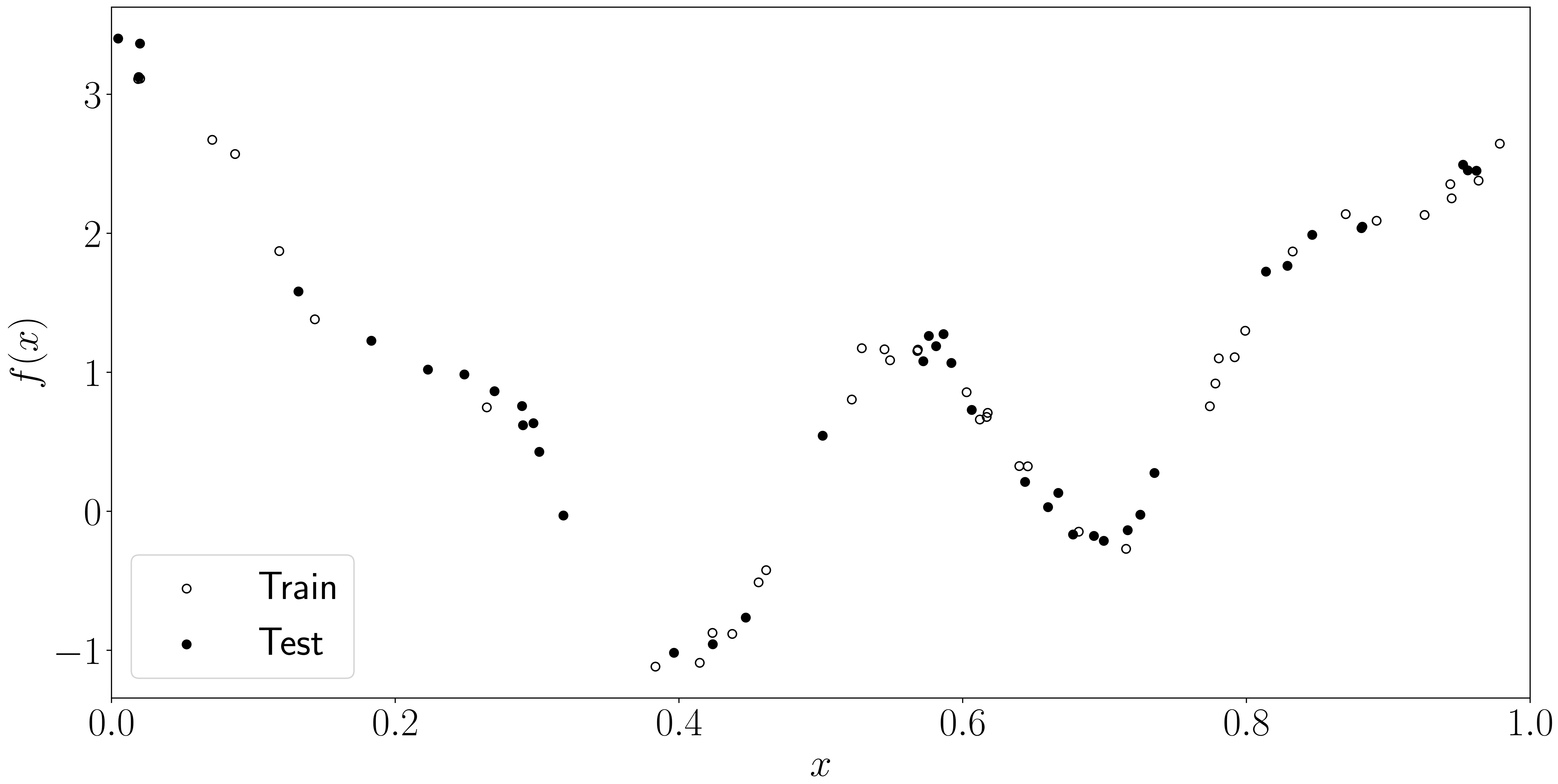
Test

- The goal of model is not to predict outcome for *given data* (Train)
- Instead, it is to predict the outcome on *new, unseen data* (Test)

↓

- A model bf generalizes if it makes reasonable predictions on unseen data
- A model **overfits** if it makes poor predictions on unseen data

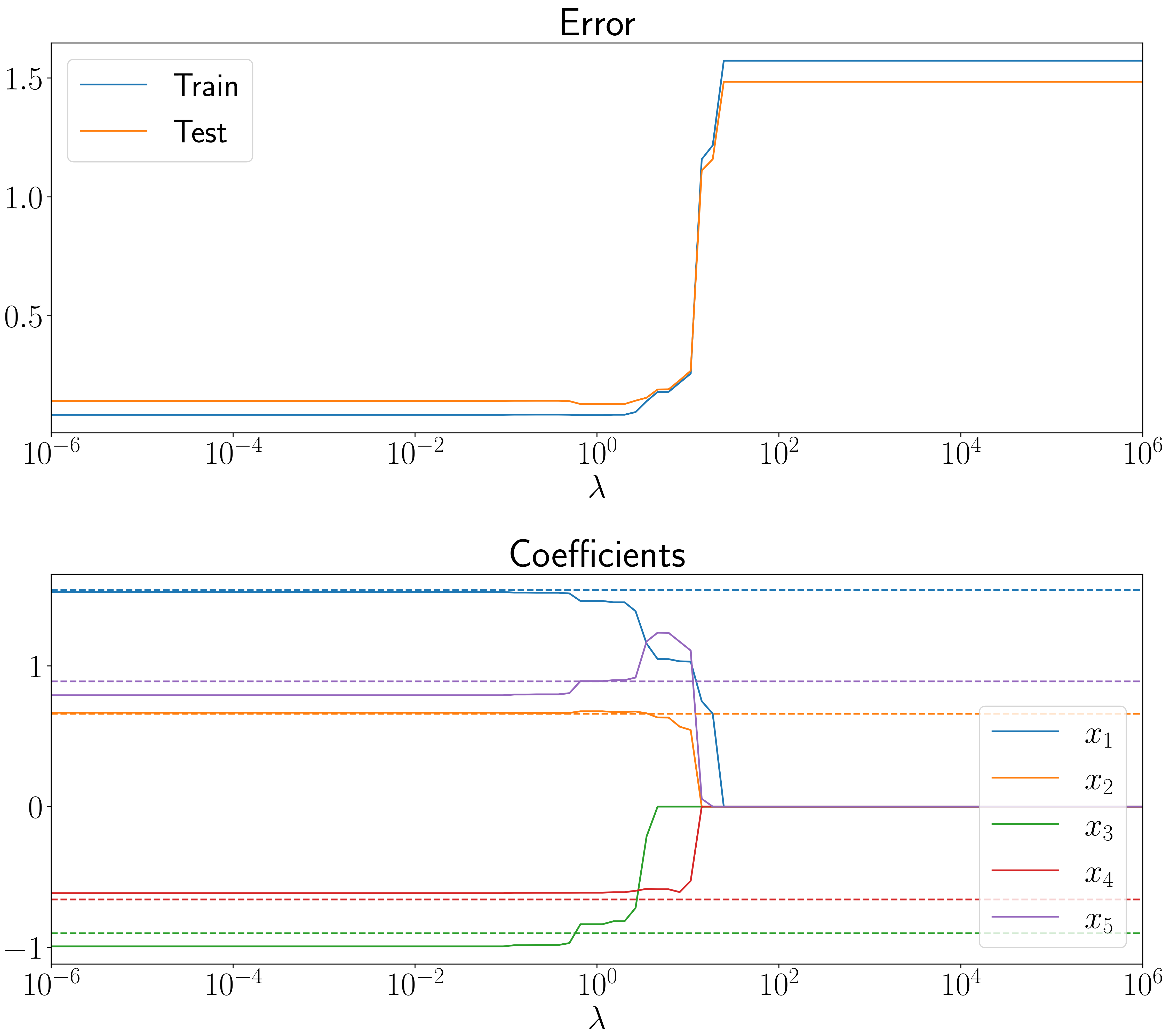
Regularization as proxy for generalization



Regularized fitting LP

minimize $\quad \|Ax - b\|_1 + \gamma \|x\|_1$ ⟵ Proxy

Train vs test error across regularization



Regularized fitting LP

minimize $\|Ax - b\|_1 + \lambda\|x\|_1$ ⟵ Proxy

- Minimum test error $\lambda \approx 1.15$
- Dashed lines: true values
- $x \to 0$ as $\lambda \to \infty$

Portfolio optimization

Goal: maximize average future returns

$$\mathbf{avg}(\tilde{R}w) = \tilde{\mu}^T w$$

from **historical returns**

$T \times n$ matrix of **asset returns**: R

Our model **generalizes** if a good w on past returns leads to good future returns

Example

- Pick w based on last 2 years of returns
- Use w during next 6 months

Portfolio optimization

Minimize risk-return tradeoff
(on historical data)

$$\begin{array}{ll} \text{minimize} & \overset{\substack{\textbf{Returns}\\ \downarrow}}{-\mu^T w} + \overset{\substack{\textbf{Risk}\\ \downarrow}}{\gamma\|Rw - \mu^T w\mathbf{1}\|_1} \longleftarrow \text{Proxy} \\ \text{subject to} & \mathbf{1}^T w = 1 \\ & w \geq 0 \end{array}$$

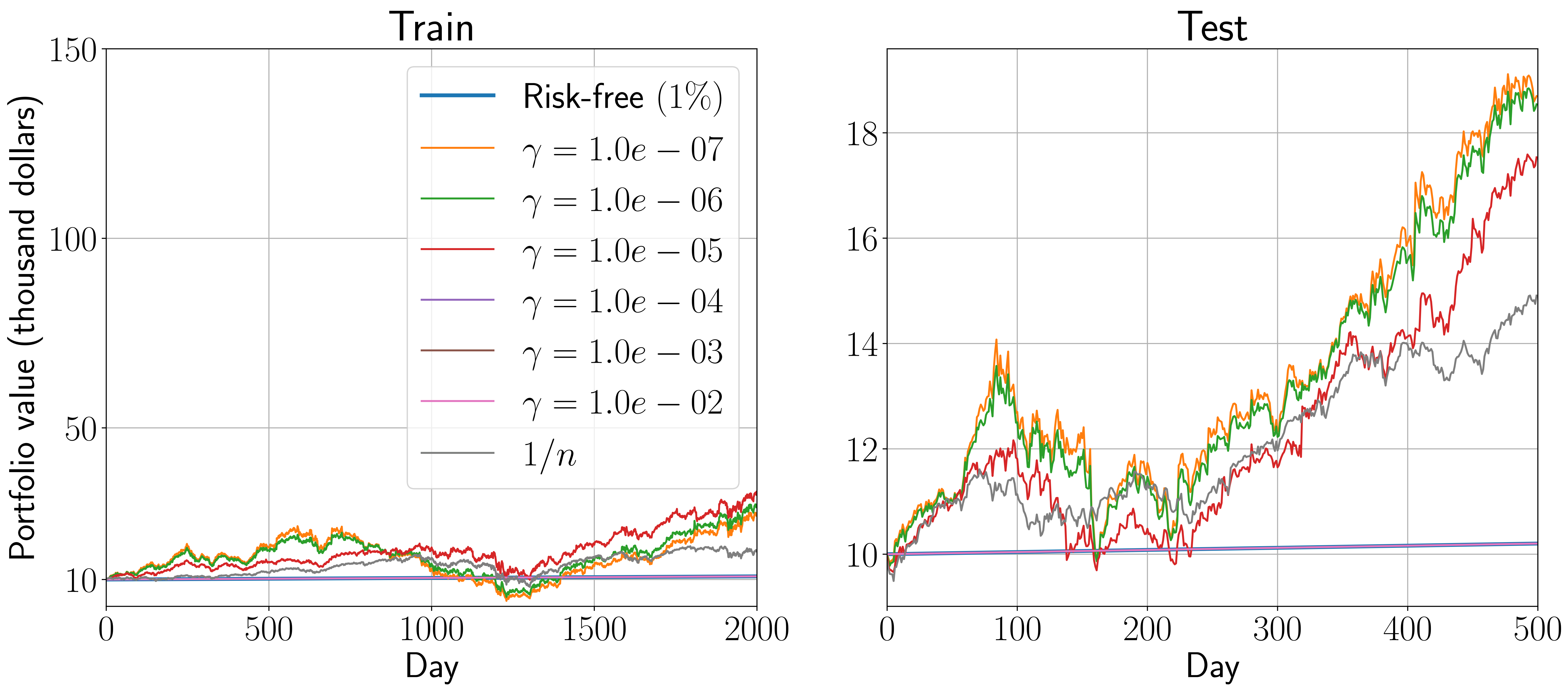
Risk is a proxy to perform well in the future

Past vs future returns on portfolio optimization

Minimize risk-return tradeoff

$$\begin{array}{ll} \text{minimize} & -\mu^T w + \gamma \|Rw - \mu^T w \mathbf{1}\|_1 \\ \text{subject to} & \mathbf{1}^T w = 1 \\ & w \geq 0 \end{array}$$

← Proxy



- As $\gamma \to 0$, more aggressive
- As $\gamma \to \infty$, risk-averse
- Future is unclear

Conclusions

In ORF307, we learned to:

- **Model decision-making problems** across different disciplines as mathematical optimization problems.
- **Apply the most appropriate optimization tools** when faced with a concrete problem.
- **Implement** optimization algorithms
- **Understand** the limitations of optimization

Optimization cannot solve all our problems

It is just a mathematical model

But it can help us making better decisions

Thank you!

Bartolomeo Stellato