ORF307 – Optimization

22. The role of optimization

Bartolomeo Stellato — Spring 2023

Ed Forum

- I was wondering what some practical examples were of cardinality minimization – i.e. how might this approach be employed in the real world?

LEAST SQUARES DATA FITTING

$$\min \|Ax - b\|_2^2$$

$$\text{s.t.} \quad \mathrm{card}(x) \leq k$$

Announcements

Participation

- Please send last note by the end of this weekend

Final Project

- Last year’s project out
- Longer coding exercise (similar to coding in homeworks)
- Topics on the whole course:
  - Least-squares
  - Linear optimization
  - Integer optimization

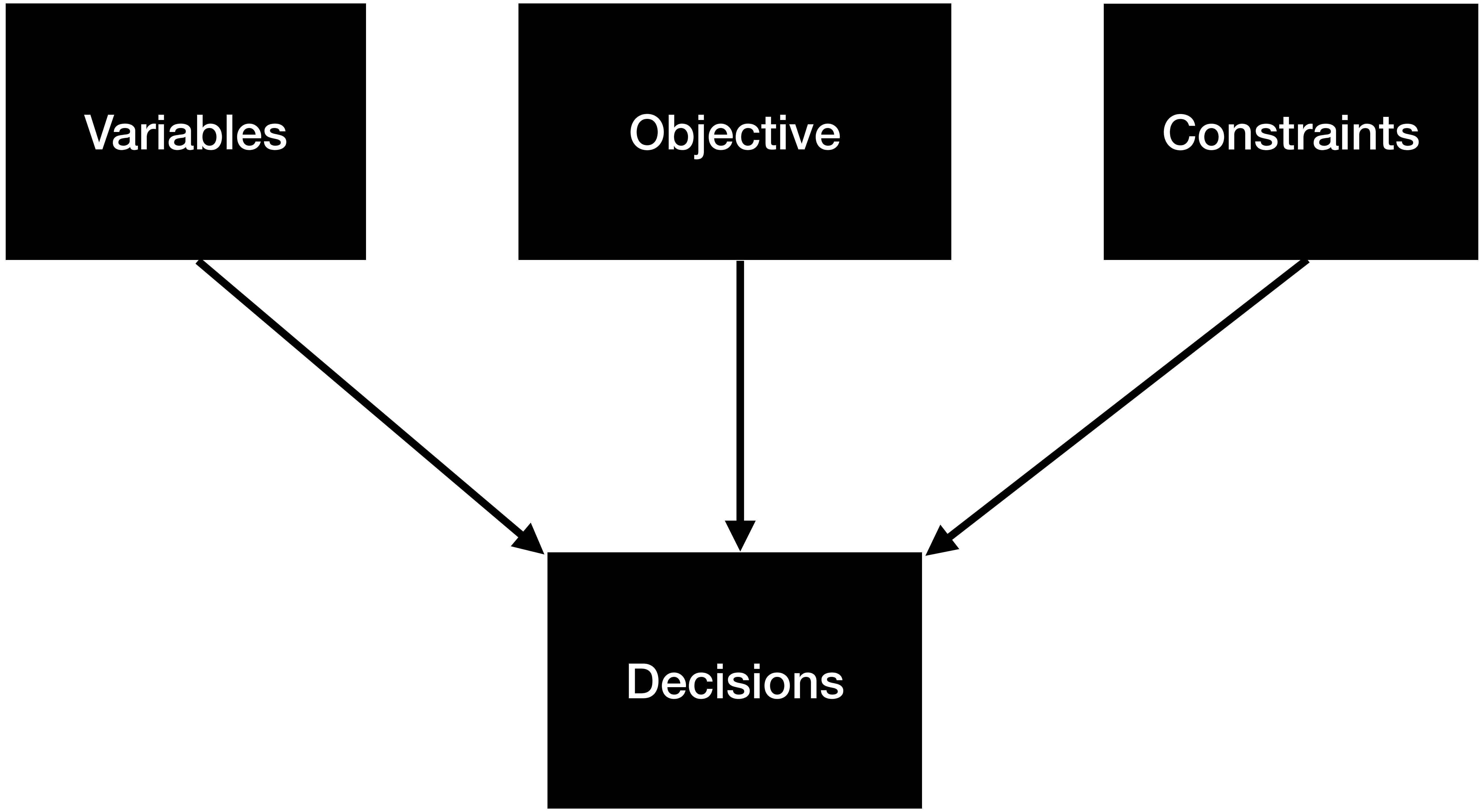
Today’s lecture

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

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

Least squares

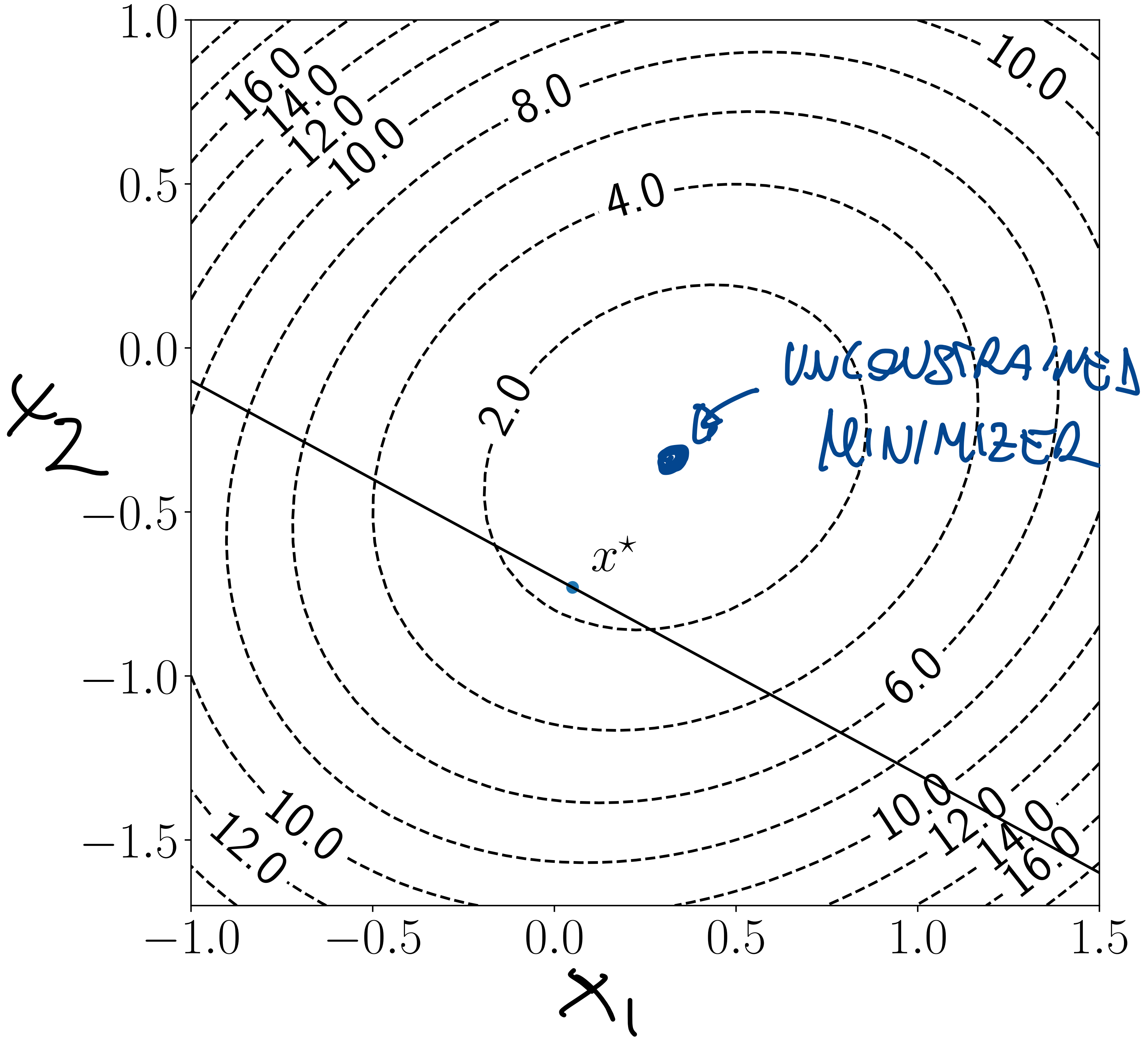
$$f(x)$$

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

Least squares

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

Example



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

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

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

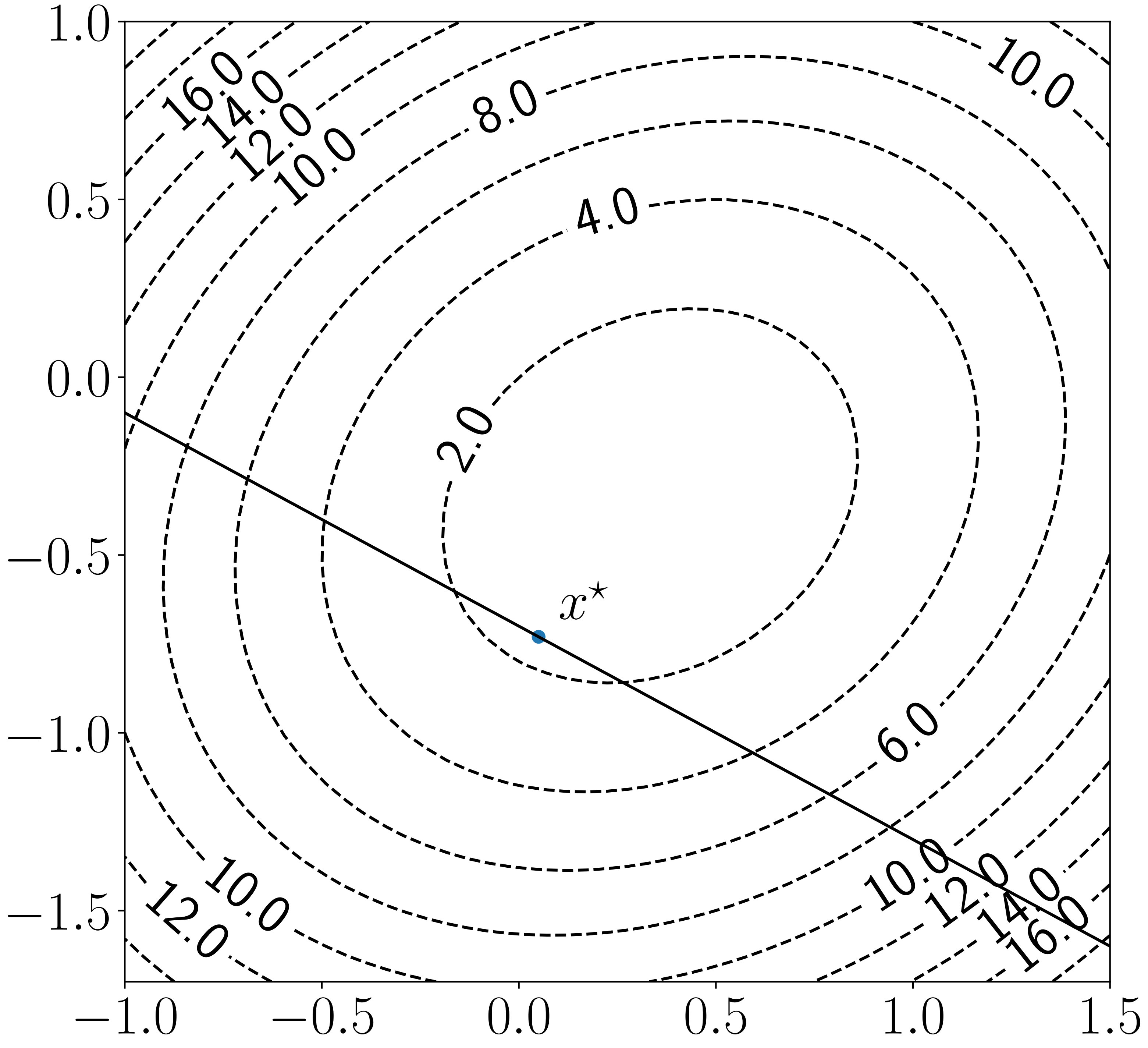
Least squares

$f(x)$

minimize $\|Ax - b\|^2$
subject to $Cx = d$

$f(x) = (Ax - b)^T (Ax - b)$

Example



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

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

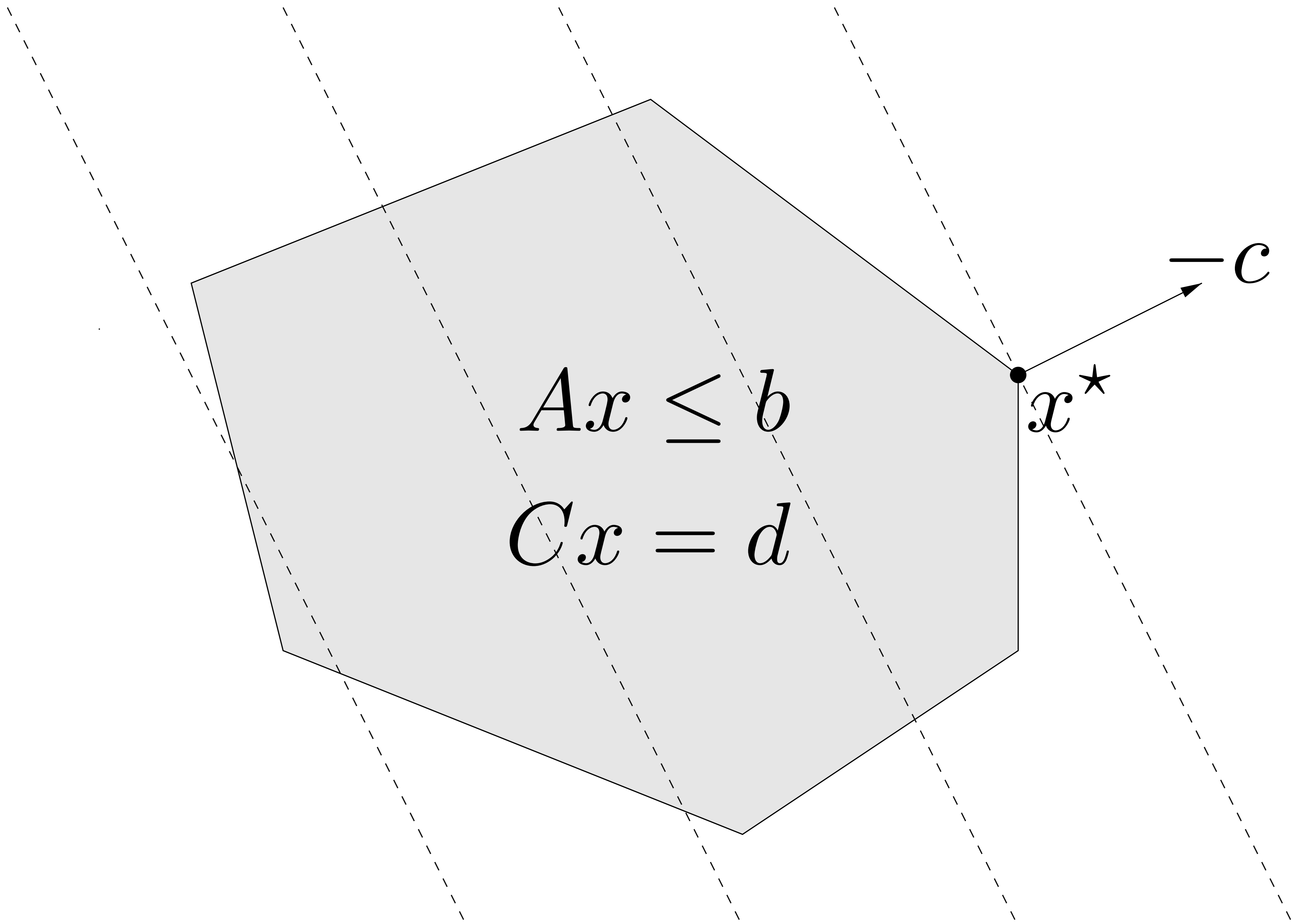
$$x^\star = (0.05, -0.73)$$

Optimal point properties

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

Linear optimization

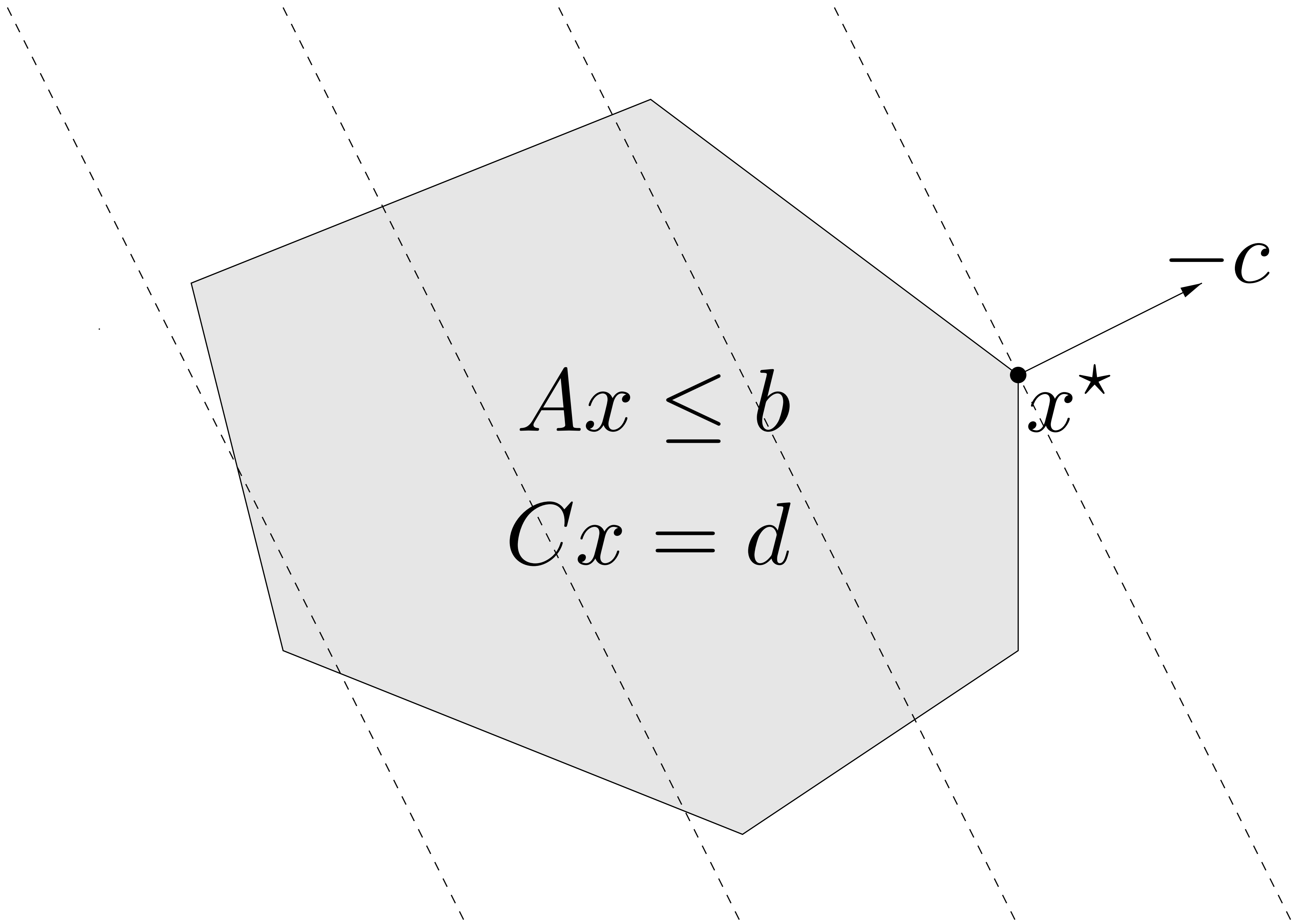
$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \\ & Cx = d \end{array}$$



Linear optimization

$$f(x)$$
$$\downarrow$$

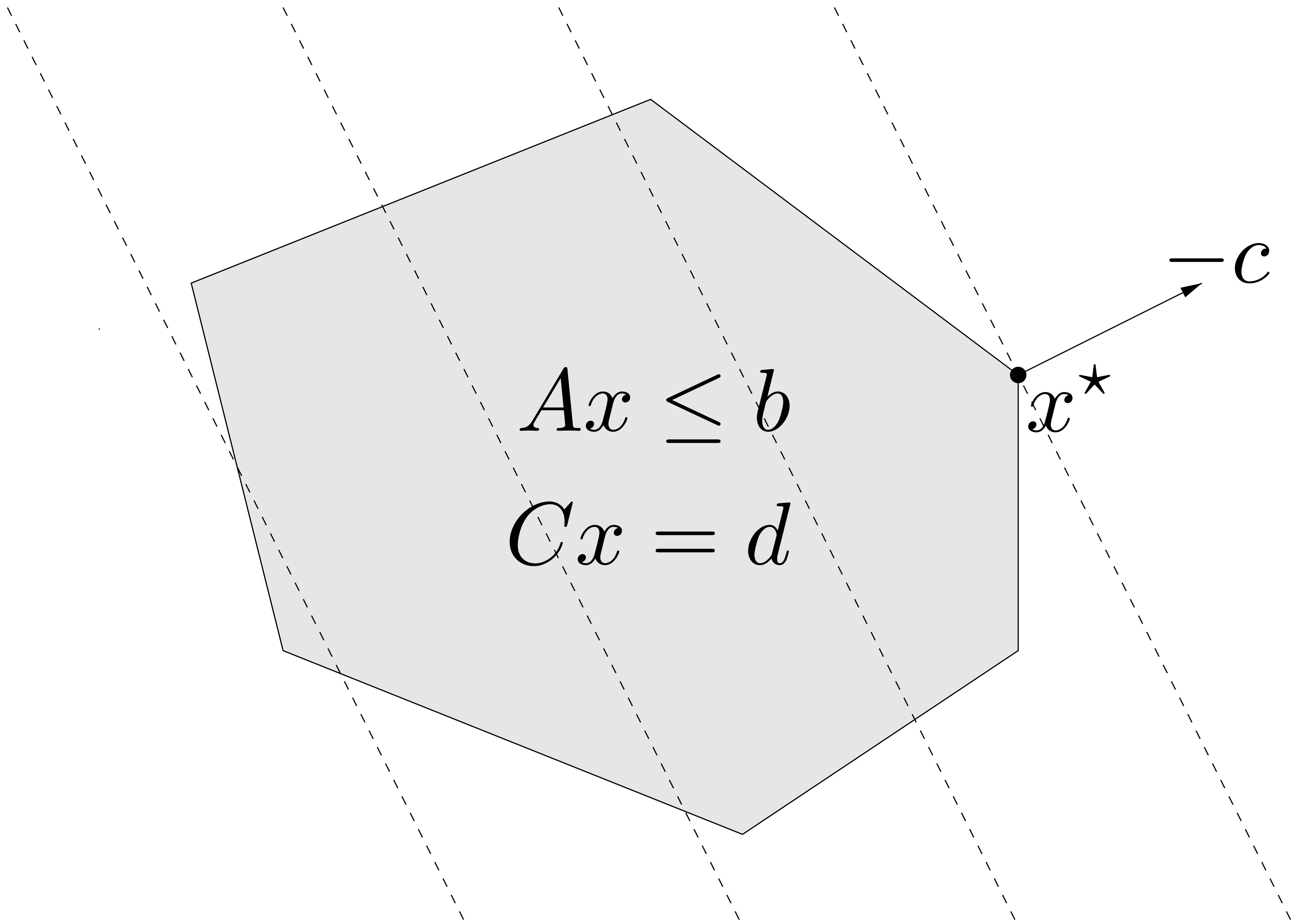
$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \\ & Cx = d \end{array}$$



Linear optimization

$$f(x)$$

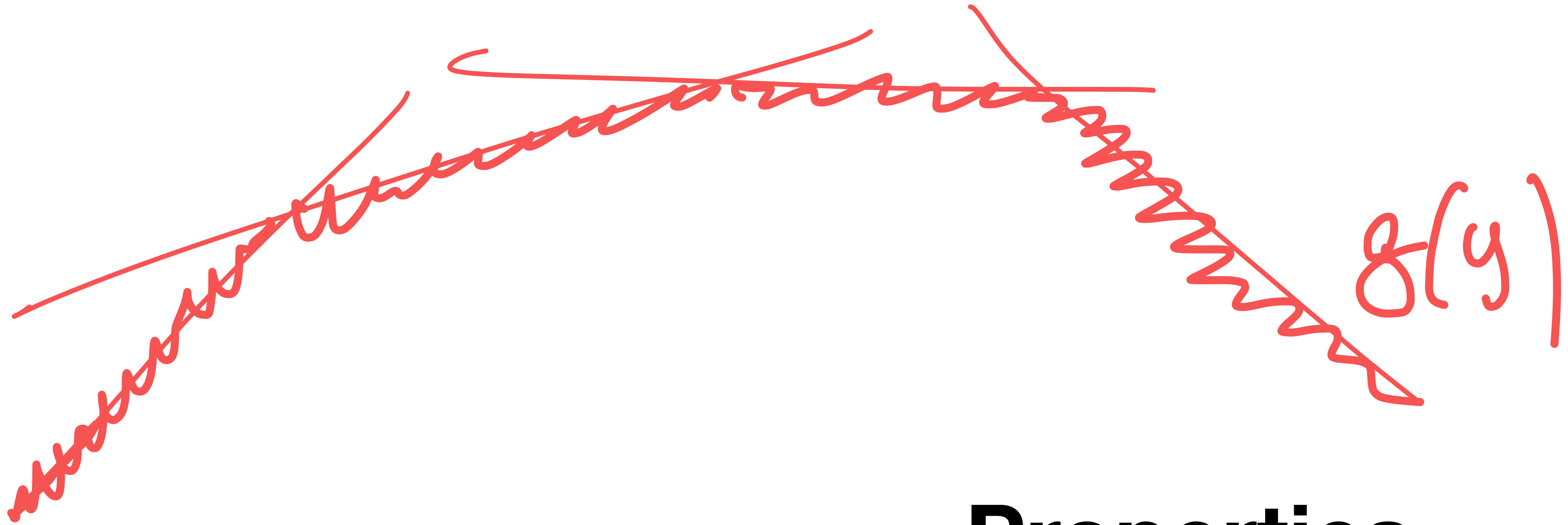
$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \\ & Cx = d \end{array}$$



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)

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 y)

Strong duality

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

It holds unless primal and dual infeasible

Optimality conditions

Linear optimization

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

Least-squares

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

Optimality conditions

$\nabla f(x) = 2A^TAx - 2(A^Tb)^Tx$

Linear optimization

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

Least-squares

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

KKT optimality conditions

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

dual feasibility

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

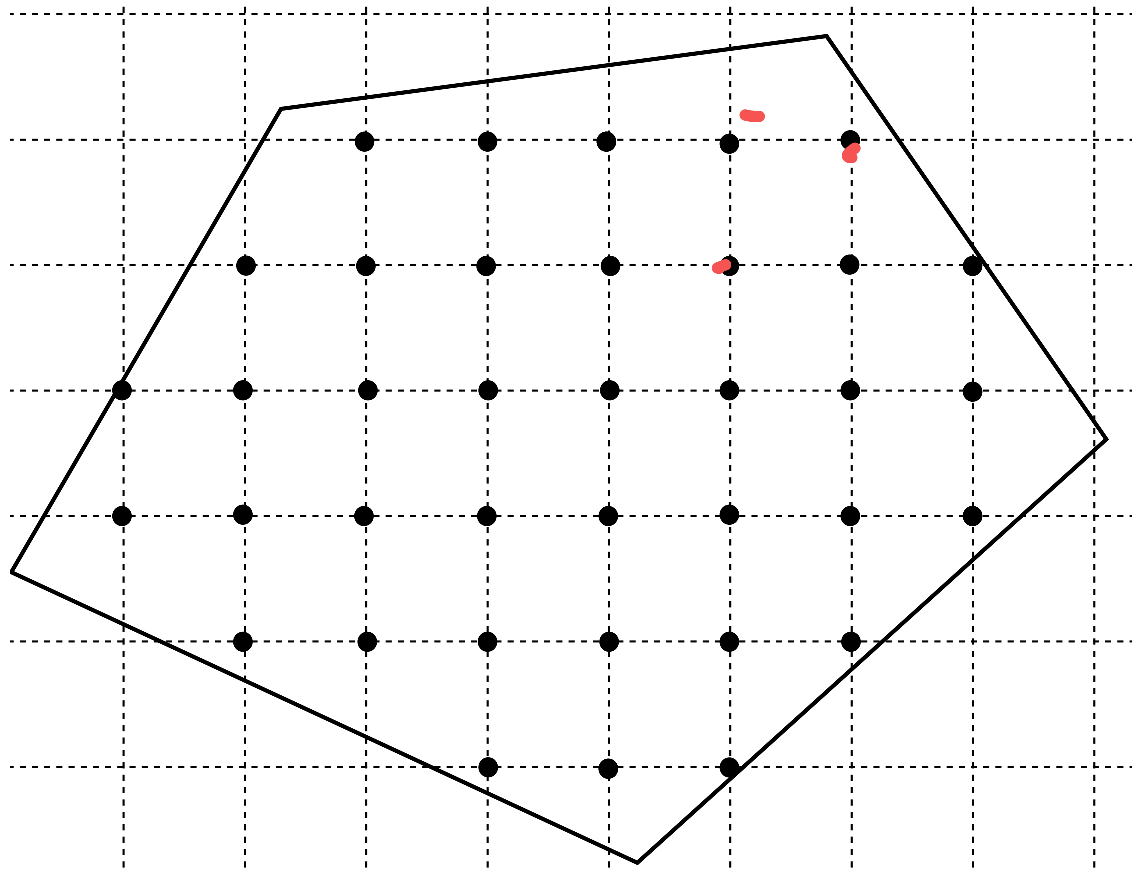
primal feasibility

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

complementary slackness

Integer optimization

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



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 ⟶ $L \leq c^T x^\star \leq U$ ⟵ return feasible point "incumbent"

Lower bounds from direct relaxation

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

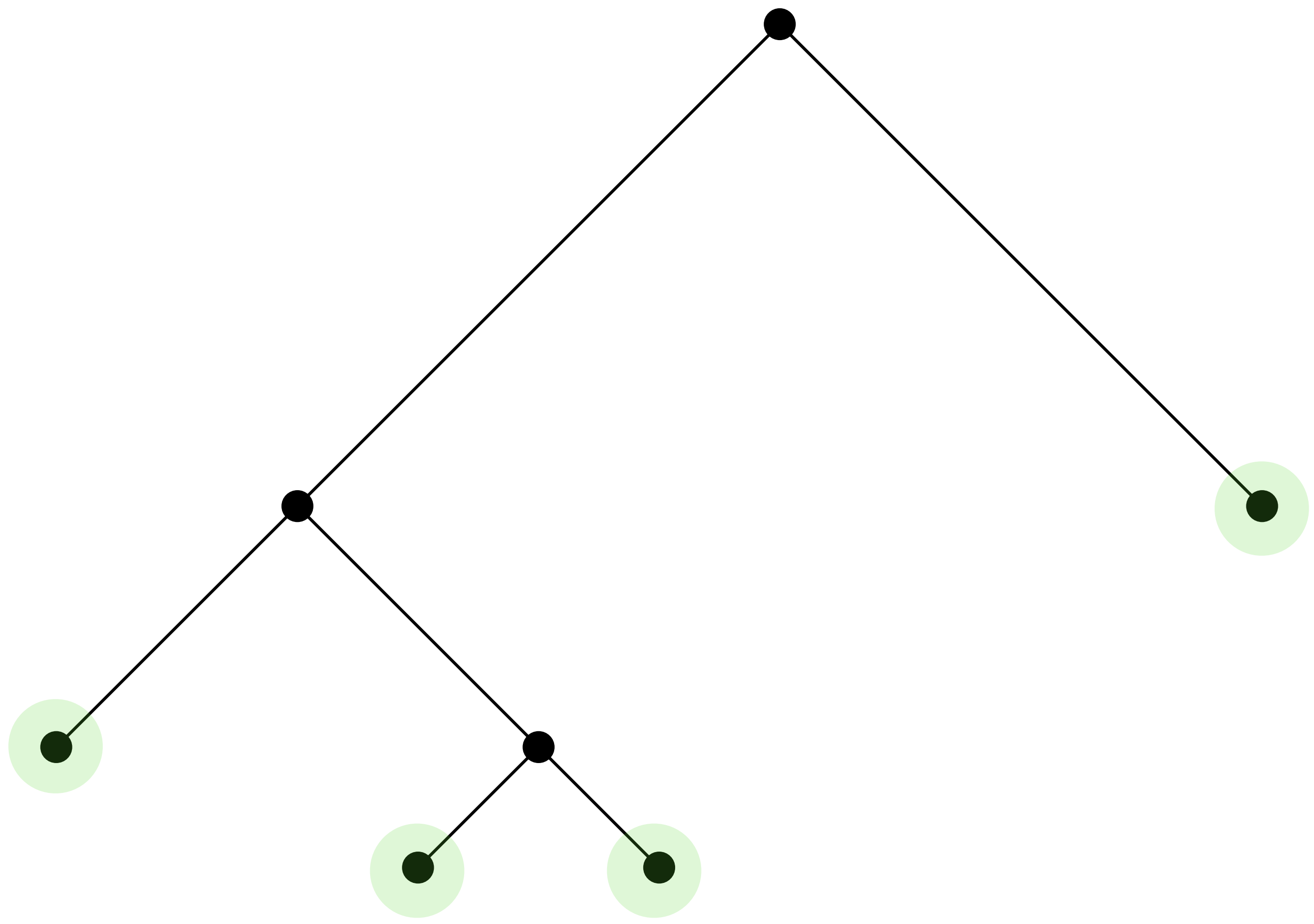
Optimality in integer optimization

certify optimality ⟶ $L \leq c^T x^\star \leq U$ ⟵ 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



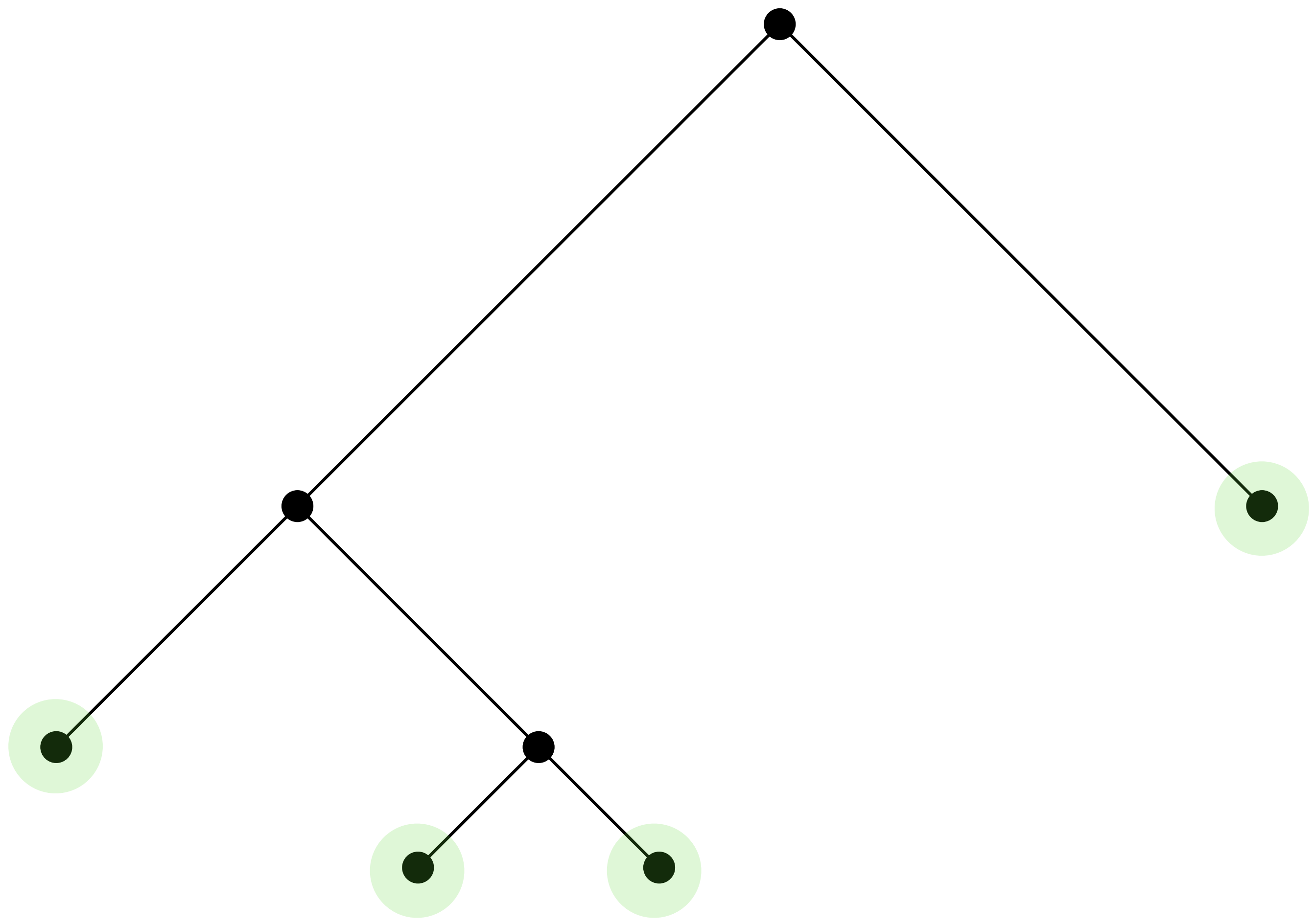
Optimality in integer optimization

certify optimality ⟶ $L \leq c^T x^\star \leq U$ ⟵ 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 ($O(n^2)$)

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

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

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

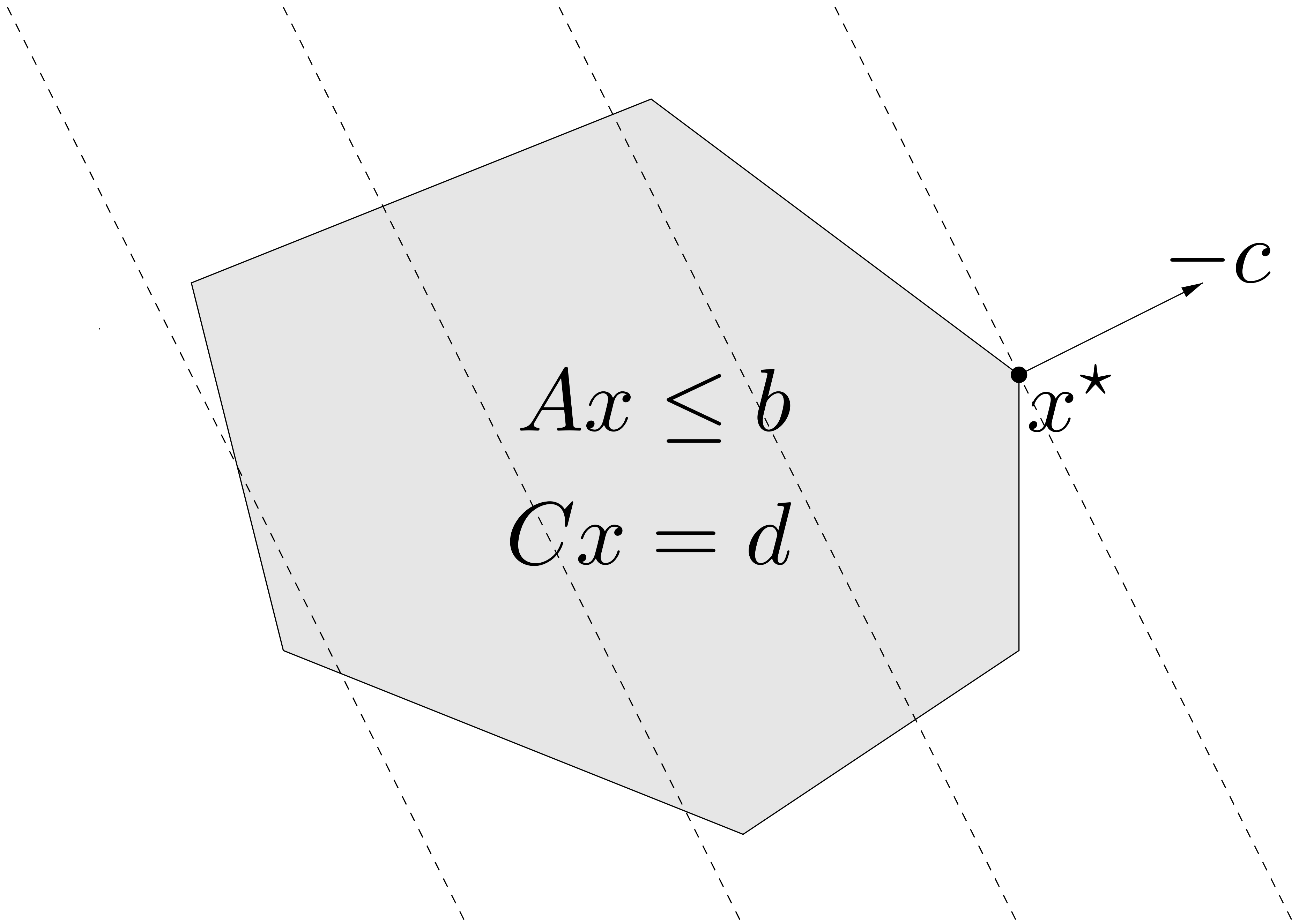
(2)

KKT linear system solution

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

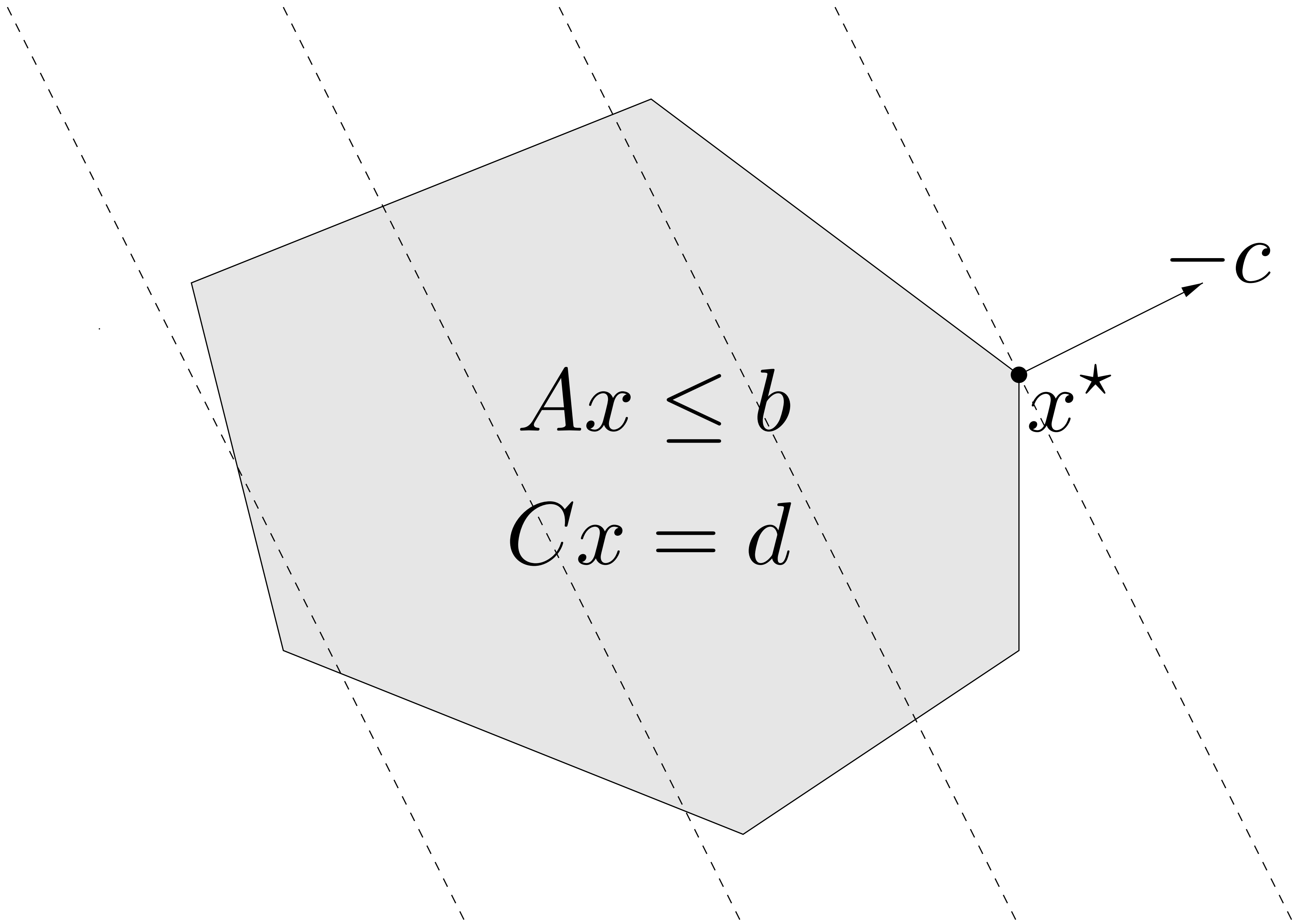
Solving linear optimization

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



Solving linear optimization

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

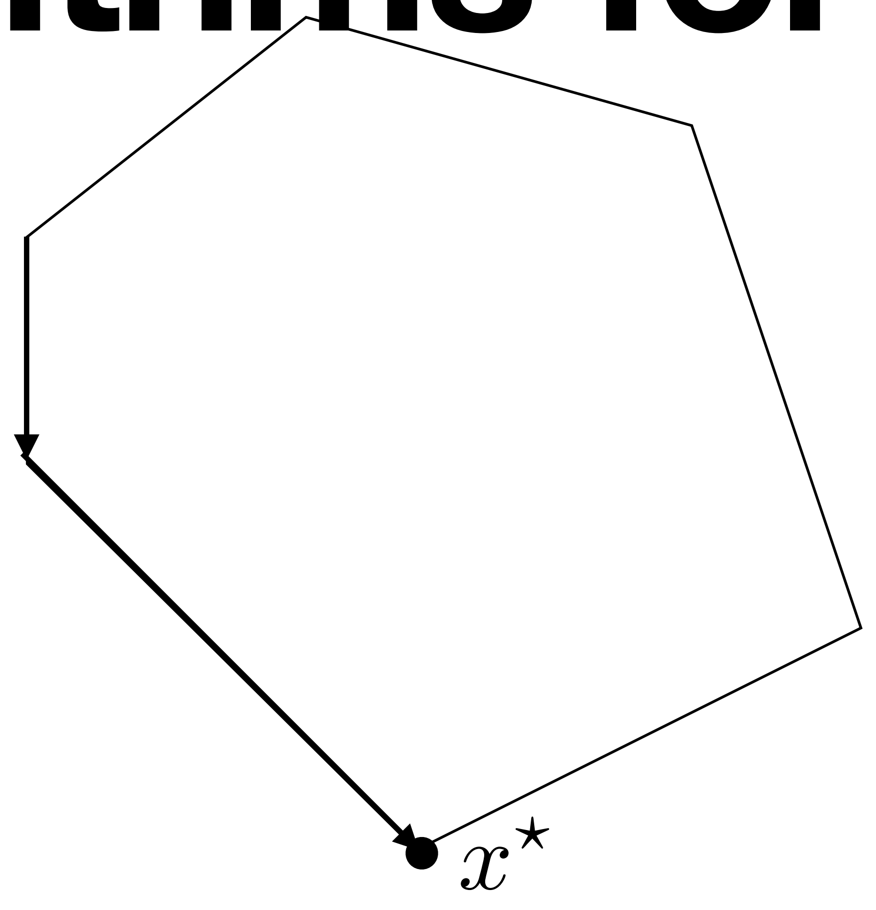


No closed form solution

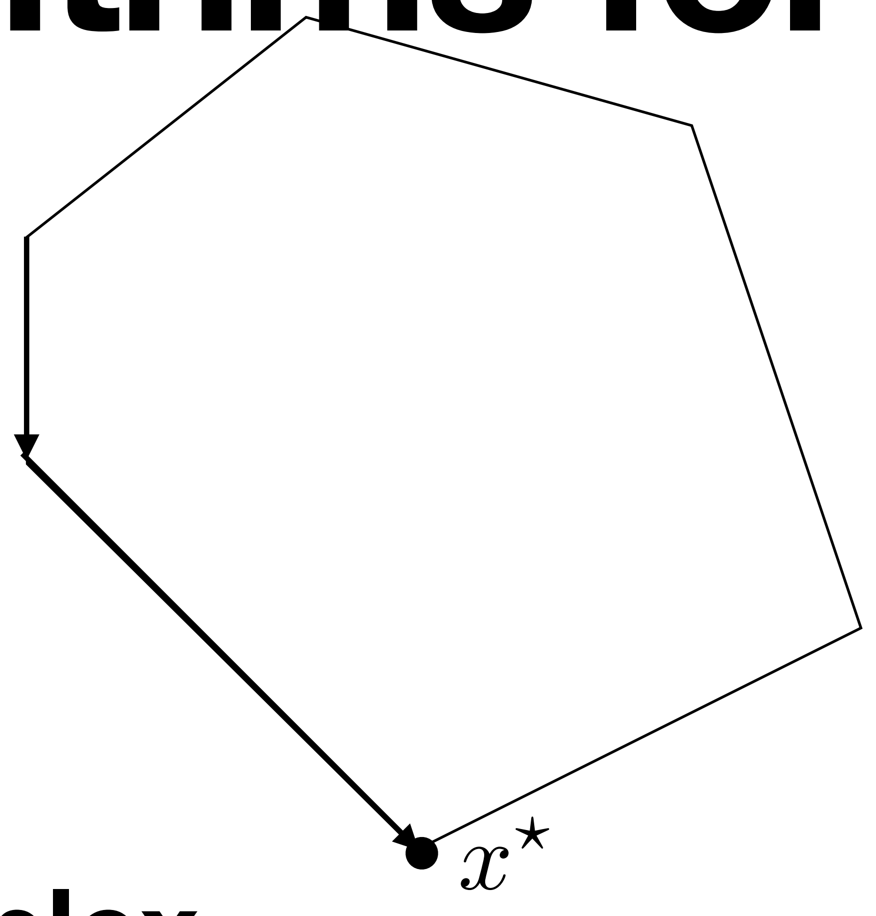
↓

We need an iterative algorithm

Algorithms for linear optimization

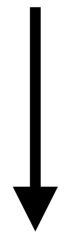


Algorithms for linear optimization



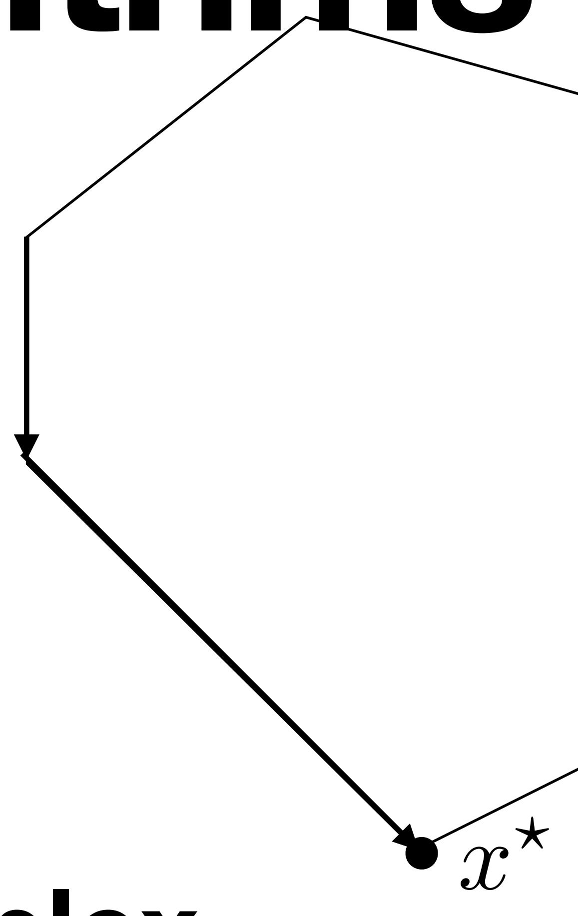
Primal simplex

- Primal feasibility



- Zero duality gap
- Dual feasibility

Algorithms for linear optimization



Primal simplex

- Primal feasibility

↓

- Zero duality gap
- Dual feasibility

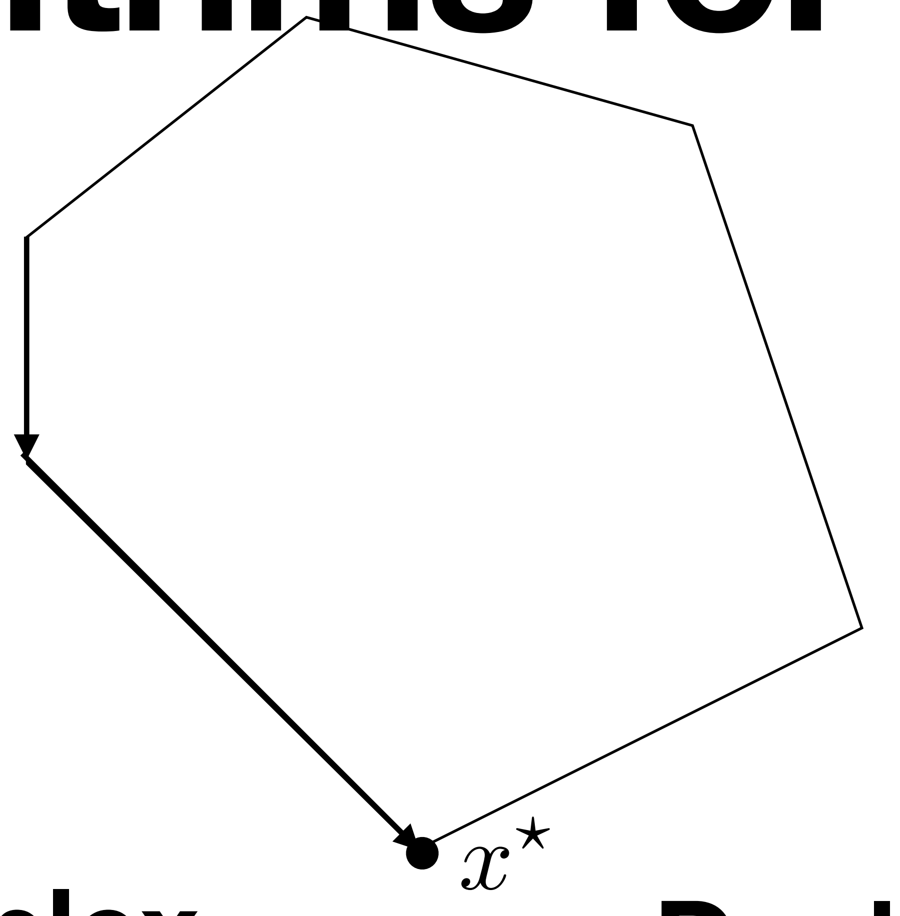
Dual simplex

- Dual feasibility

↓

- Zero duality gap
- Primal feasibility

Algorithms for linear optimization



Primal simplex
- Primal feasibility

↓

- Zero duality gap
- Dual feasibility

Dual simplex
- Dual feasibility

↓

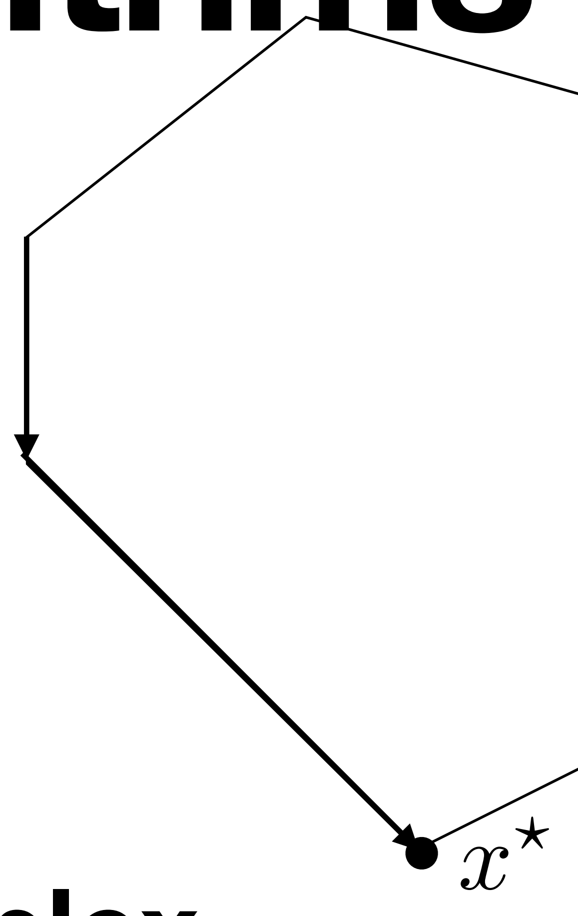
- Zero duality gap
- Primal feasibility

Exponential worst-case complexity

Requires feasible point

Can be warm-started

Algorithms for linear optimization



Primal simplex

- Primal feasibility

↓

- Zero duality gap
- Dual feasibility

Dual simplex

- Dual feasibility

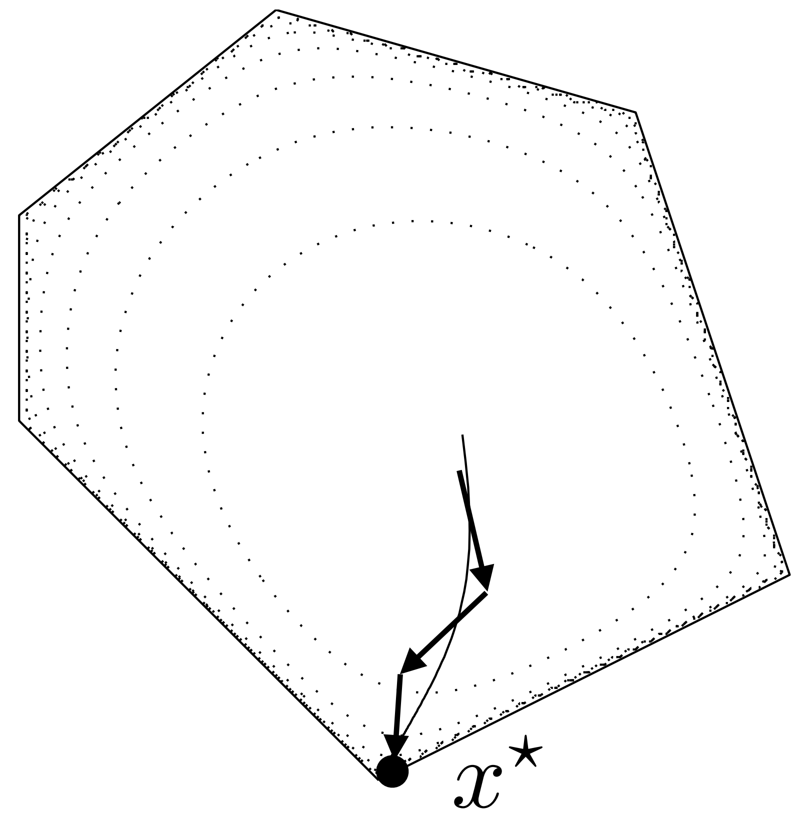
↓

- Zero duality gap
- Primal feasibility

Exponential worst-case complexity

Requires feasible point

Can be warm-started



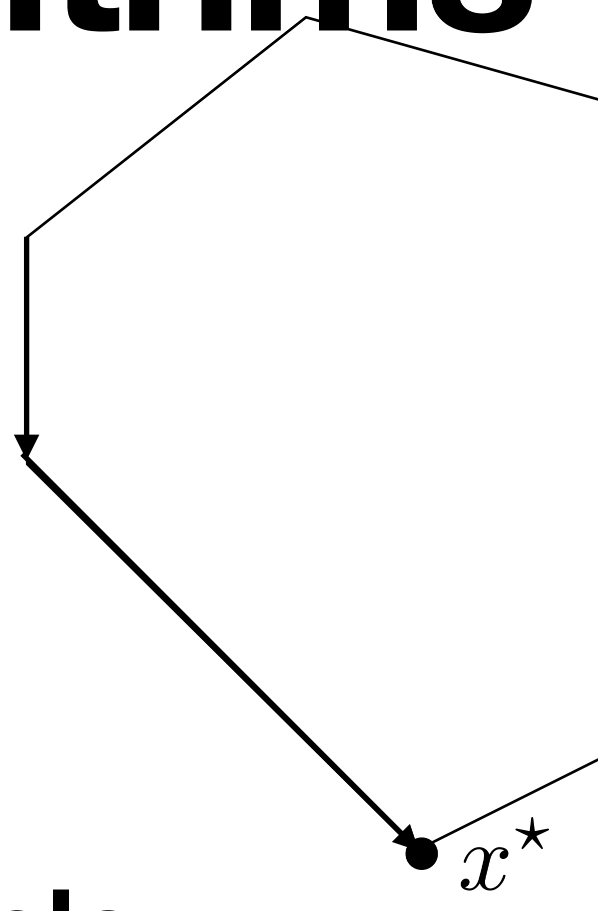
Interior-point methods

- Interior condition

↓

- Primal feasibility
- Dual feasibility
- Zero duality gap

Algorithms for linear optimization



Primal simplex
- Primal feasibility

↓

- Zero duality gap
- Dual feasibility

Dual simplex
- Dual feasibility

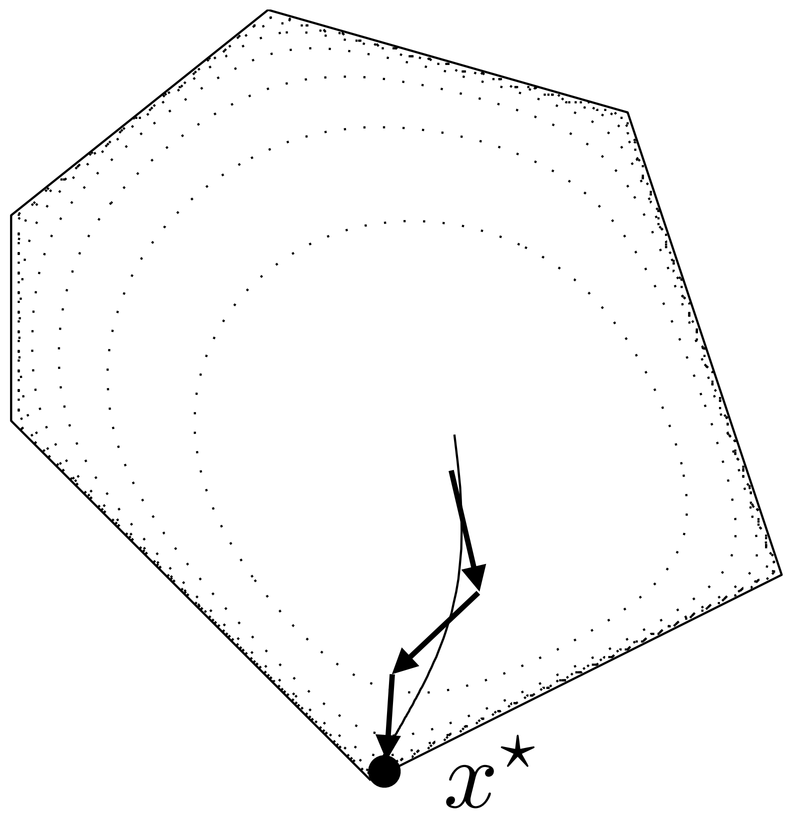
↓

- 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

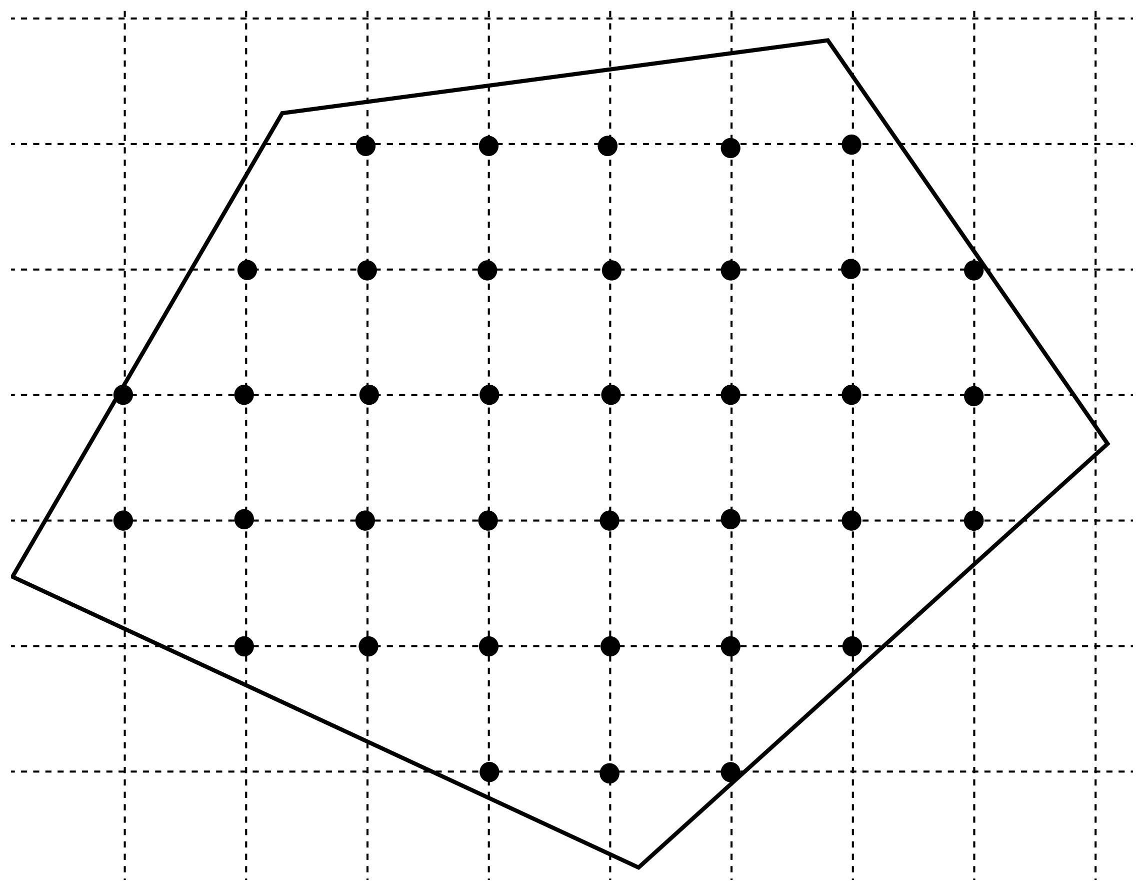
Cannot be warm-started

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

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

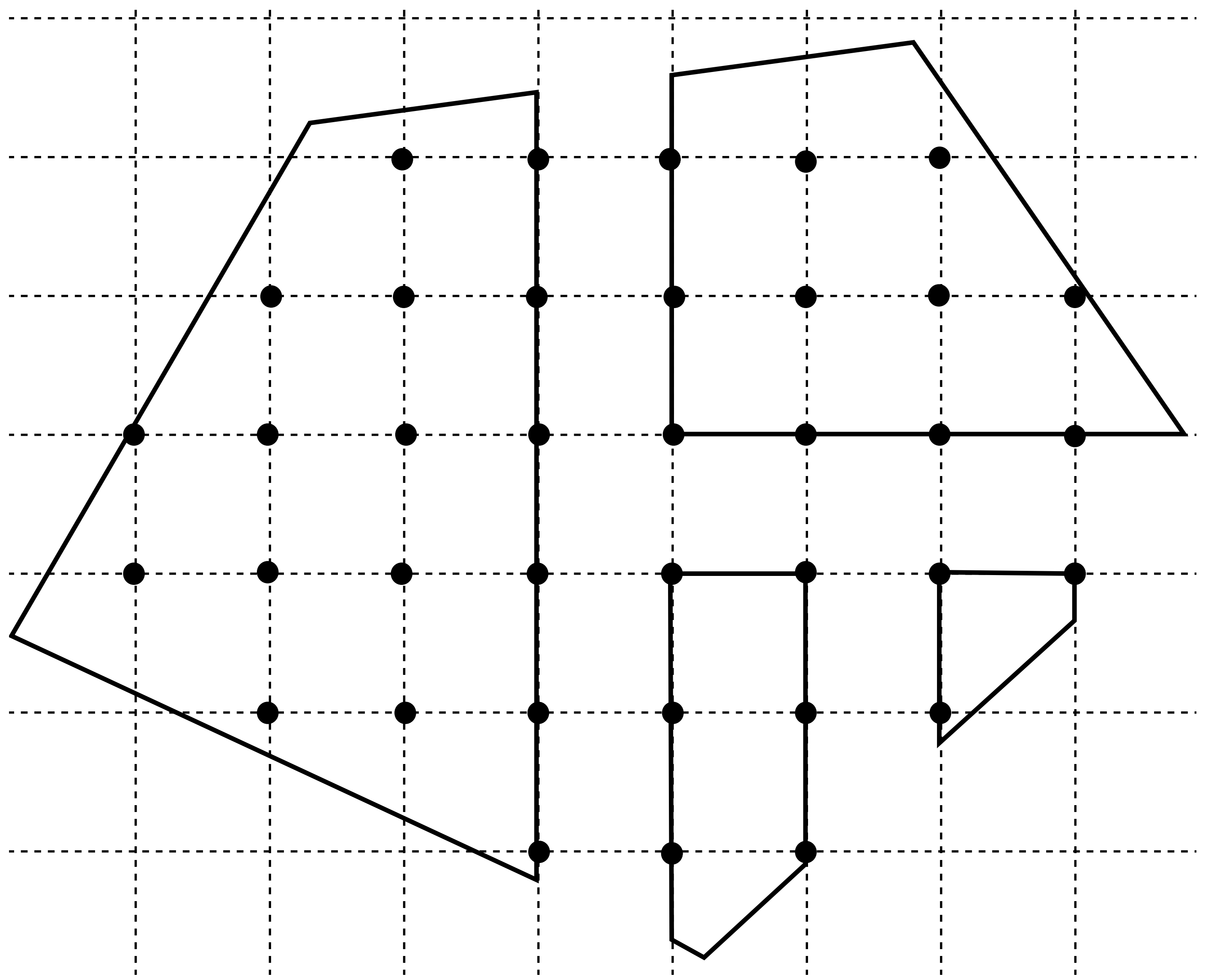


Relaxation does not always give feasible solutions → **Recursively partition the feasible space**

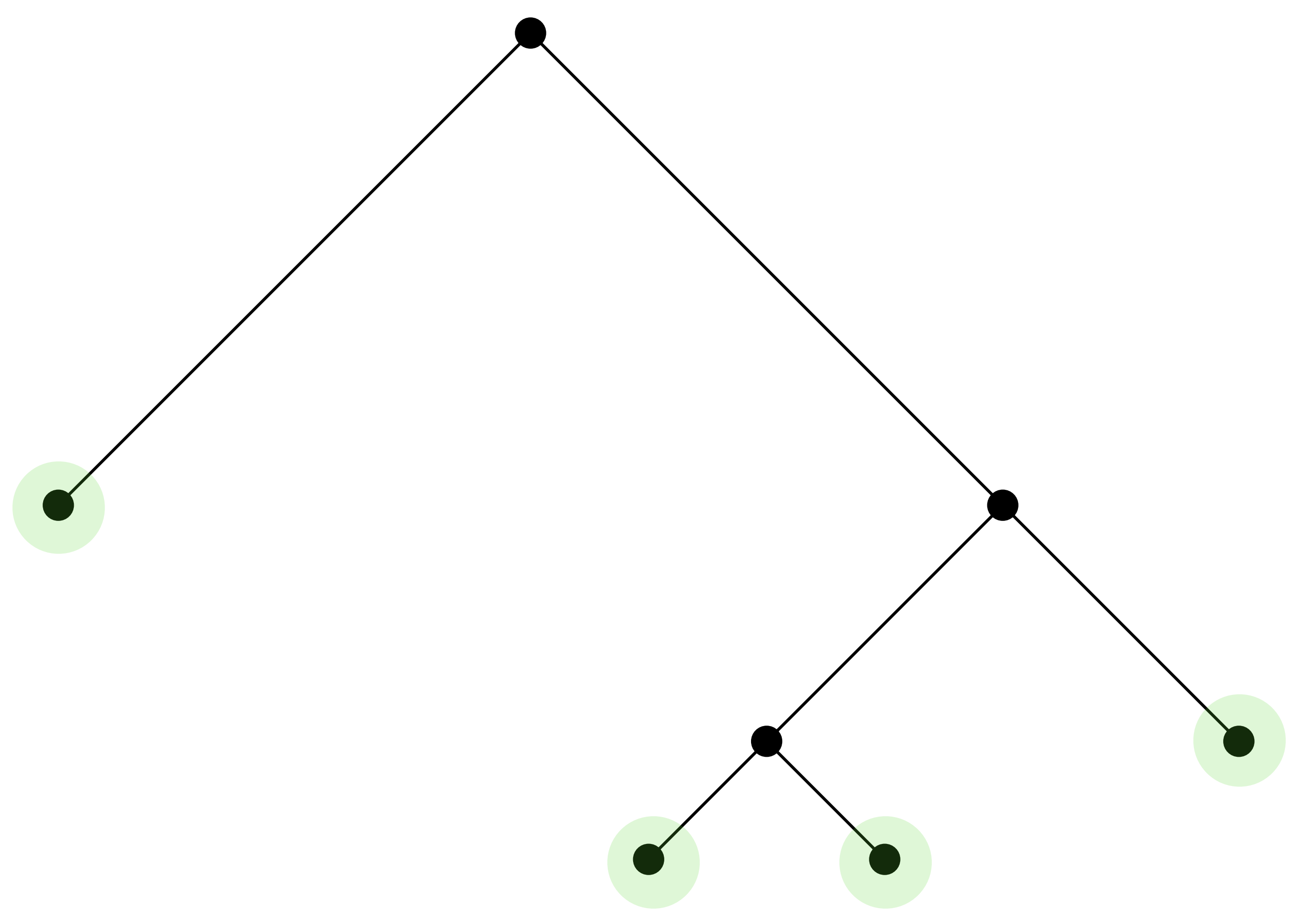
Algorithms for mixed-integer optimization

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)
- Design, Synthesis, and Optimization of Chemical Processes (CBE442)

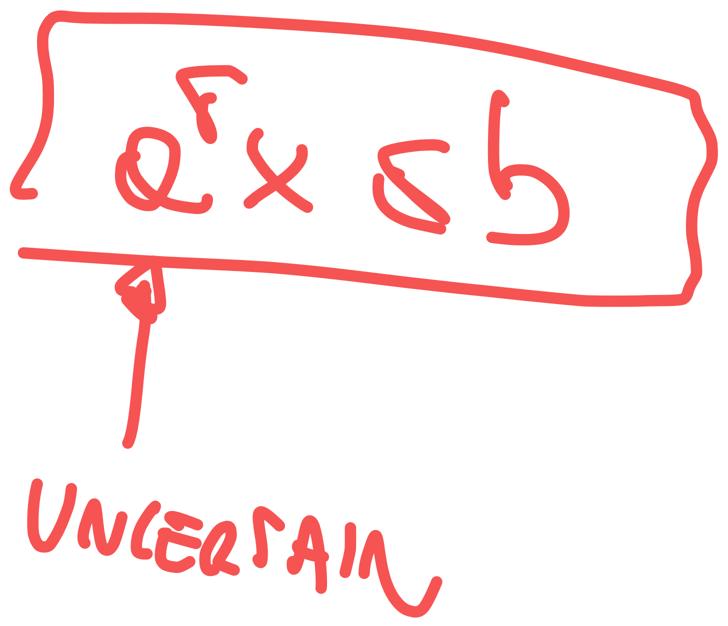
What we did not cover in machine learning?

Machine learning

- Analysis of big data (ORF350)
- Introduction to Machine Learning (COS324)

Decision-making under uncertainty

- Optimal learning (ORF418)
- Stochastic Optimization (ELE544)



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

Optimization problems are just models

"All models are wrong, some are useful."

— George Box

$U - L \leq \varepsilon$ ← 10^{-8}

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

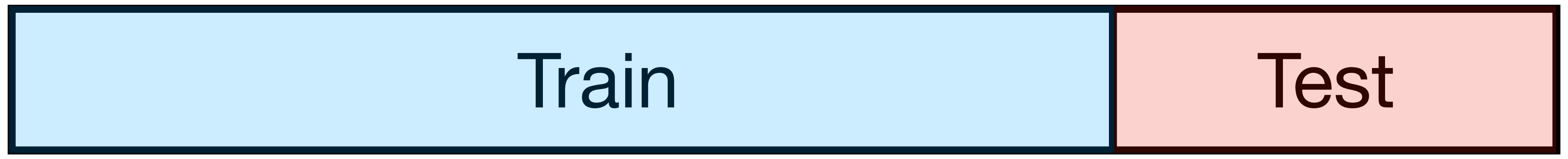
Goal learn model

$$y \approx f(x)$$

from **training data**

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

Data



- 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)

Data fitting

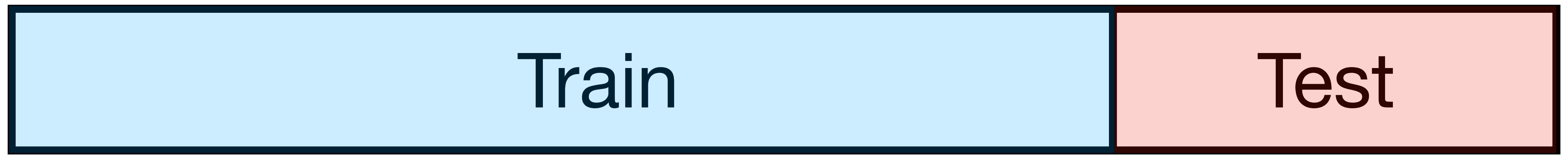
Goal learn model

$$y \approx f(x)$$

from **training data**

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

Data

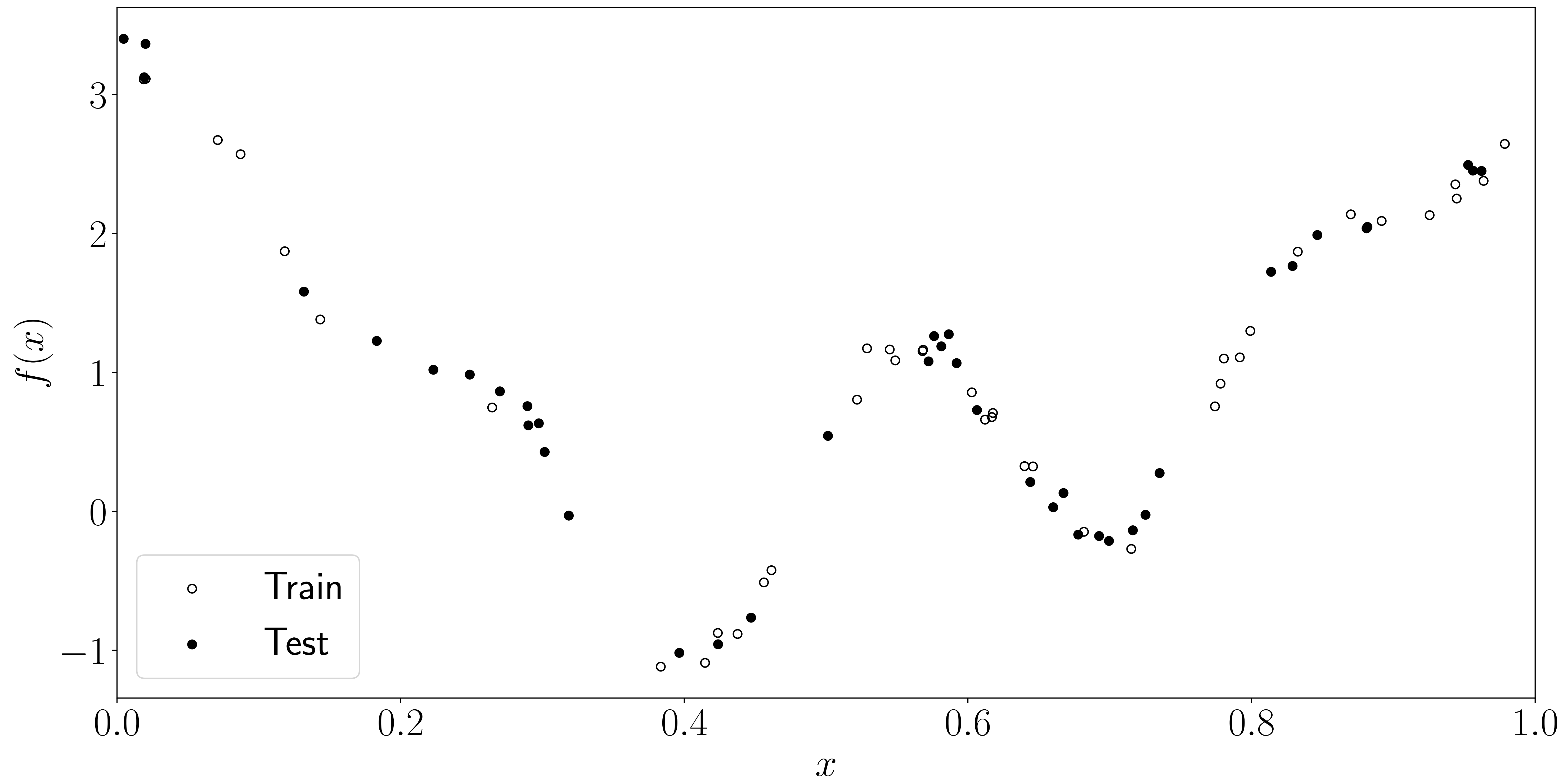


- The goal of model is not to predict outcome for *given data* (Train)
- Instead, it is to predict the outcome on *new, unseen data* (Test)

↓

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

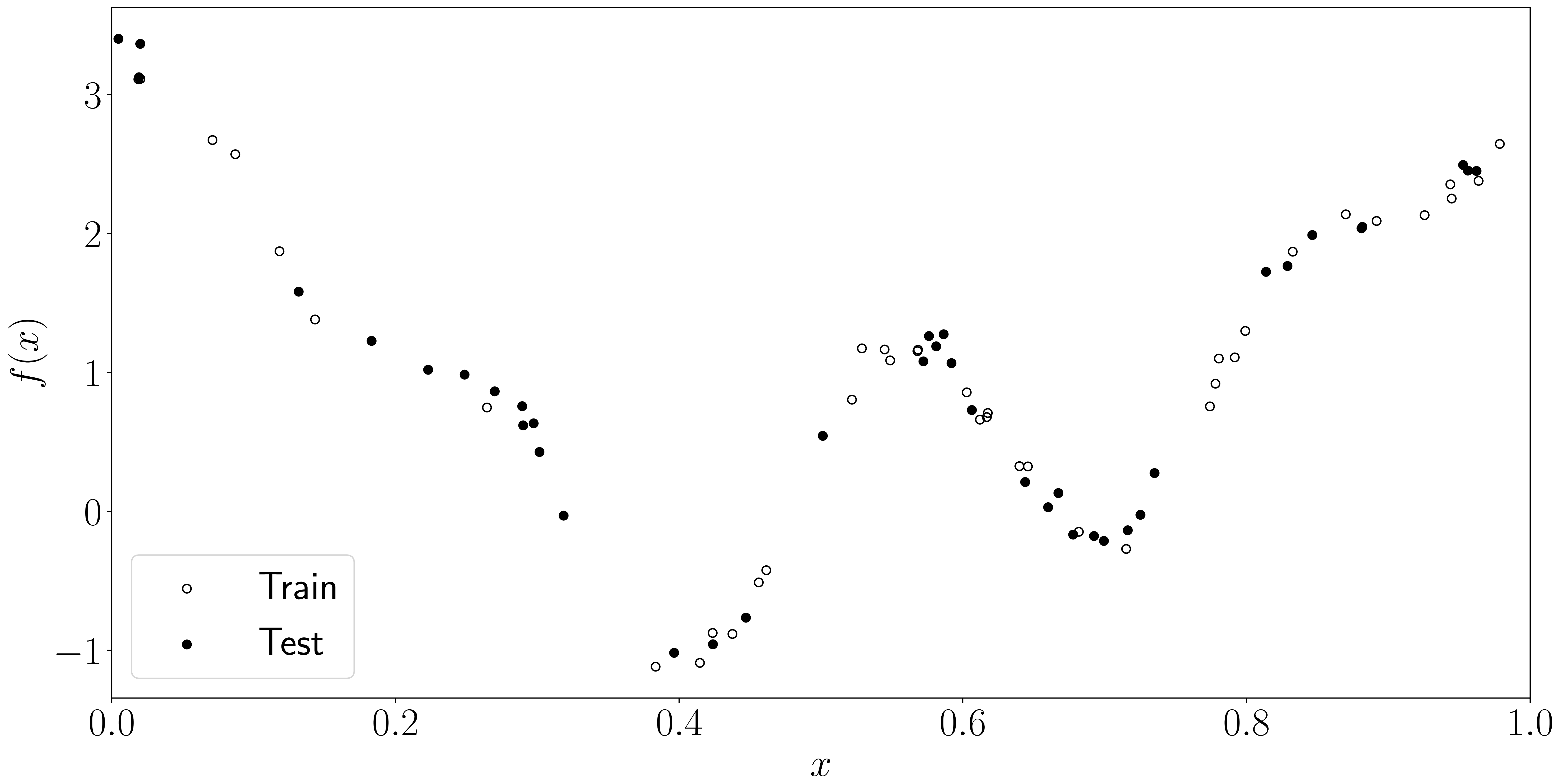
Regularization as proxy for generalization



Regularized fitting LP

minimize $\|Ax - b\|_1 + \gamma \|x\|_1$

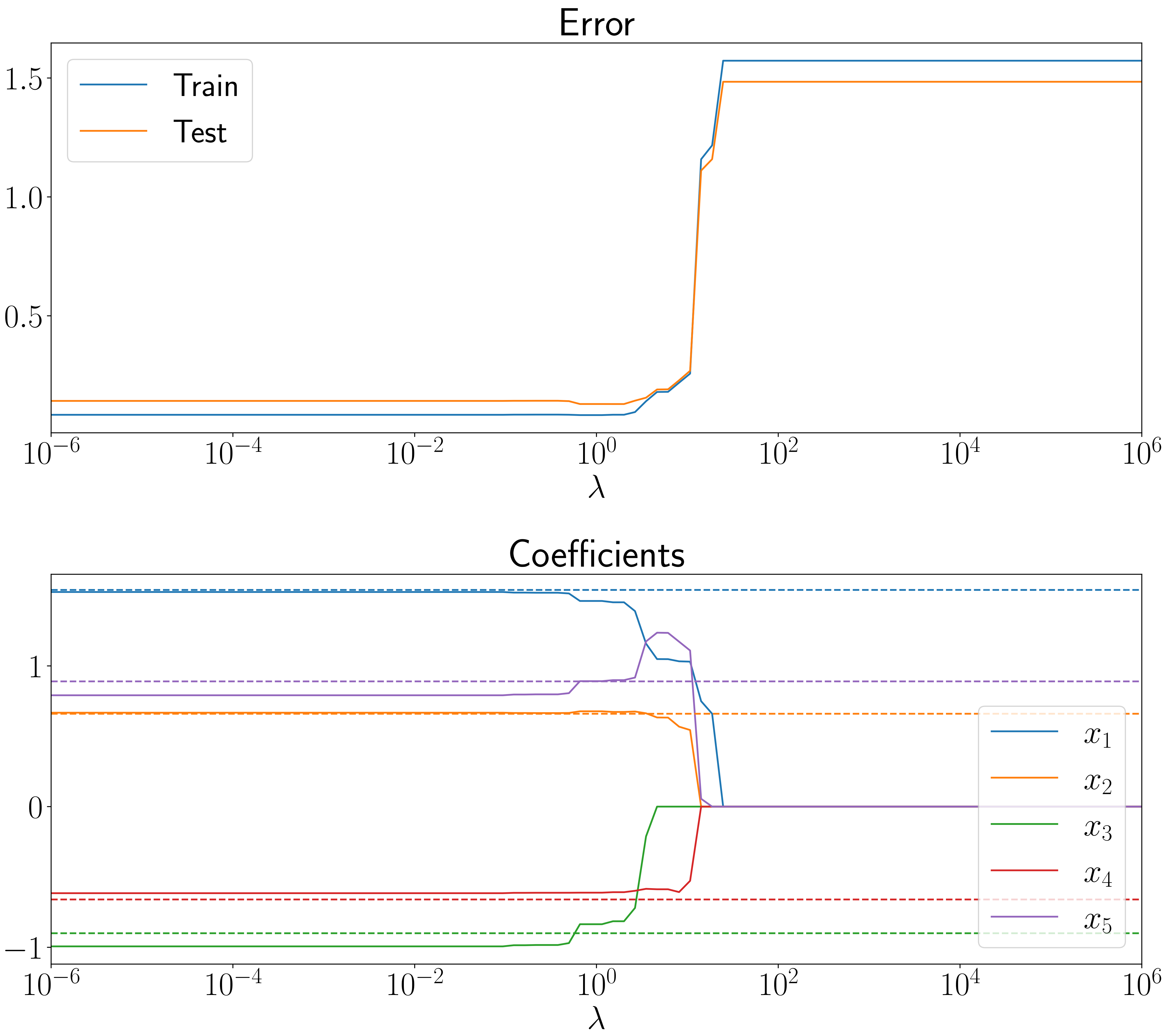
Regularization as proxy for generalization



Regularized fitting LP

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

Train vs test error across regularization



Regularized fitting LP

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

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)

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

Portfolio optimization

Minimize risk-return tradeoff
(on historical data)

Returns

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

Portfolio optimization

Minimize risk-return tradeoff
(on historical data)

Returns → $-\mu^T w$

Risk → $\gamma \|Rw - \mu^T w \mathbf{1}\|_1$

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

Portfolio optimization

Minimize risk-return tradeoff
(on historical data)

Returns **Risk**

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

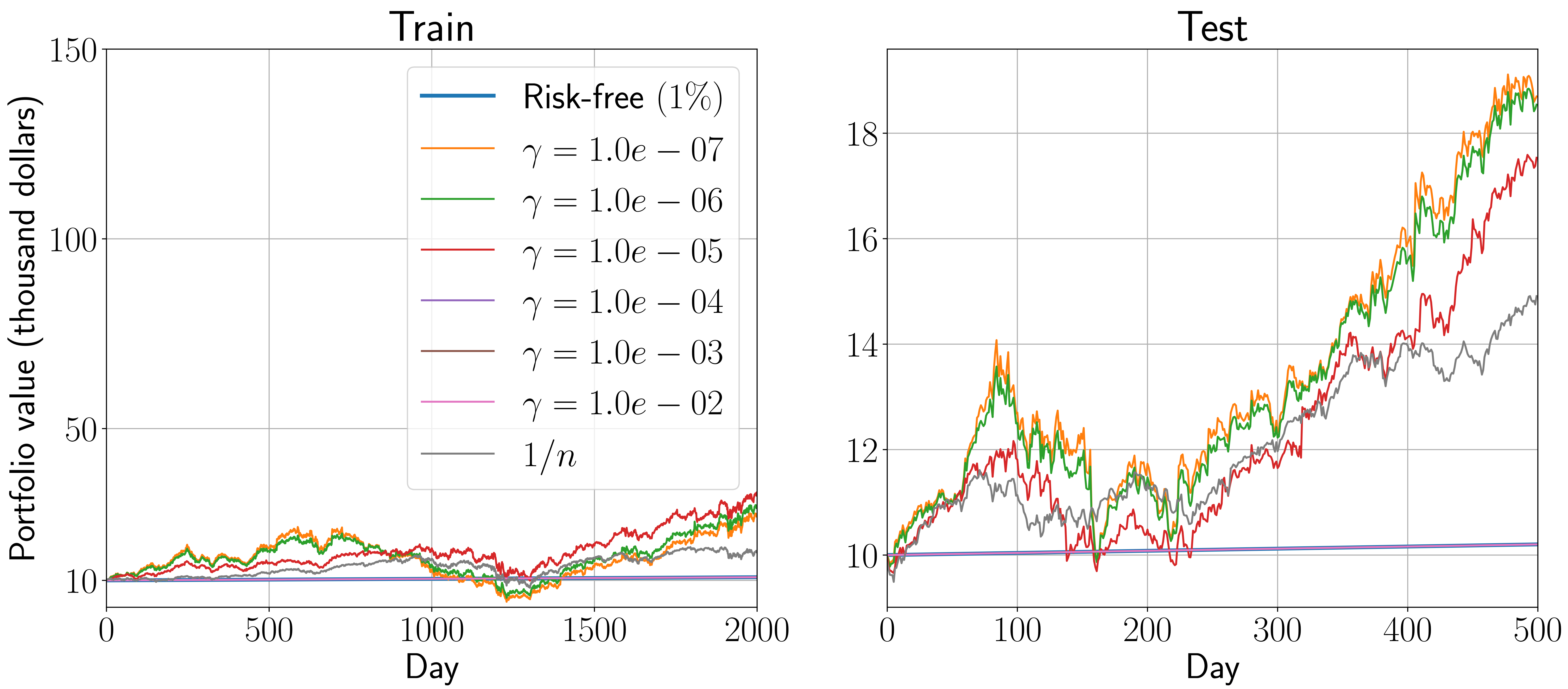
← Proxy

Risk is a proxy to perform well in the future

Past vs future returns on portfolio optimization

Minimize risk-return tradeoff

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



- 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