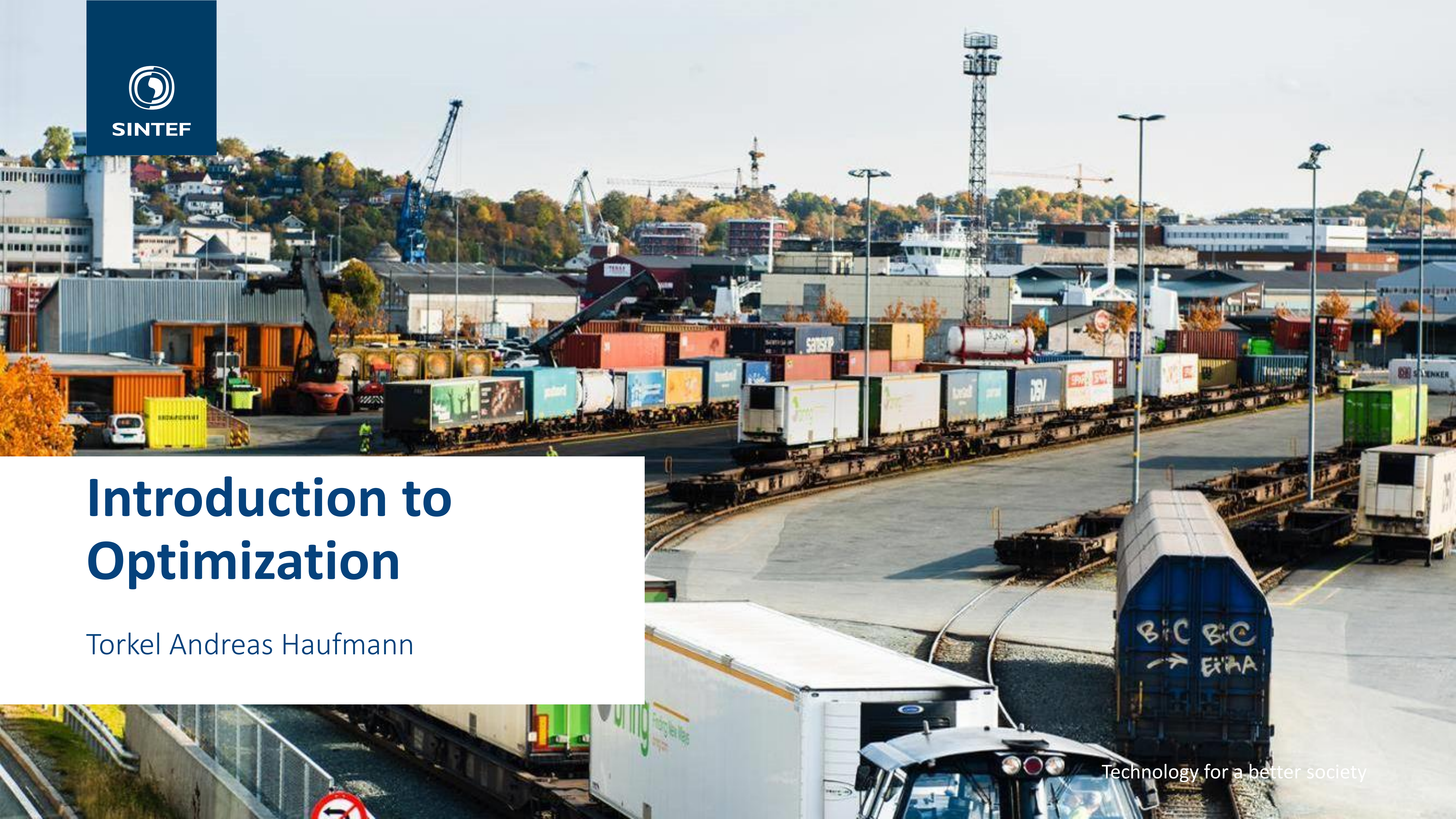
SINTEF



Introduction to Optimization

Torkel Andreas Haufmann

Technology for a better society

Itinerary

- What is optimization?
  - Some history
  - What is it used for?
- On continuous problems
- On discrete problems
  - Examples
  - Solution methods
- Summary

What is optimization?

How to find the *solution* in a *domain* that minimizes an *objective* subject to *constraints.*

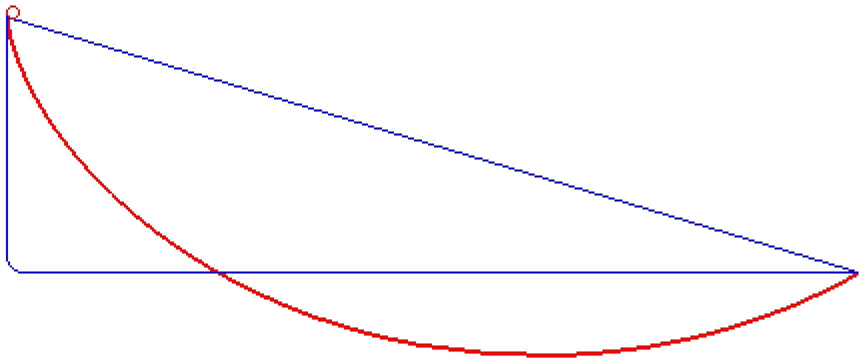
minimize	$f(x)$	←	Objective
subject to	$g(x) \leq 0$	←	Constraints
	$x \in D$	←	Domain

Here

- f is the objective,
- g represents the constraints,
- D is the domain, and
- x is the solution, or set of *decision variables.*

In practical applications, *modeling* is a core activity.

What is optimization?



Credit: Robert Ferréol

- Long mathematical tradition
  - Calculus of variations (e.g. the *brachistochrone curve*) from the 1600s
  - Lagrange, Euler, Newton and others contributing
  - The *transportation problem defined* in late 1700s
  - Gradient methods, linear programming problems, convexity and other theory
- Since World War 2:
  - Tied to Operations Research (OR)
  - Massive advances in theory… and in practice
    - Computer speedups help, but are not the full story

What is optimization?

We need to make good choices:

- When we invest money.
- When we plan transportation routes.
- When we operate a hydroelectric dam.
- When we train a neural network.
- When we plan activities in a project.
- When we design the shape of an object.
- And so on.

The mathematical specifics matter, like *continuous* versus *discrete* domains.

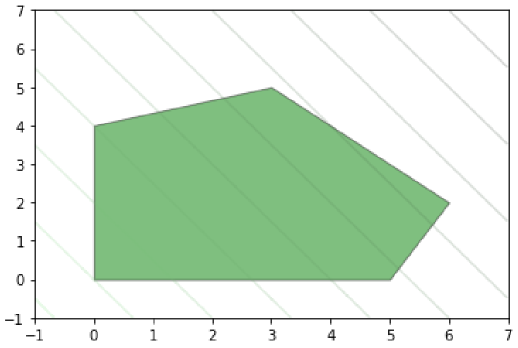
On continuous problems

- *Linear programming* problems:

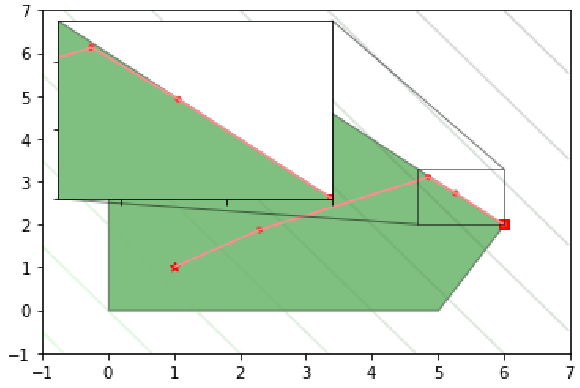
  minimize $\quad c^T x$

  subject to $\quad Ax \leq b$

  $\quad x \in R^n$



- More general problems:
  - *Convex* problems
  - *Interior-point methods*
  - Generally intractable



On discrete problems

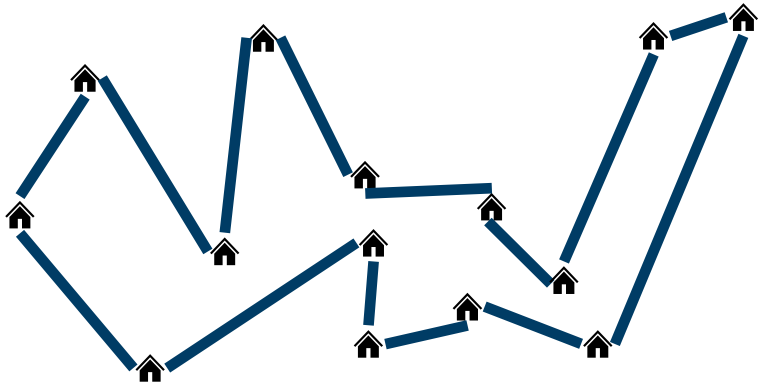
- How to make decisions!
- *Very often* linear, discrete problems are enough:

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \\ & x \in \{0,1\}^n \end{aligned}$$

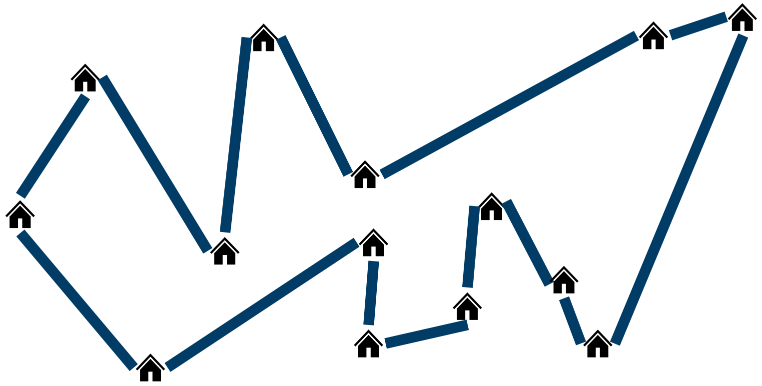
- While linear programming is "easy", this is very hard in general

On discrete problems

- The traveling salesman problem:



… or



On discrete problems

- Example: DFJ formulation for TSP.
  - *n* cities
  - x_{ij} binary variable, 1 means "path goes from *i* to *j*"

$$\text{minimize} \quad \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij}$$

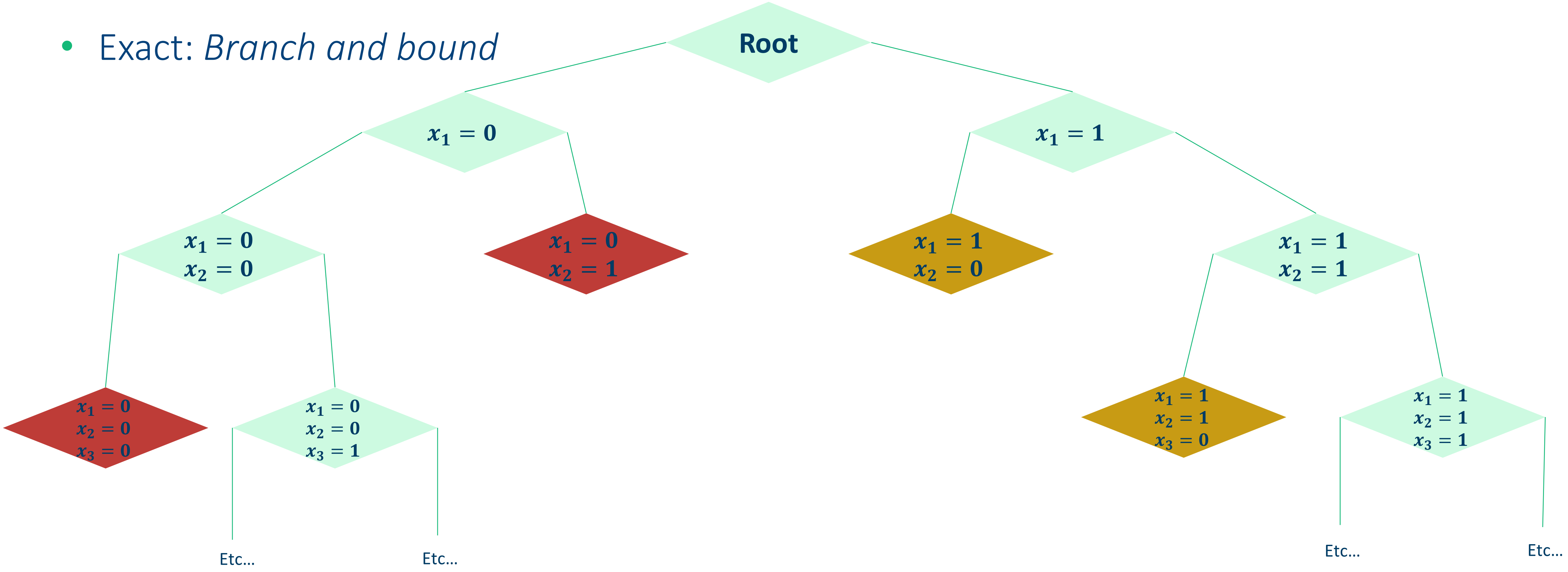
$$\text{subject to} \quad \sum_{i=1, i \neq j}^{n} x_{ij} = 1 \qquad j = 1, \ldots, n$$

$$\sum_{j=1, j \neq i}^{n} x_{ij} = 1 \qquad i = 1, \ldots, n$$

$$\sum_{i \in S} \sum_{j \neq i, j \in S} x_{ij} \leq |S| - 1 \qquad S \subset \{1, \ldots, n\}, |S| \geq 2$$

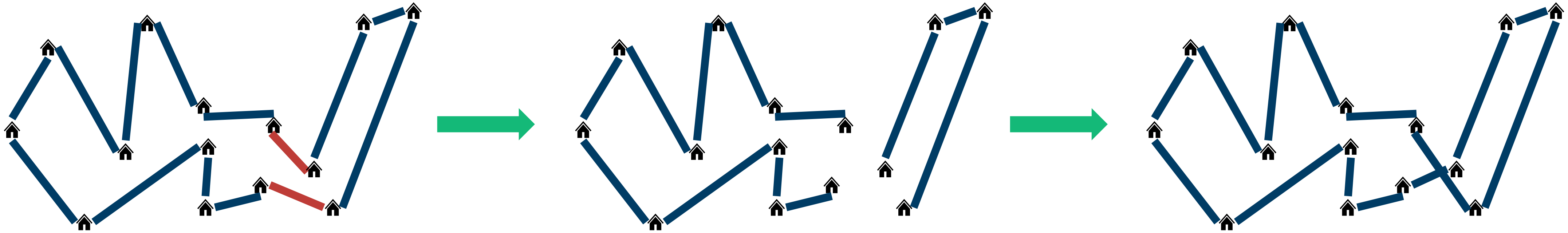
On discrete problems

- Exact: *Branch and bound*

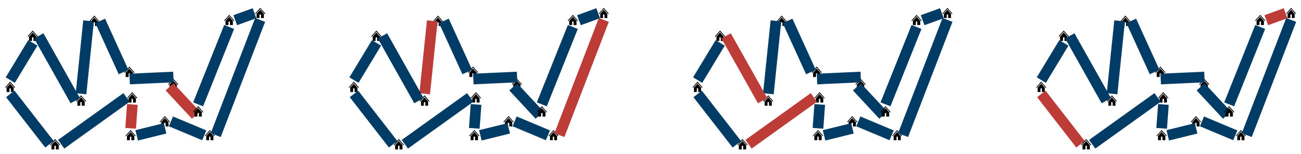


On discrete problems

- Heuristics!
  - Iterative improvement, no formal guarantees.
  - Exact methods may time out anyway.
- Example: 2-opt for TSP



- Many options:

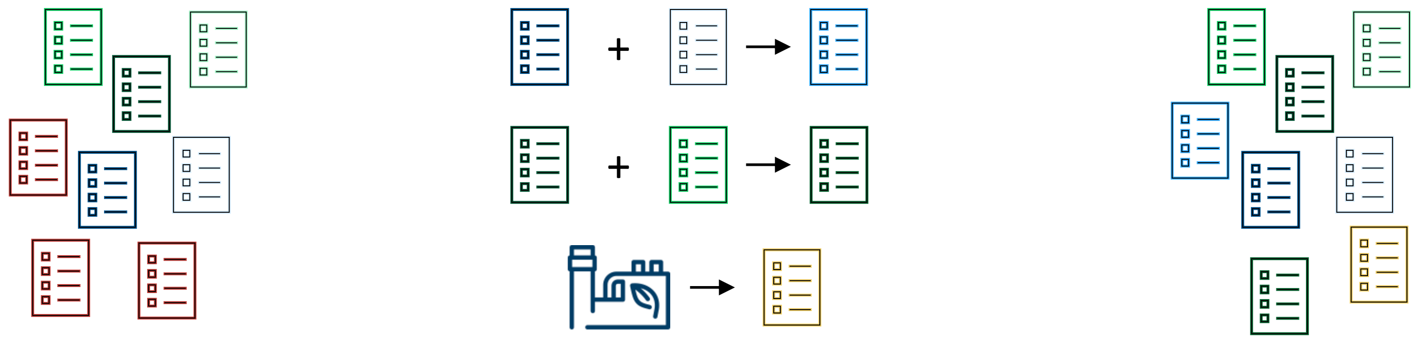


… and so on

SINTEF

On discrete problems

- Metaheuristics
  - Frameworks or strategies for guiding a search
  - Famous example: Genetic algorithms



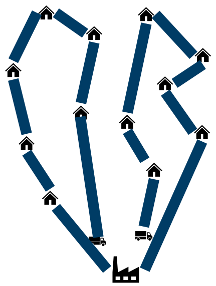
On discrete problems

- Bringing it in
  - Exact methods
  - Heuristics
  - Metaheuristics
  - Ad hoc-solutions
- Far too many solutions!
  - Requires both algorithmic and domain knowledge

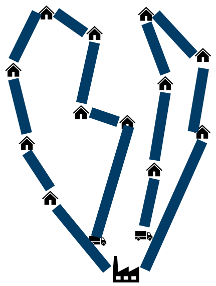
n	$n!$
1	2
4	24
6	720
8	40 320
10	3 628 800
12	479 001 600

Summary

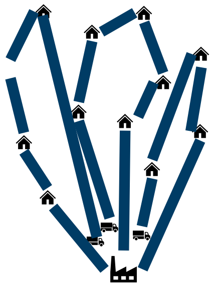
- Under constraints, what is the best decision you can make?
  - Hard limits (Safety rules, physical limitations, etc.)
  - Softer limits (Time, budget, ...)
- Possibility space too large!
- Careful modeling required to solve real-world problems.



... or



... or even



- Used world-wide in industry.

Technology for a better society