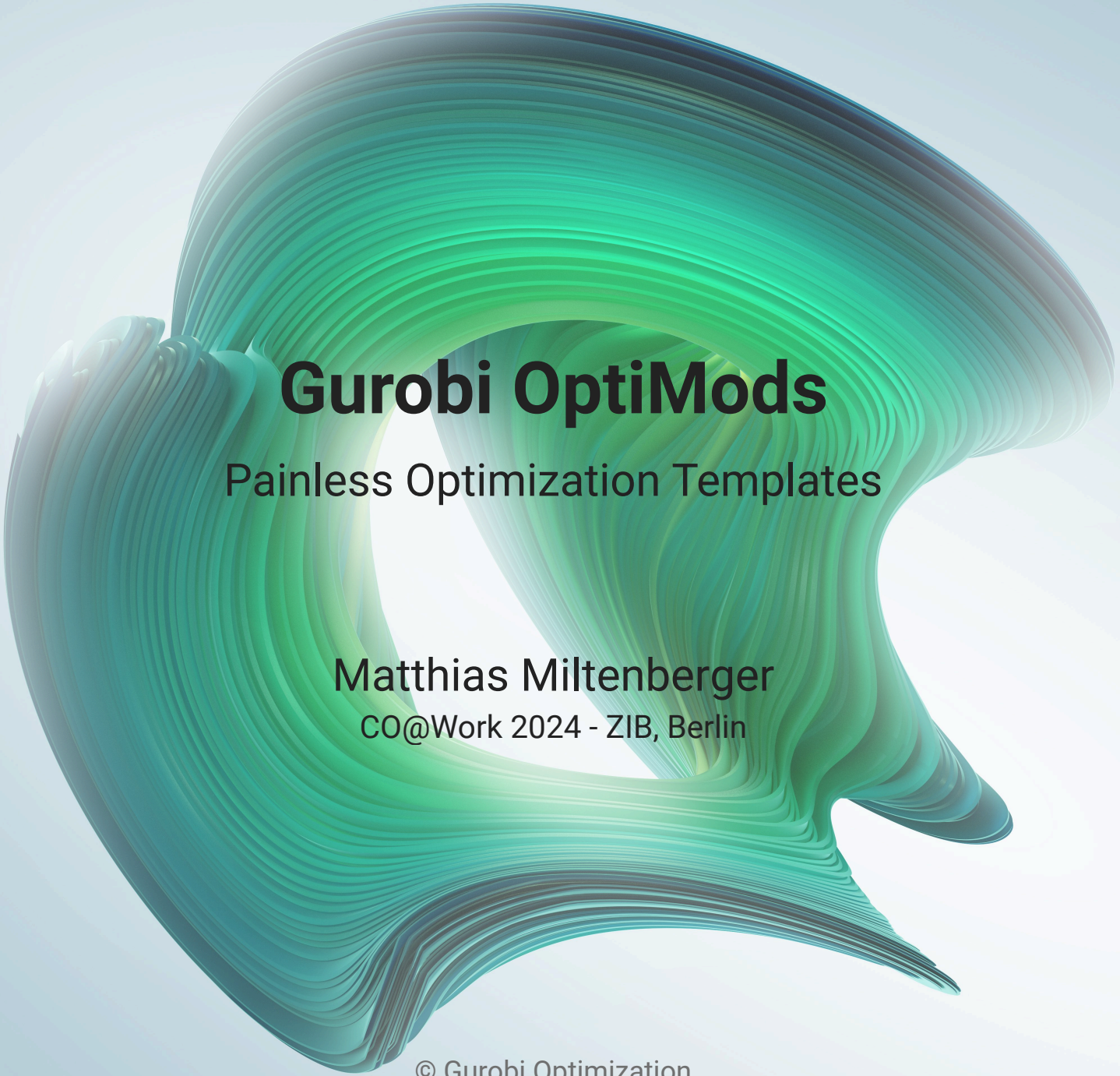


Gurobi OptiMods

Painless Optimization Templates

Matthias Miltenberger

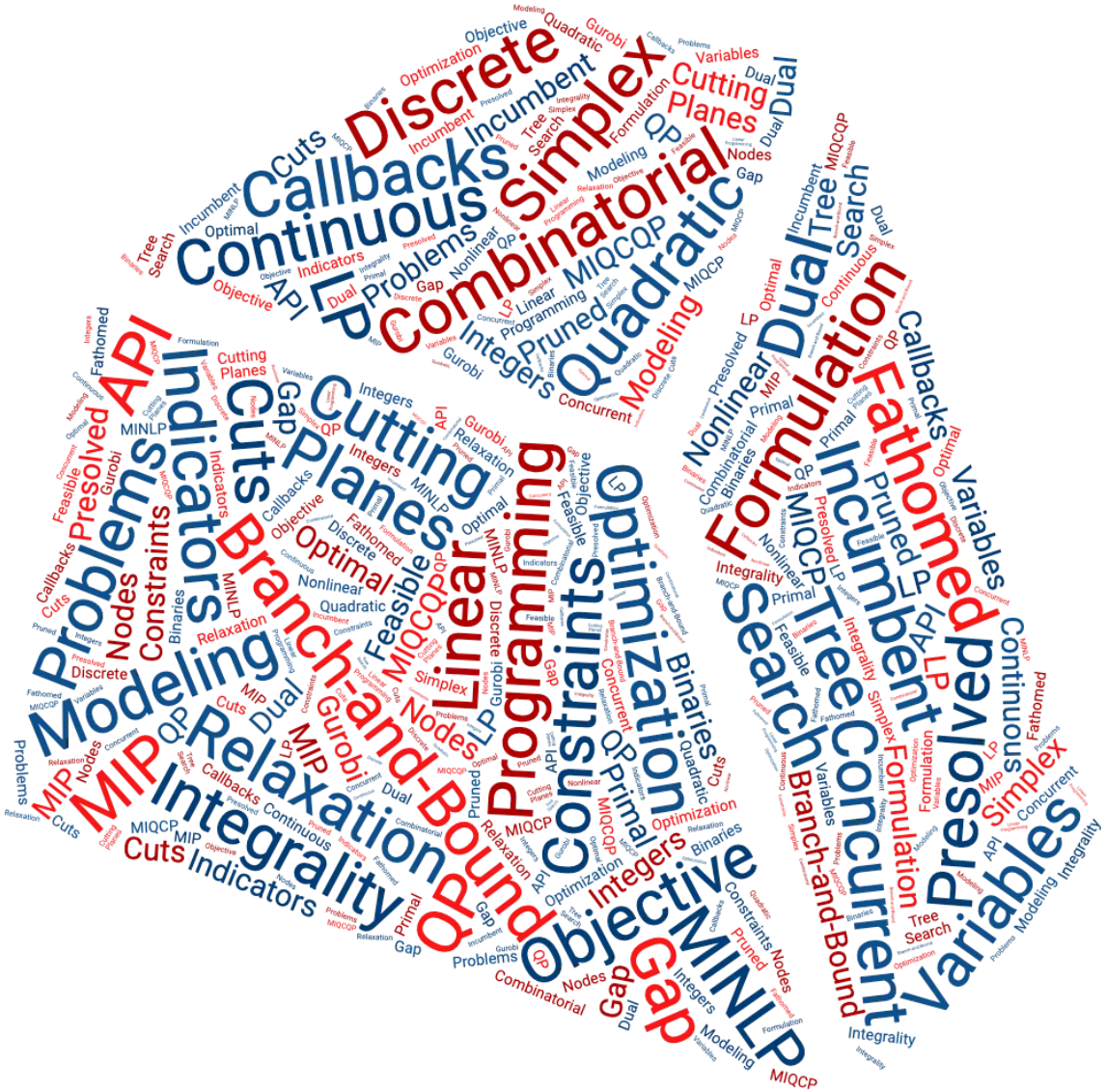
CO@Work 2024 - ZIB, Berlin

gurobi.github.io/slides/

GUROBI OPTIMIZATION

Why OptiMods?

Diving head-first into Optimization can be daunting!



You should be proficient in:

- maths
- modeling
- programming
- data science
- …

enter Gurobi OptiMods!

gurobi.github.io/slides/

GUROBI OPTIMIZATION

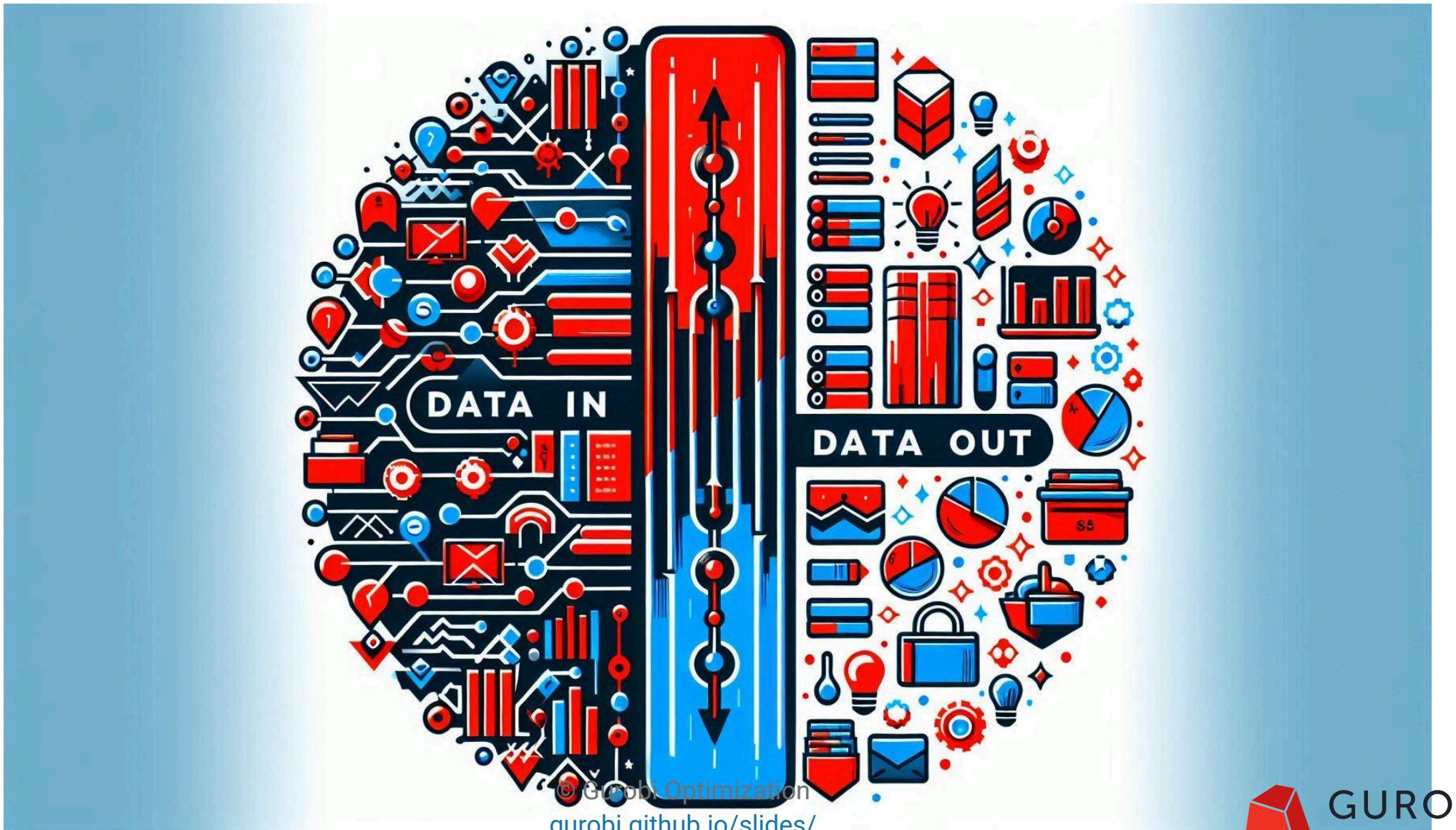
What are OptiMods?

Each OptiMod solves a single, specific problem

gurobi.github.io/slides/

What are OptiMods?

Take input data in natural form, return solutions in natural form



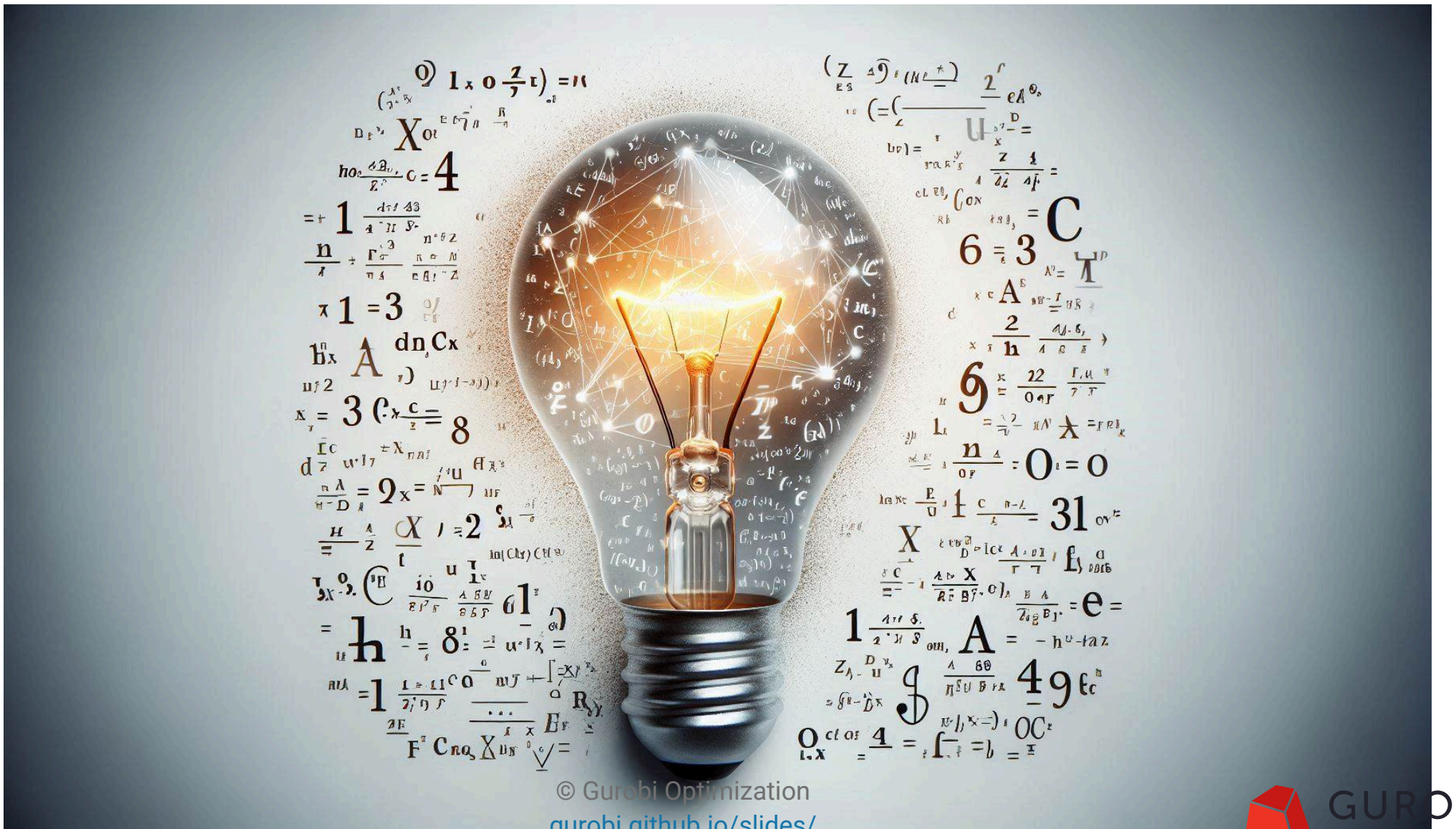
What are OptiMods?

Using gurobipy and common Python packages (pandas, scipy, networkx, …)

gurobi.github.io/slides/

What are OptiMods?

Solves an optimization problem using Gurobi without explicit modeling



What are OptiMods?

Each OptiMod is comprehensively documented

Drawbacks

Are OptiMods the solution to all your problems?

… unfortunately not

Simplicity comes at the price of flexibility!

GUROBI OPTIMIZATION

GUROBI OPTIMIZATION

Search

START

Installation

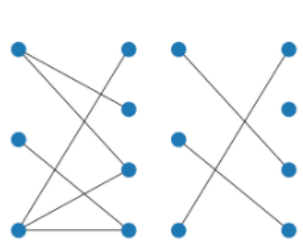
Usage

USER GUIDE

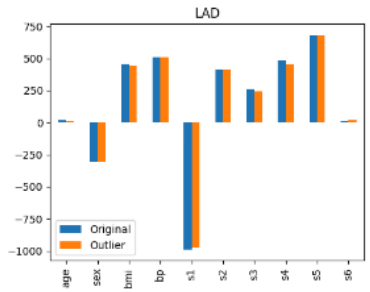
The OptiMods Gallery

- Maximum Bipartite Matching
- Least Absolute Deviation Regression
- Line Optimization in Public Transport
- Metro Map: Computing an Octilinear Graph Representation
- Minimum-Cost Flow
- Maximum Flow / Minimum Cut
- Maximum Weighted Independent Set/Clique
- Optimal Power Flow
- Mean-Variance Portfolio
- Quadratic Unconstrained Binary Optimization (QUBO)
- Maximum Sharpe Ratio

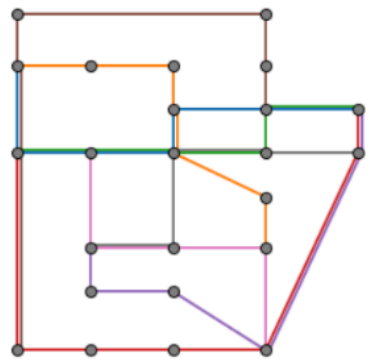
The OptiMods Gallery



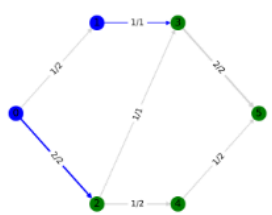
Maximum Bipartite Matching



Least Absolute Deviation Regression

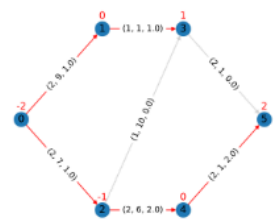


Line Optimization

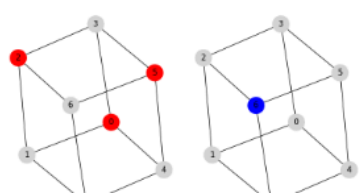


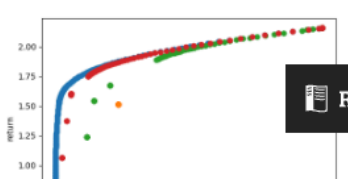
Maximum-Flow/Minimum-Cut

Metro Map



Minimum-Cost Flow





Read the Docs stable

gurobi.github.io/slides/

Quick installation and usage instructions

1. Installation via PyPI:

```
pip install gurobi-optimods
```

2. Documentation on your selected Mod:
   https://gurobi-optimods.readthedocs.io
3. Bring your data into the required format and run the Mod

Documentation is a major part of the project

Demo: Min-Cost-Flow

Mathematical description

For a given graph G with set of vertices V and edges E. Each edge $(i, j) \in E$ has the following pair of attributes:

- cost: $c_{i,j} \in \mathbb{R}$
- capacity: $B_{i,j} \in \mathbb{R}$

Each vertex $i \in V$ has a demand $d_i \in \mathbb{R}$ that can be positive (requesting flow), negative (supplying flow), or zero (a transshipment node).

The problem can be stated as finding the flow with minimal total cost such that:

- demand at each vertex is met exactly and
- flow respects the capacity limit

Demo: Min-Cost-Flow

Input data

```
1 from gurobi_optimods import datasets
2
3 edge_data, node_data = datasets.simple_graph_pandas()
4 edge_data
5 node_data
```

		capacity	cost
source	**target**		
0	1	2	9
	2	2	7
1	3	1	1
2	3	1	10
	4	2	6
3	5	2	1
4	5	2	1

	demand
0	-2
1	0
2	-1
3	1
4	0
5	2

Demo: Min-Cost-Flow

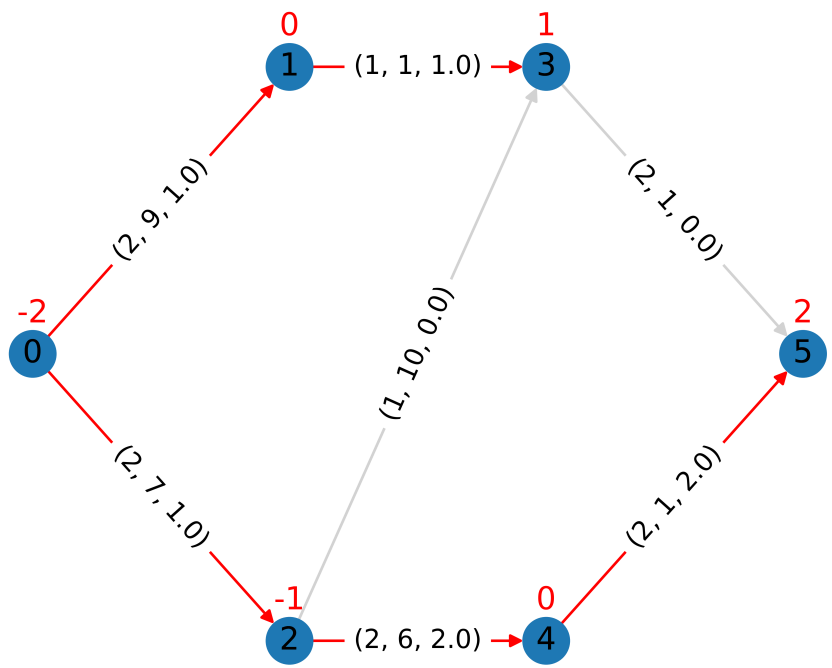
Solution

```
1  from gurobi_optimods import datasets
2  from gurobi_optimods.min_cost_flow import min_cost_flow_pandas
3  edge_data, node_data = datasets.simple_graph_pandas()
4  obj, sol = min_cost_flow_pandas(edge_data, node_data, verbose=False)
5  obj
6  sol
```

```
31.0
```

source	target	flow
0	1	1.0
	2	1.0
1	3	1.0
2	3	0.0
	4	2.0
3	5	0.0
4	5	2.0

Name: flow, dtype: float64



GUROBI OPTIMIZATION

Demo: Line Optimization in Public Transport

Abstract description

The line optimization problem in public transport is to choose a set of lines (routes in the transportation network) with associated frequencies (how often the lines are operated) such that a given transportation demand can be satisfied.

The total operational costs of the computed line plan has to be minimized.

Demo: Line Optimization in Public Transport

Input data

```
from gurobi_optimods import datasets

node_data, edge_data, line_data, linepath_data, demand_data = (
     datasets.load_siouxfalls_network_data()
)
frequencies = [1,3]
```

```
node_data.head(4)
```

	number	posx	posy
0	1	50000.0	510000.0
1	2	320000.0	510000.0
2	3	50000.0	440000.0
3	4	130000.0	440000.0

Demo: Line Optimization in Public Transport

Input data

```
from gurobi_optimods import datasets

node_data, edge_data, line_data, linepath_data, demand_data = (
     datasets.load_siouxfalls_network_data()
)
frequencies = [1,3]
```

```
edge_data.head(4)
```

	source	target	length	time
0	1	2	0.010	360
1	2	1	0.010	360
2	1	3	0.006	240
3	3	1	0.006	240

Demo: Line Optimization in Public Transport

Input data

```
from gurobi_optimods import datasets

node_data, edge_data, line_data, linepath_data, demand_data = (
     datasets.load_siouxfalls_network_data()
)
frequencies = [1,3]
```

```
line_data.head(4)
```

	linename	capacity	fix_cost	operating_cost
0	new7_B	600	15	3
1	new15_B	600	15	2
2	new23_B	600	15	6
3	new31_B	600	15	6

Demo: Line Optimization in Public Transport

Input data

```
from gurobi_optimods import datasets

node_data, edge_data, line_data, linepath_data, demand_data = (
     datasets.load_siouxfalls_network_data()
)
frequencies = [1,3]
```

```
linepath_data.head(4)
```

	linename	edge_source	edge_target
0	new7_B	1	2
1	new7_B	2	6
2	new7_B	6	8
3	new7_B	8	6

Demo: Line Optimization in Public Transport

Input data

```
from gurobi_optimods import datasets

node_data, edge_data, line_data, linepath_data, demand_data = (
     datasets.load_siouxfalls_network_data()
)
frequencies = [1,3]
```

```
demand_data.head(4)
```

	source	target	demand
0	1	2	5
1	1	3	5
2	1	4	25
3	1	5	10

Demo: Line Optimization in Public Transport

Solution

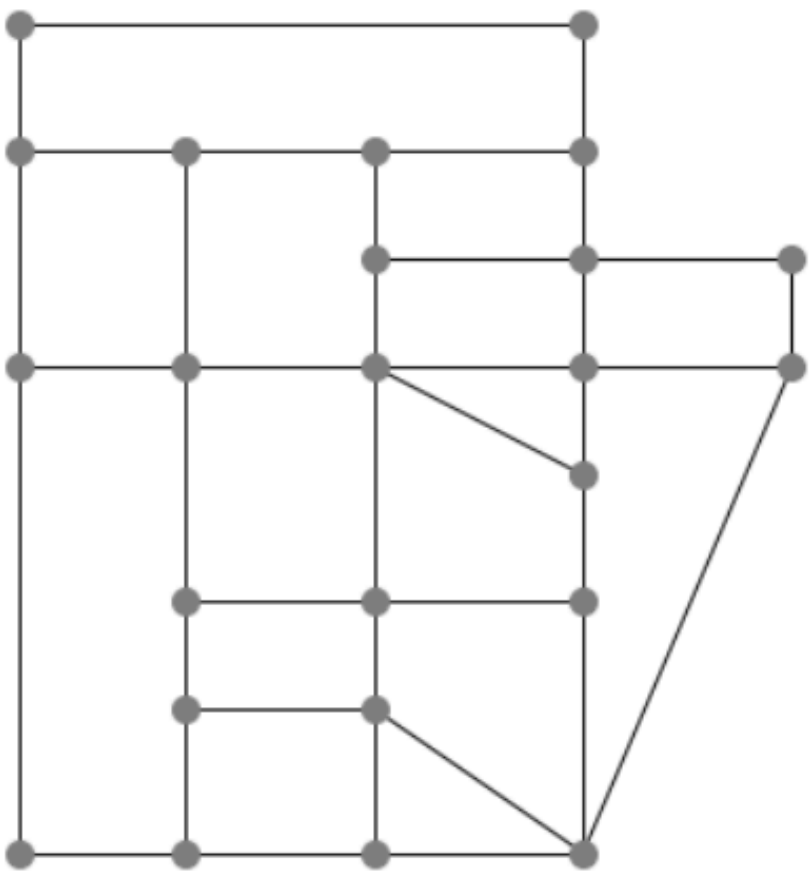
```
from gurobi_optimods import datasets
from gurobi_optimods.line_optimization import line_optimization
node_data, edge_data, line_data, linepath_data, demand_data = (
     datasets.load_siouxfalls_network_data()
)
frequencies = [1,3]
obj_cost, final_lines = line_optimization(
    node_data,
    edge_data,
    line_data,
    linepath_data,
    demand_data,
    frequencies,
    verbose=False,
)
obj_cost
```

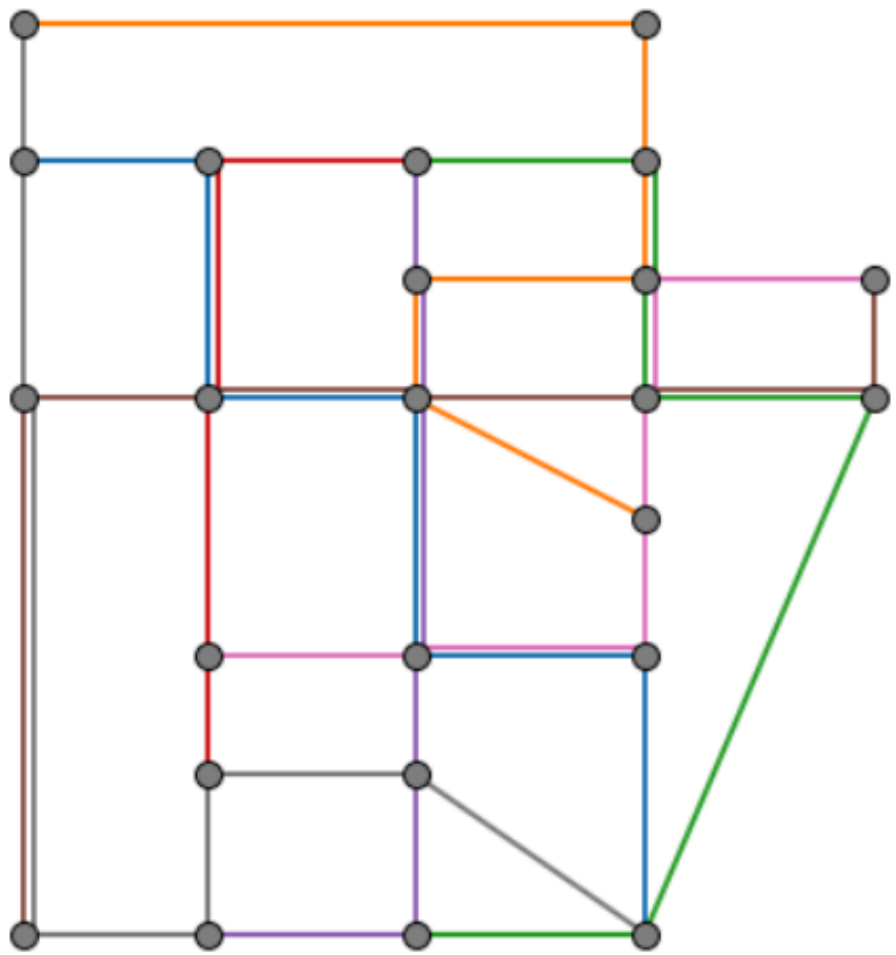
```
211.0
```

Demo: Line Optimization in Public Transport

Visualization

```
1  from gurobi_optimods.line_optimization import plot_lineplan
2  plot_lineplan(node_data, edge_data, linepath_data, final_lines)
```





Our latest Addition: Metro Map OptiMod

Computing an Octilinear Graph Representation

gurobi.github.io/slides/

GUROBI OPTIMIZATION

Summary

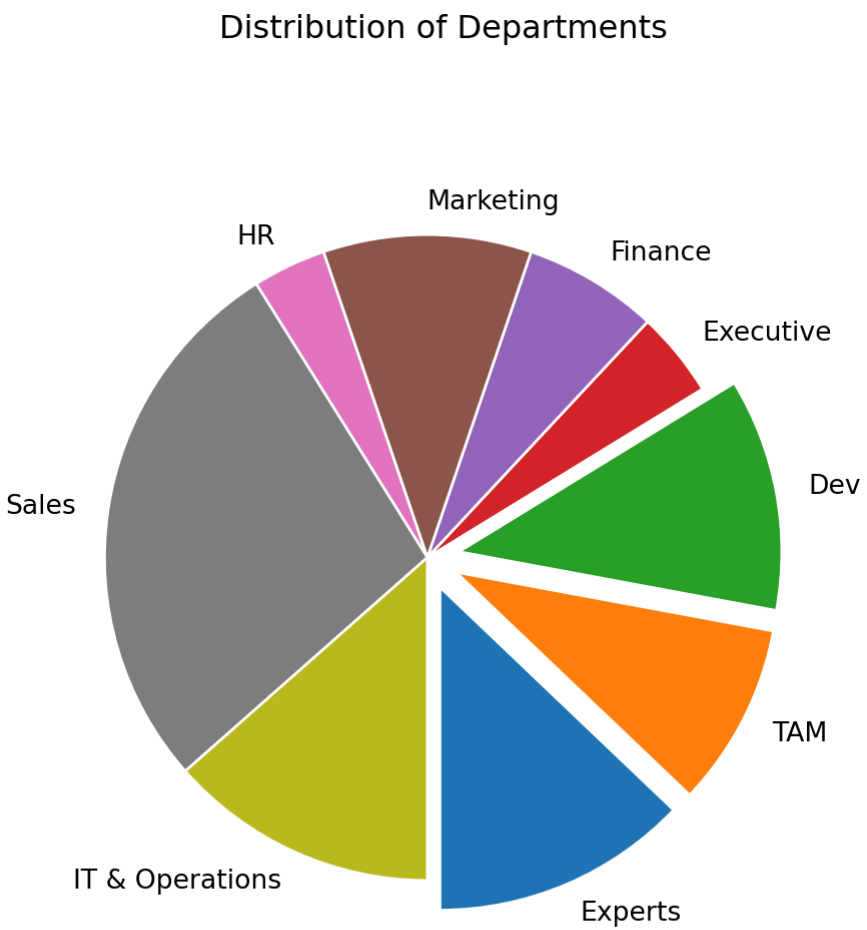
https://github.com/Gurobi/gurobi-optimods

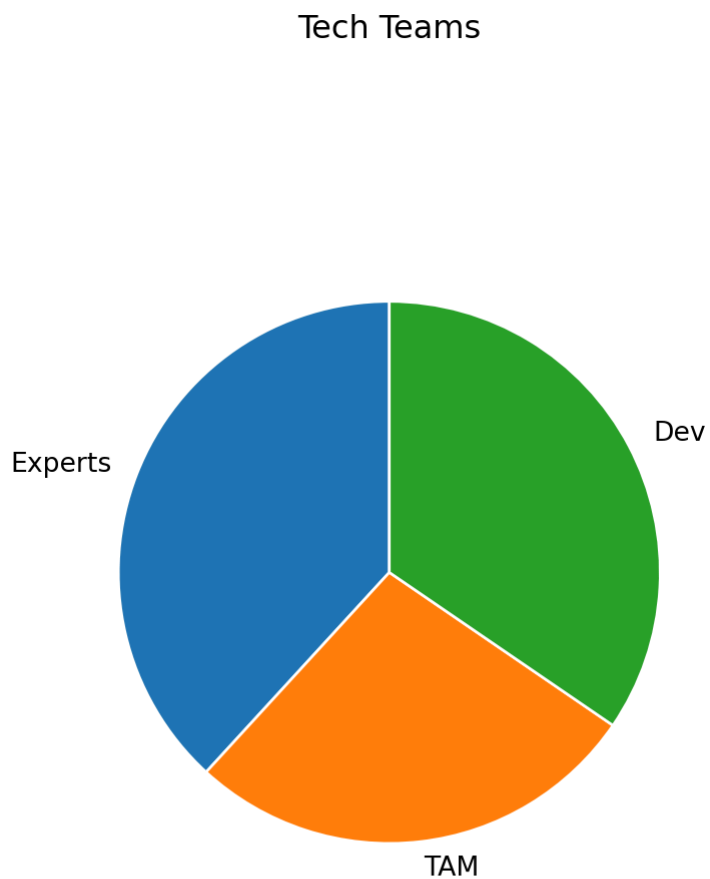
- OptiMods provide data-driven APIs to solve common optimization problems
- Great way to make optimization more accessible
- Many optimization topics are still waiting to be turned into a new OptiMod
- Gurobi's other open-source Python packages:
  - gurobipy-pandas: directly connect DataFrames with gurobipy
  - gurobi-logtools: parse logfiles into DataFrames, e.g., for plotting
  - gurobi-machinelearning: use trained models within optimization problems
  - gurobi-modelanalyzer: compute numerical features of LPs

Get involved and share your feedback and ideas!

Gurobi as a Company

170 people all around the world working remotely

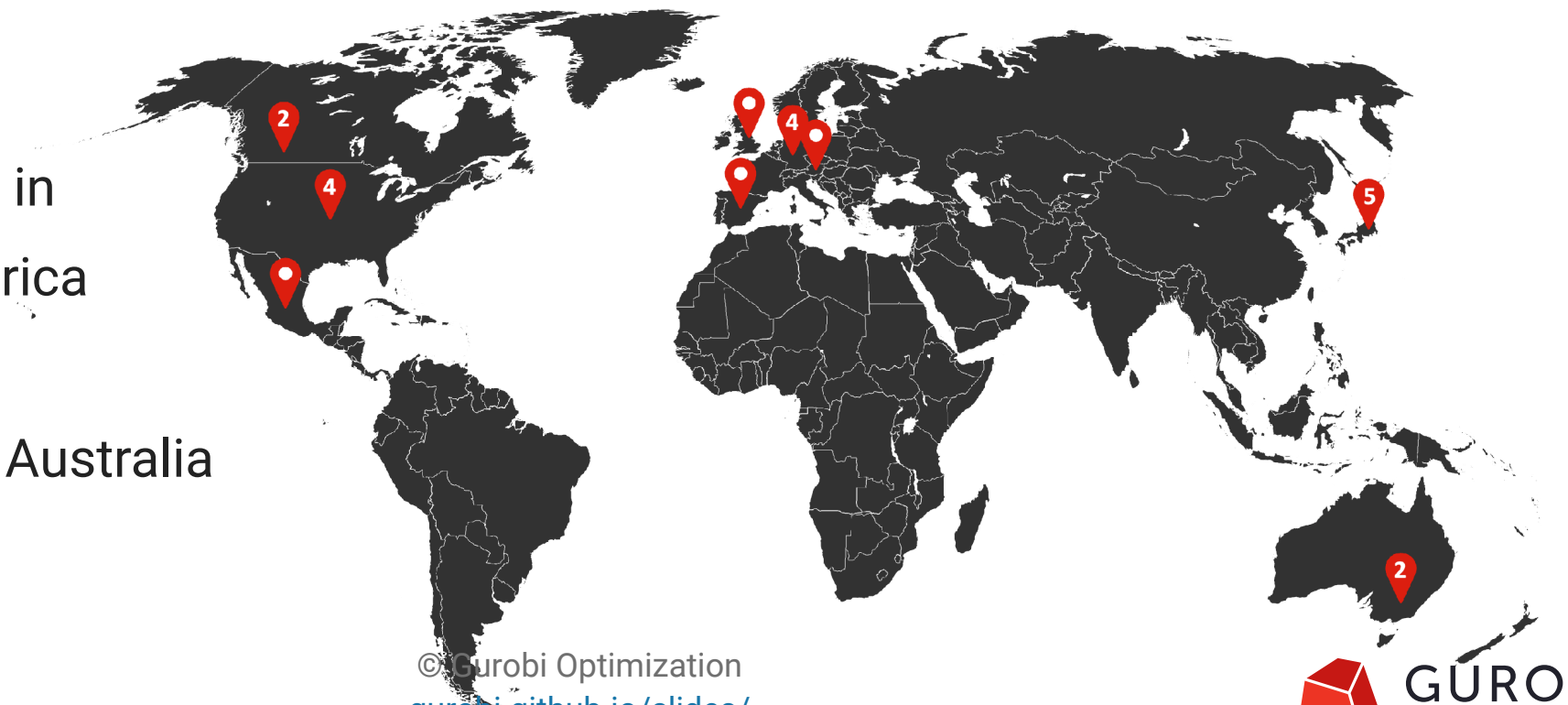




The Gurobi Experts Team

Supporting all our users around the clock and driving innovation

- Global team
- 7 people each in
  - North America
  - Europe
  - Japan and Australia

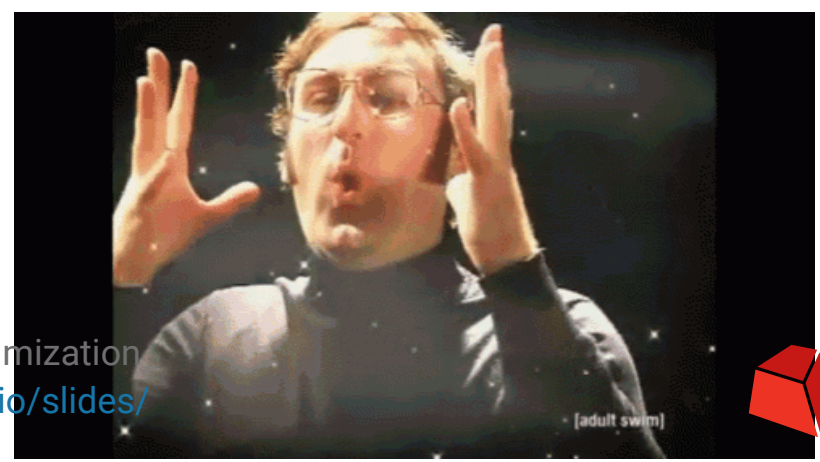


gurobi.github.io/slides/

GUROBI OPTIMIZATION

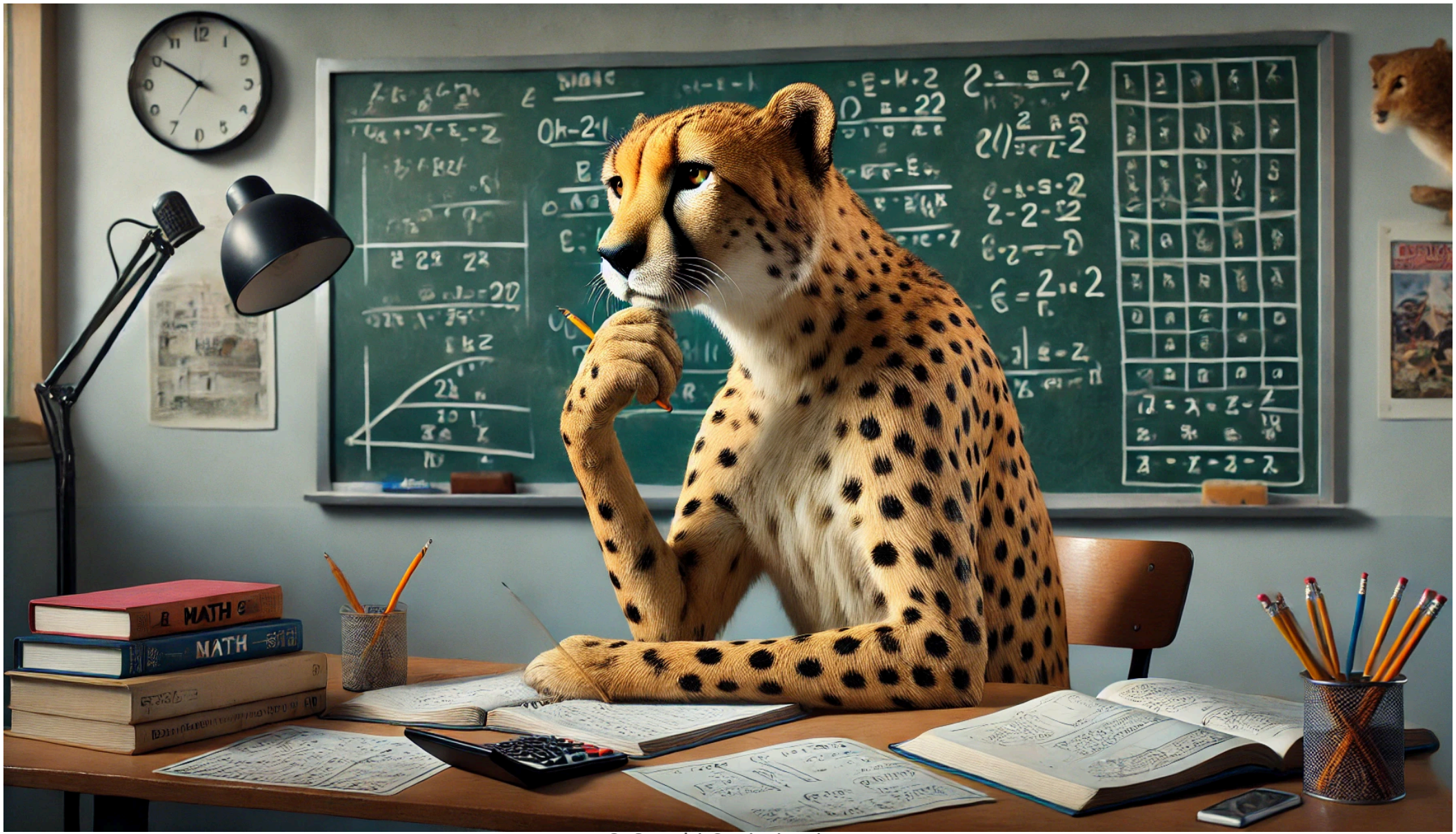
Recent Success Story

- Customer opens a ticket and asks for a parameter tuning:
  - starting point: 5% gap in 30 minutes
- Goals:
  - 🥈 1% gap in 10 minutes would be great
  - 🥉 1% gap in 30 minutes would good
  - ⏱️ 5% gap in 5 minutes would also be helpful
- After automatic parameter tuning for 35 hours on 12 machines:
  - 🥇 2% gap in 30 minutes
- **After Simran's reformulation idea:**
  - 🚀 **0.03% gap in 60 seconds!**



Raw Performance is good…

… but don’t forget about the Mathematics!



GUROBI OPTIMIZATION

Happy Customers

- Many customers have a strong OR/math background
- We engage with the subject matter experts of the specific company
- Less tech-savvy customers are always super thankful for our help

We send out a quick survey after each ticket to ask how they liked our Support…

For the last 2 years the rating has never been below 98% positive replies

How are you going to start your Journey?

gurobi.github.io/slides/

GUROBI OPTIMIZATION