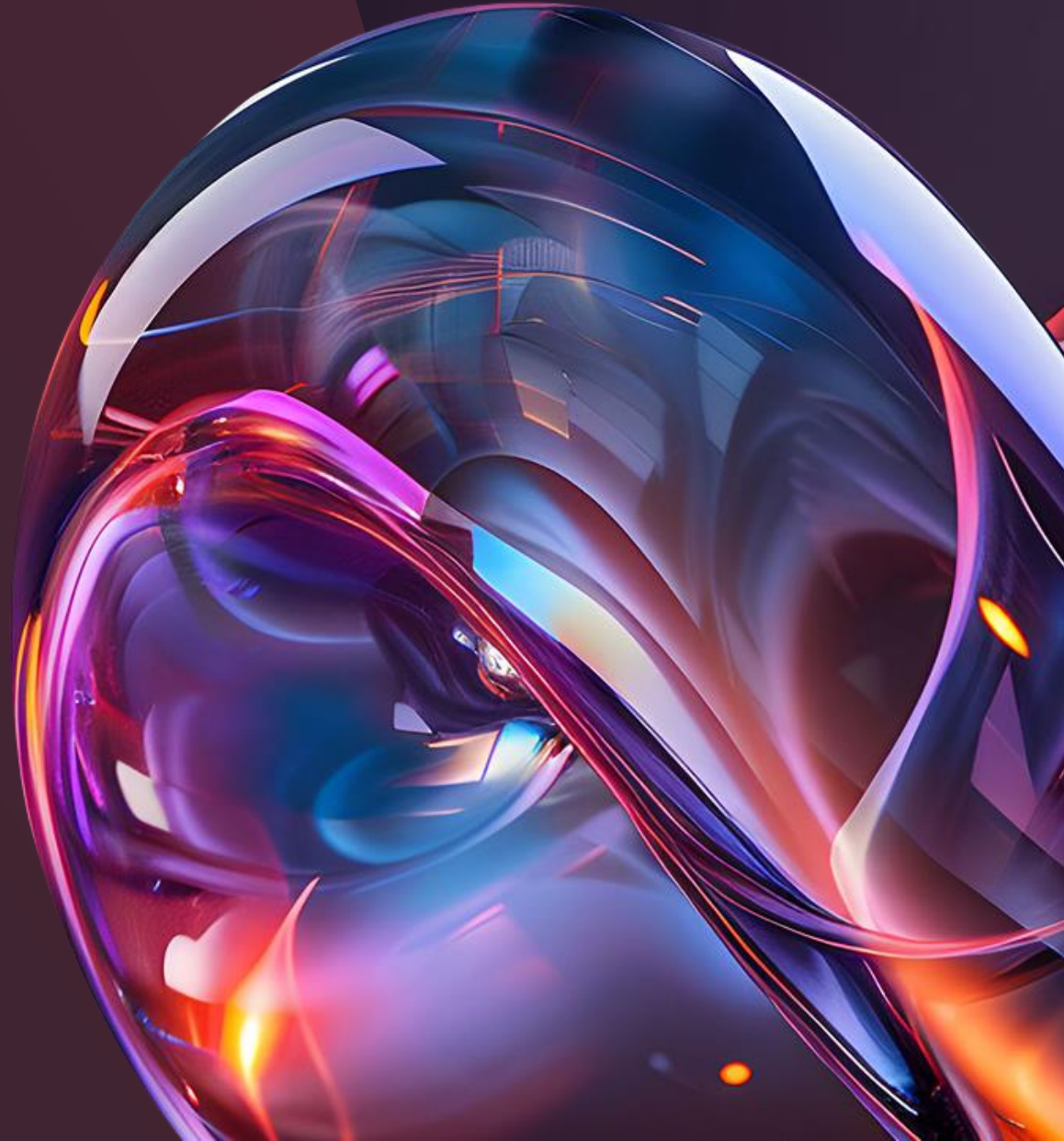


# What's ahead?

Dr. Oliver Bastert  
CTO, Gurobi



© 2025 Gurobi Optimization, LLC. Confidential, All Rights Reserved

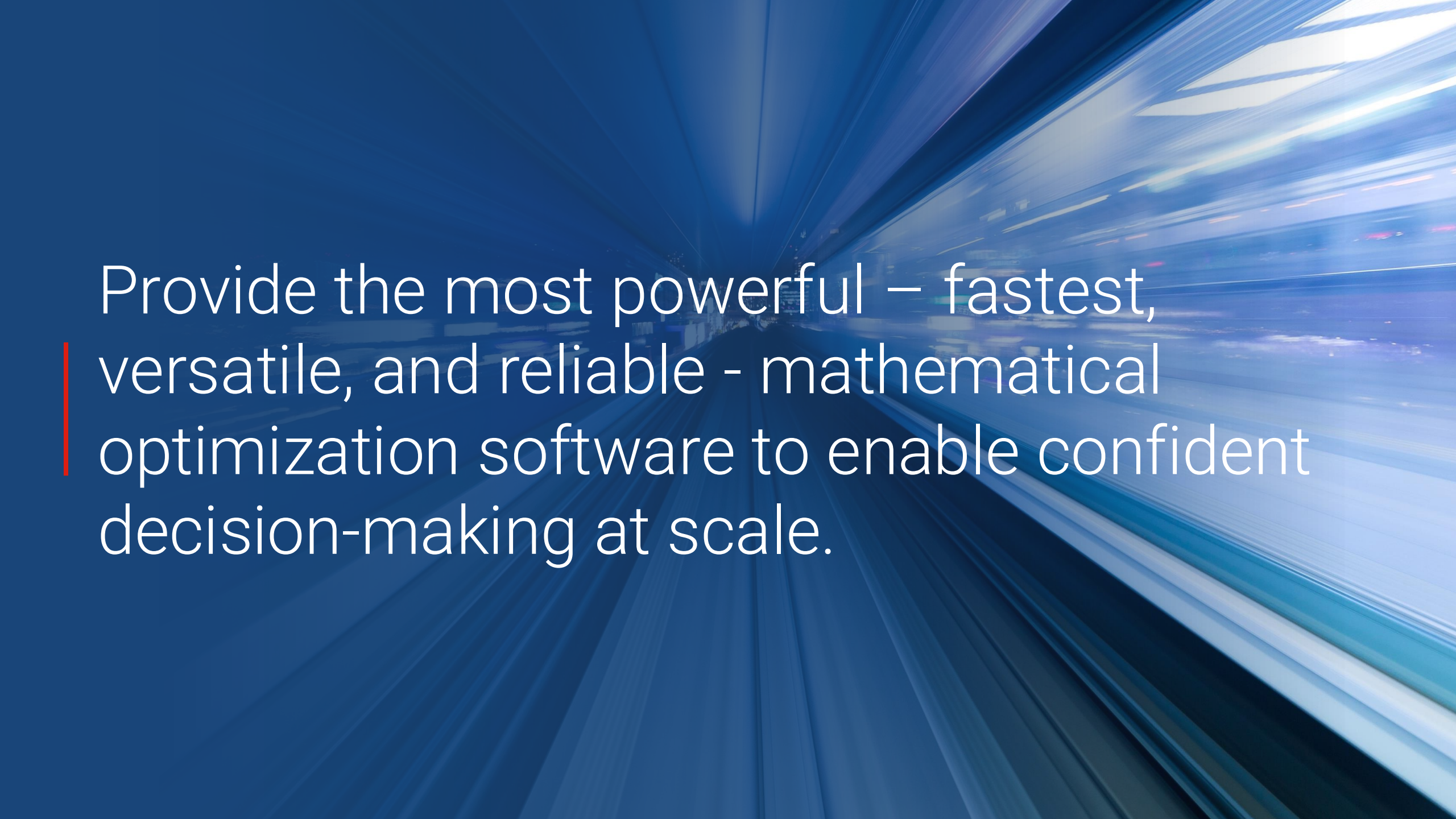




## Disclaimer

This presentation includes forward-looking statements, including, but not limited to, product roadmaps, features under consideration, and anticipated timelines. These statements are for informational purposes only and do not represent a commitment, promise, or legal obligation for Gurobi Optimization, LLC (“Gurobi”) to deliver any product, feature, or functionality to a customer or prospective customer (“Customer”). The development, release, and timing of any feature or functionality described remain at Gurobi’s sole discretion and may be changed without notice.

GUROBI DOES NOT MAKE ANY PROMISES, REPRESENTATIONS, WARRANTIES OR GUARANTEES UNDER OR ARISING FROM THIS PRESENTATION. CUSTOMER SHALL NOT RELY ON ANY INFORMATION SET FORTH IN THIS PRESENTATION. CUSTOMER SHALL NOT BE ENTITLED TO ANY RECOURSE, IN THE FORM OF DAMAGES OR OTHERWISE, ARISING FROM OR RELATING TO THIS PRESENTATION.



Provide the most powerful – fastest, versatile, and reliable - mathematical optimization software to enable confident decision-making at scale.

# Where does Gurobi stand out? \*

**01**

## Performance

Performance is dominating the decision. Related numerical robustness comes in fifth place.

**02**

## Gurobi Experts

True experts working to resolve customer challenges

**04**

## APIs

Thanks to our excellent Python API

\* MO Survey, 2025, Reasons for Switching to Gurobi

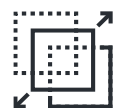
# Agenda

For today and going forward



## Performance

Performance is the #1 R&D objective.



## Versatility

Expanding on the range of problem types to address more customer problems.



## Acceleration

Make use of new hardware options, develop for GPUs, observe Quantum progress



## Velocity

Improve algorithms to find better solutions earlier.



## Integration

Expanding to making integrations and use in complex application easy.



## Guidance

Utilize GenAI for developer efficiency and business user interactions.

# Talent and Compute

Performance

**>30%**

**Growth of R&D  
last 12 months**

The most critical ingredient for creating mathematical optimization engines is excellent research & development talent.

**6x**

**Compute capacity**

Second most important is brain-power is compute capacity and our growing benchmark dataset.

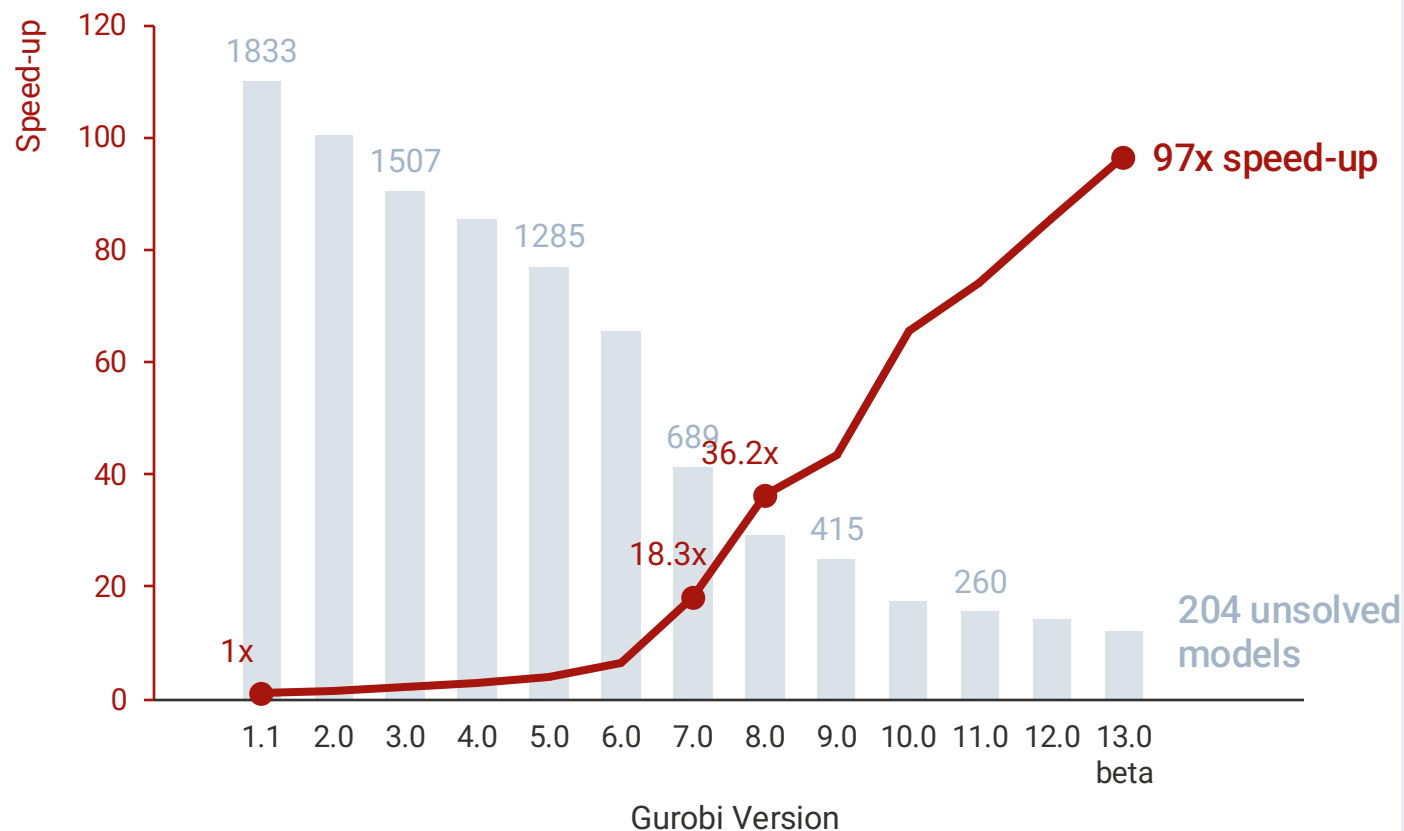
**GEMS**

**Gurobi Engineering  
MIP School**

We are now training our own MIP researchers & developers.

## Gurobi Version Comparison: Speed and Solvability (PAR-10)

Gurobi MILP Benchmark Suite



## MILP

### Performance Evolution

In Gurobi's MILP benchmark suite, the latest version delivers:

- A 97x speed-up over version 1.1 in geometric mean (PAR-10) of runtimes
- Only 204 models remain unsolved after 10,000 seconds with the latest version
- The test set consists of all models that can be solved by at least one version

Time limit: 10000 sec.  
Intel Xeon CPU E3-1240 v5 @ 3.50GHz  
4 cores, 8 hyper-threads  
32 GB RAM

Test set has 8266 models:  
- 798 discarded due to inconsistent answers  
- 2237 discarded that none of the versions can solve  
- speed-up measured on >100s bracket: 3113 models

# Nonlinear

Versatility / Performance

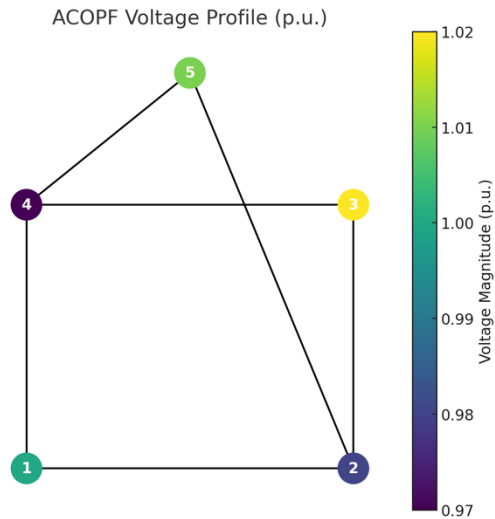
- New NL barrier implementation to
  - Solve (continuous) NLPs
  - Improve MINLP performance
- We will release the NL barrier as a preview feature in v13
- Expect NL barrier to be competitive across the board in 2026

A graphic featuring the text "2X FASTER" in a bold, stylized font. The "2X" is in white with a red outline and is followed by "FASTER" in black. The "2X" has a red motion blur effect behind it, and the "FASTER" has horizontal lines above and below it, suggesting speed and performance.

*\* MINLP compared to v12*

# Optimal Power Flow: ACOPF (QCQP)

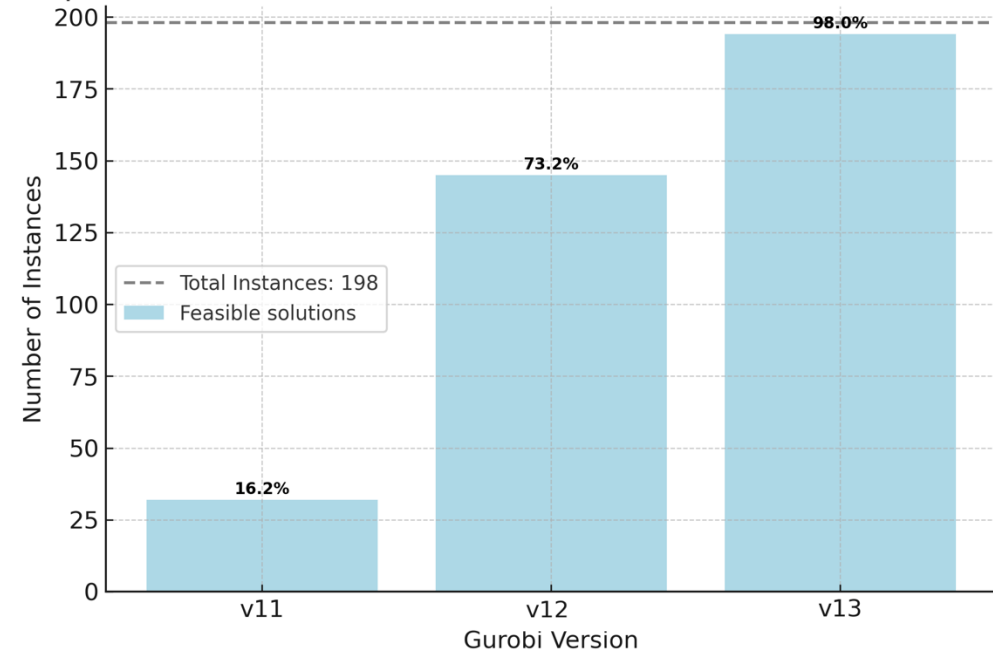
Versatility / Performance



## Why ACOPF Matters:

- Ensures economical, secure operation of power systems
- Represents the true physics of grid operations

ACOPF OptiMods on PGLIB Benchmark (total of 198 instances, TimeLimit=600)



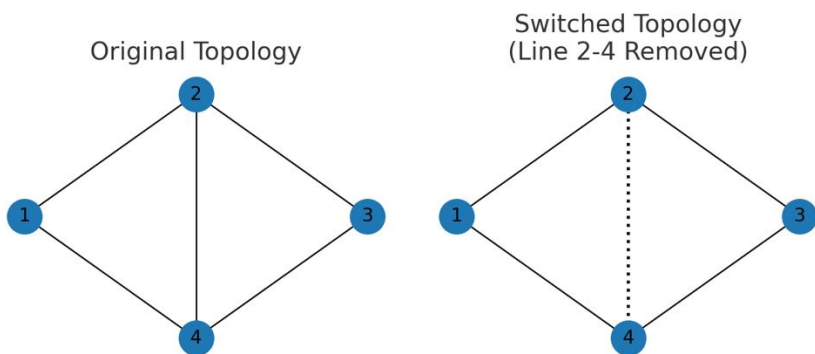
ACR: rectangular formulation (QCQP)

PGLIB Benchmarks: <https://github.com/power-grid-lib/pglib-opf>

Further performance enhancements expected with Polar ACOPF formulation

# Optimal Power Flow: DCOPF Topology Optimization

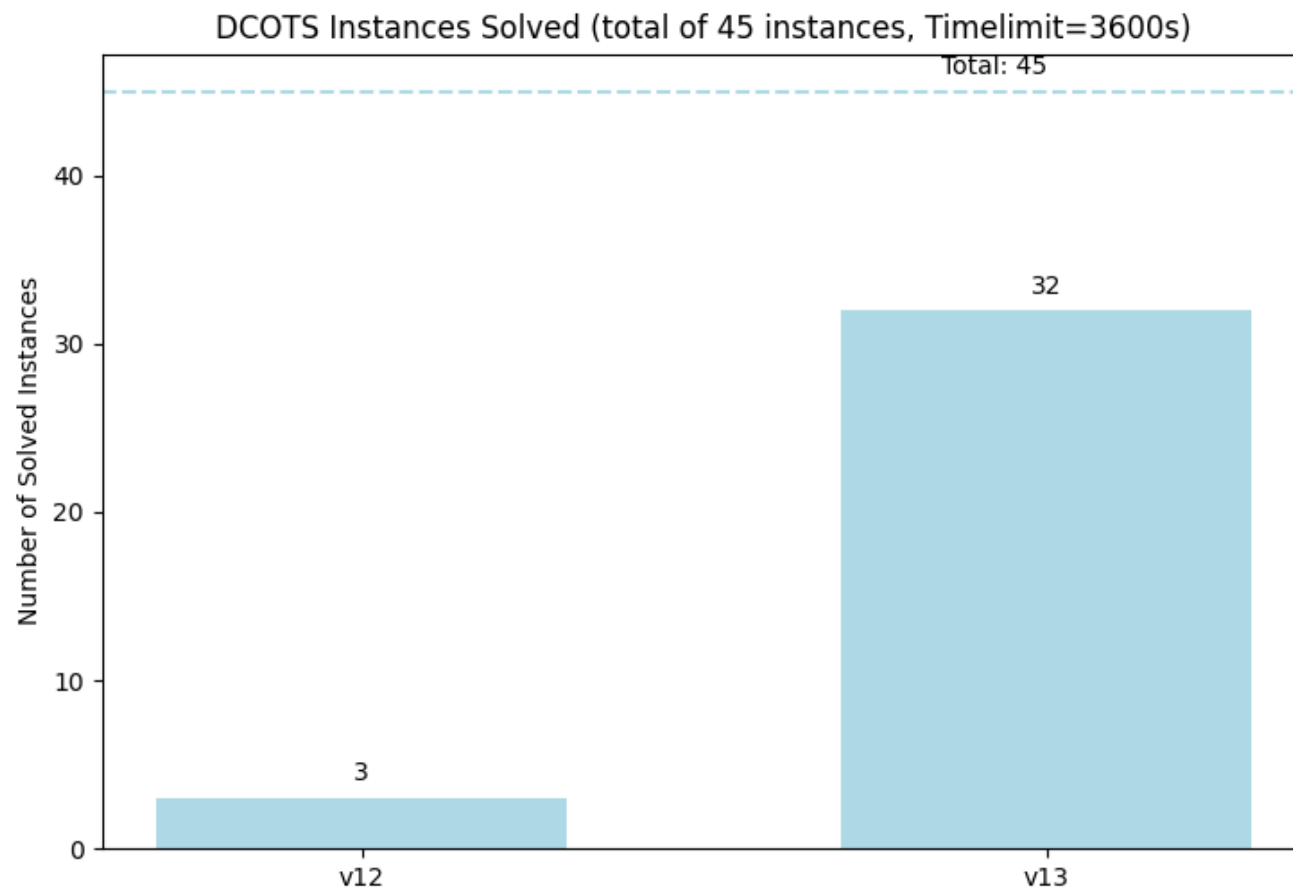
Performance



$$0 \leq p_{ij} \leq Mz_{ij}$$

## Why Topology Optimization Matters:

- Failure Recovery
- Reduce stress on congested lines  
(Germany alone spent EUR 4 billion on Congestion Management in 2022\*)



Further benchmark results at <https://openenergybenchmark.org/>

\* <https://www.iea.org/commentaries/grid-congestion-is-posing-challenges-for-energy-security-and-transitions>



## LP on GPUs

Acceleration

### GPU friendly algorithms required

Most existing algorithms don't work efficiently on GPUs.

Promising areas are

- Cholesky factorizations in **Barrier**
  - Limited speedup potential
- **PDHG** for solving LPs with limited accuracy
  - Basic/precise solutions require crossover
  - Increasing success on very large LPs
  - Roughly a factor of 3 difference between PDHG on CPU vs GPU\*
- Heuristics & Presolve operations

Beta version of PDHG on NVIDIA hardware is available.

Both, GPU (as preview) and CPU version of PDHG are in v13, development will continue, and further opportunities will be explored.

\* NVIDIA GH 200 Grace Hopper (GPU and CPU results)



## Good Solutions, Quickly

Velocity

Reasons are

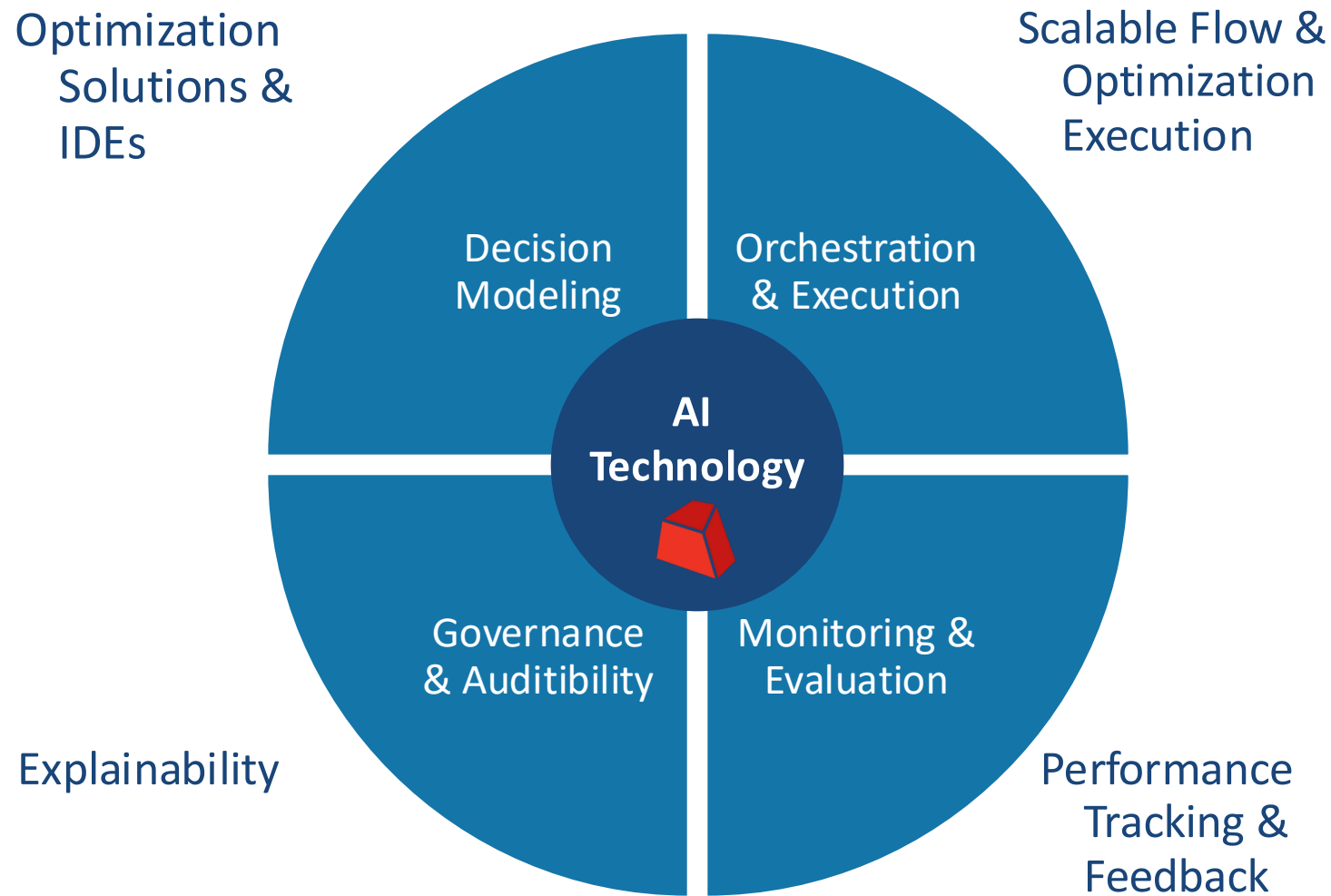
- Improving general performance
- Good solutions are good enough
- Proving optimality is hard
- Need for an (unreasonably) fast solutions

An enhanced RINS heuristic with inspiration from the Feasibility Pump will be available in v13.

Finding better solution earlier is high priority going forward.

# Decision Intelligence

Integration



# **We don't build a platform but ...**

Integration

**01**

## **Integrate with Modeling tools**

Integration with classic and GenAI  
powered development environments

**02**

## **Expand Execution functionality**

We are enabling deployment in various  
environments, we plan to provide more  
native deployment options

**03**

## **Facilitate Validation and Explainability**

We will utilize GenAI for explainability,  
and we are exploring ideas for  
performance tracking and validation.

# Gurobot

Guidance

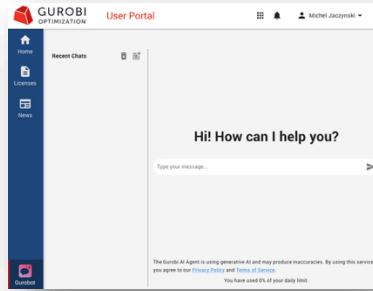
Delivers instant, expert-level assistance for its users and marks our first step in making optimization more accessible, interactive, and intelligent through Generative AI.



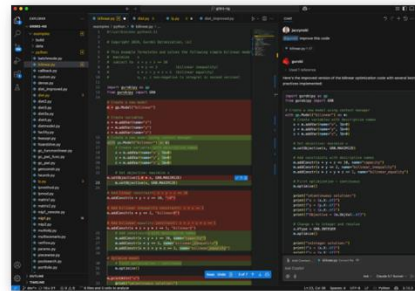
# Gurobi Intelligence – Vision

Guidance

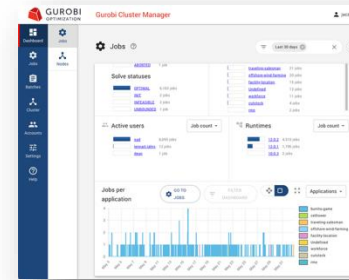
## Interfaces and Integration



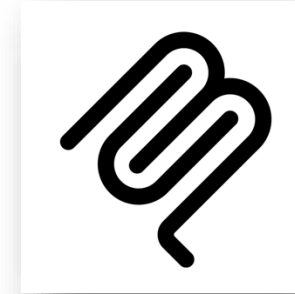
Chatbot



IDE: VSCode,  
Jetbrain...

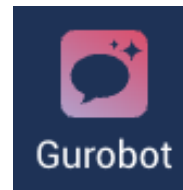


Assistants in context:  
Cluster Manager, CLI



Interoperability:  
MCP, A2A

## Orchestration and APIs



Agents for Gurobi related tasks:  
Support, Tuning, Insight, Modeling, Solving...

## Components



Knowledge Base to capture  
Gurobi best practices



Gurobi Tools:  
Optimizer, Tuner, Explainer  
Instant Cloud, Compute Server



LLMs:  
Claude, Cohere...

# Using Gurobi Intelligence in VS Code

## Guidance

```
1 #!/usr/bin/env python3.11
2
3 # Copyright 2024, Gurobi Optimization, LLC
4
5 # This example formulates and solves the following simple bilinear model:
6 # maximize x
7 # subject to x + y + z <= 10
8 #            x * y <= 2      (bilinear inequality)
9 #            x * z + y * z = 1 (bilinear equality)
10 #            x, y, z non-negative (x integral in second version)
11
12 import gurobipy as gp
13 from gurobipy import GRB
14
15 # Create a new model
16 m = gp.Model("bilinear")
17
18 # Create variables
19 x = m.addVar(name="x")
20 y = m.addVar(name="y")
21 z = m.addVar(name="z")
22
23 # Set objective: maximize x
24 m.setObjective(1.0 * x, GRB.MAXIMIZE)
25
26 # Add linear constraint: x + y + z <= 10
27 m.addConstr(x + y + z <= 10, "c0")
28
29 # Add bilinear inequality constraint: x * y <= 2
30 m.addConstr(x * y <= 2, "bilinear0")
31
32 # Add bilinear equality constraint: x * z + y * z == 1
33 m.addConstr(x * z + y * z == 1, "bilinear1")
34
35 # Add constraints with descriptive names
36 m.addConstr(x + y + z <= 10, name="capacity")
37 m.addConstr(x * y <= 2, name="bilinear_inequality")
38 m.addConstr(x * z + y * z == 1, name="bilinear_equality")
39
40 # Optimize model
41 # First optimization - continuous
42 m.optimize()
43
44 m.printAttr("x")
45 print("\nContinuous solution:")
```

```
import gurobipy as gp
from gurobipy import GRB

# Create a new model using context manager
with gp.Model("bilinear") as m:
    # Create variables with descriptive names
    x = m.addVar(name="x", lb=0)
    y = m.addVar(name="y", lb=0)
    z = m.addVar(name="z", lb=0)

    # Set objective: maximize x
    m.setObjective(x, GRB.MAXIMIZE)

    # Add constraints with descriptive names
    m.addConstr(x + y + z <= 10, name="capacity")
    m.addConstr(x * y <= 2, name="bilinear_inequality")
    m.addConstr(x * z + y * z == 1, name="bilinear_equality")

    # First optimization - continuous
    m.optimize()

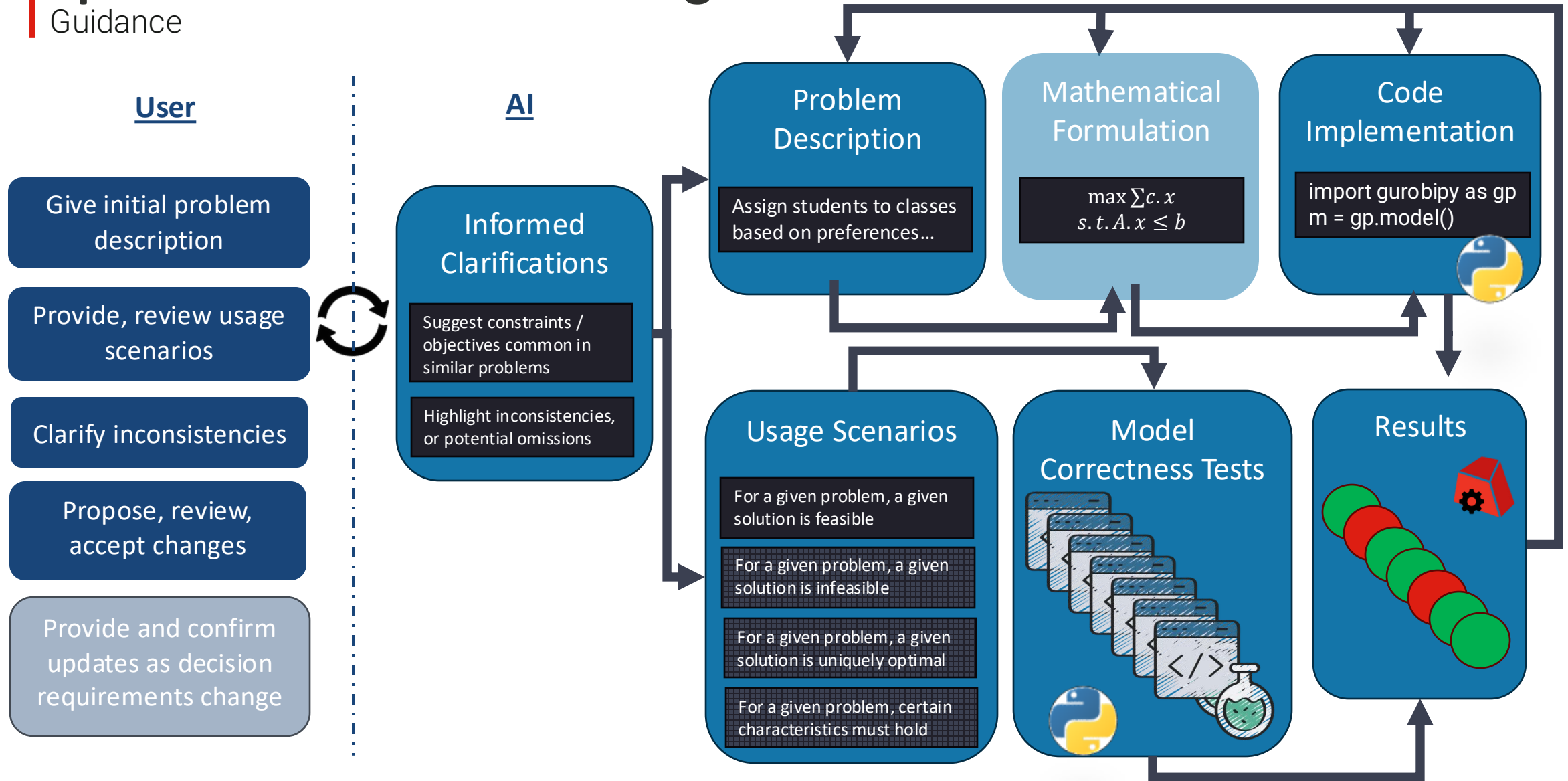
    print("\nContinuous solution:")
    print(f"x = {x.X:.6f}")
    print(f"y = {y.X:.6f}")
    print(f"z = {z.X:.6f}")
    print(f"Objective = {m.ObjVal:.6f}")

    # Change x to integer and resolve
    x.vType = GRB.INTEGER
    m.optimize()

    print("\nInteger solution:")
    print(f"x = {x.X:.6f}")
    print(f"y = {y.X:.6f}")
    print(f"z = {z.X:.6f}")
```

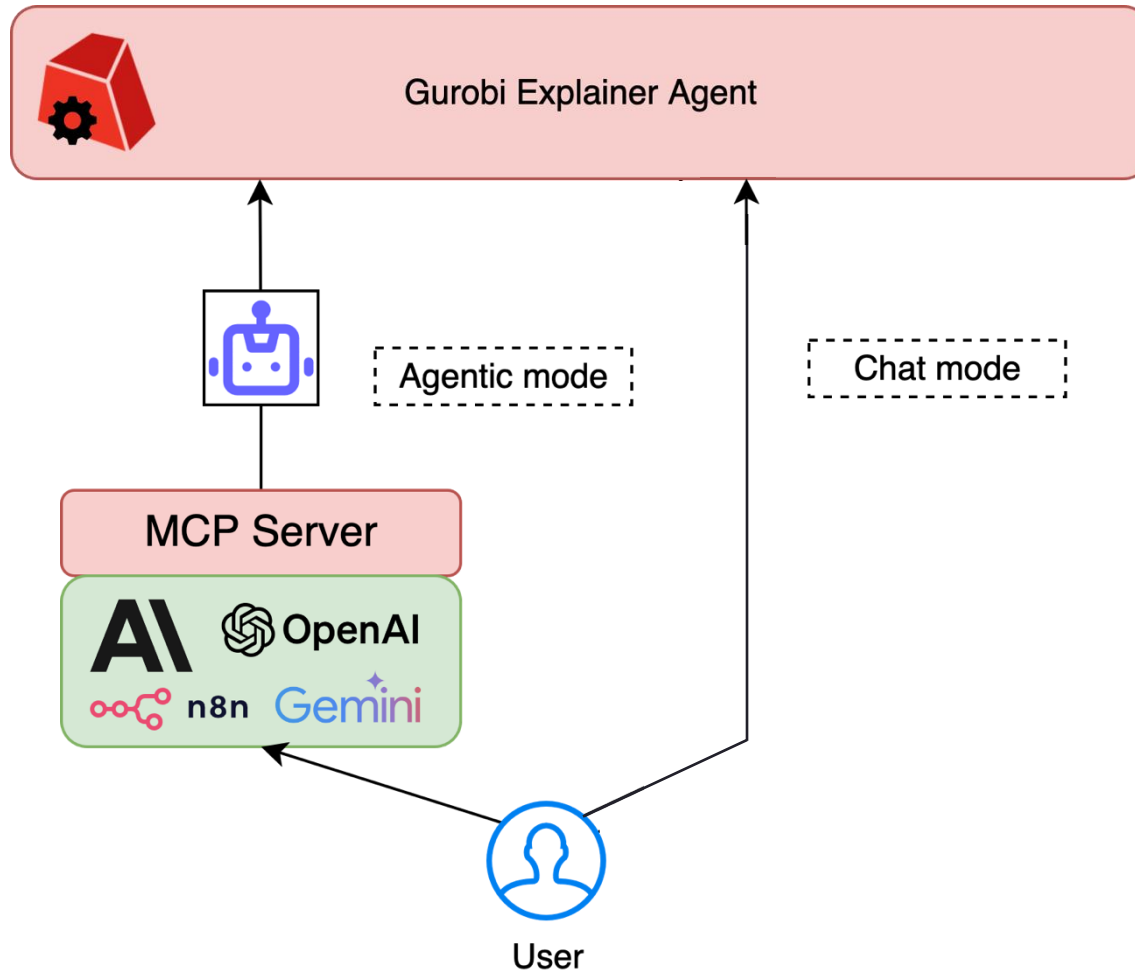
# Spec & Test based Modeling

Guidance



# Explainer

Guidance



Step-by-step contradiction:

- For each term with coefficient  $-1$  and  $UB=0$ , we have
$$x \leq 0 \Rightarrow -x \geq 0.$$
- For each term with coefficient  $+1$  and  $LB=0$ , we have
$$x \geq 0 \Rightarrow +x \geq 0.$$
- The only potentially negative contributions on the LHS come from the four bounded variables with coefficient  $-1$ :
$$-C'30 \geq -12509, \quad -C'50 \geq -12509, \quad -C'434 \geq -14434, \quad -C'474 \geq -14434.$$
- Therefore, the smallest possible LHS is
$$-12509 - 12509 - 14434 - 14434 = -53886.$$
- But the equality requires the LHS to be  $-60004$ , which is strictly smaller by
$$60004 - 53886 = 6118,$$
yielding a contradiction. No assignment can satisfy this equality under the listed bounds.

### Steps of remediation

1. Relax the RHS/constant term of the reduced equality to at least  $-53886$  (or make it less tight), or
2. Increase the upper bound of at least one of  $C'30, C'50, C'434, C'474$  so that the sum of their UBs grows by  $\geq 6118$ , or
3. Allow some of the  $UB=0$  variables to take positive values (raise their UBs) or allow some  $+coefficient$  variables to go negative (lower their LBs), and
4. Inspect the originating constraints  $c124-c171, c280-c293$  to find which modeling assumption sets those zero bounds; correct any unintended "off-switch" logic that clamps variables to zero.

🔄 ↶ 🗑️ 🗑️

Add Context... `mondou2.rlp` +

@gurobi-explainer Let me know if you have any questions about your Gurobi model.

Agent ▾ Claude Sonnet 4 ▾

# GenAI Roadmap

Guidance

**01**

## Making Developers Efficient

Improve on usability of development tools, like debugging, analysis, tuning functionality. Examples are Gurobot, Explainer Agent, Tuner Agent

**02**

## Enabling Spec & Test based Modeling

Provide the future optimization modeling experience based on recent development approaches and Gurobi specific development tools .

**03**

## Proactively guiding business users

Enable business users to directly interact and modify with optimization models and results. Proactively carry out analysis and recommend actions.

# Innovation Timeline

---

June	July	August	September	October	November	December / January 2026
<p style="text-align: center;">Gurobot Re-launch</p> <p style="text-align: center;">Innovation Lab Launch</p>	<p style="text-align: center;">GEMS Launch</p>	<p style="text-align: center;">Gurobot as First Line of Support</p>	<p style="text-align: center;">GPU Alpha</p>	<p style="text-align: center;">Gurobi v13 Public Beta</p>	<p style="text-align: center;">Gurobi v13 Release</p>	<p style="text-align: center;">Gurobi 13 GPU Release</p> <p style="text-align: center;">Explainer / MCP Available</p>

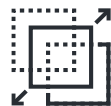
# Roadmap Summary

---



## **Performance**

Performance is the #1 R&D objective.



## **Versatility**

Expanding on the range of problem types to address more customer problems.



## **Acceleration**

Make use of new hardware options, develop for GPUs, observe Quantum progress



## **Velocity**

Improve algorithms to find better solutions earlier.



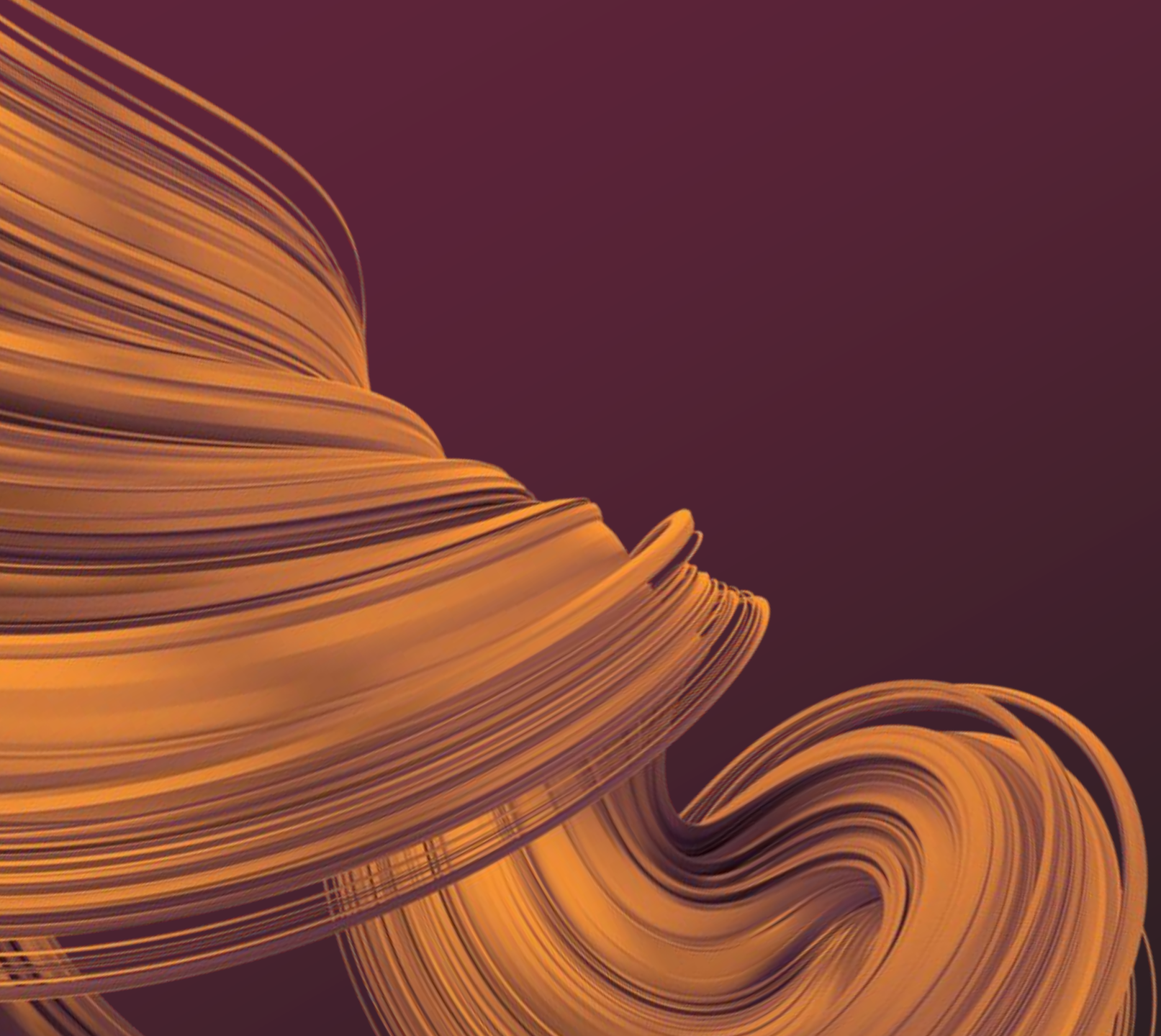
## **Integration**

Expanding to making integrations and use in complex application easy.



## **Guidance**

Utilize GenAI for developer efficiency and business user interactions.



**Thank You**