2026 FERC SOFTWARE CONFERENCE

JUL 8TH, 2026

AUTOMATE GRID PLANNING WITH AGENTIC AI

WEI GAO
Associate Computational Scientist
Argonne National Laboratory

ALINSON SANTOS XAVIER
Principal Computational Scientist
Argonne National Laboratory

FENG QIU
Group Leader, Principal Computational Scientist
Argonne National Laboratory

THE GRID PLANNING BOTTLENECK

- Over 2,060 GW of generation and storage waiting to connect (end of 2025)[1]
- Surging, uncertain load (AI data centers)
- FERC Order 1920 → broader scenario coverage **required**
- **Long wait:** about 5 years from request to operation, and growing.
- **Most withdraw:** roughly 7 in 10 projects drop out before they get built.

Yet planning workflows are unchanged:

  - manual scenarios, ad-hoc scripts, rigid tools

[1] Berkeley Lab, Queued Up 2025 Edition. https://emp.lbl.gov/queues

HOW CAN AI HELP GRID PLANNING?

- **Approach one: AI as a solver**
  - A legitimate question: success in many domains, e.g., protein folding
    - Remain discovery and predictive tools
  - Power system is a physics-governed system
    - Accuracy is very important. (Alphafold is only 90% accurate)
    - Physics-informed learning is still in preliminary stage
  - Conventional computers cannot solve NP-hard problems
    - Most power system problems are NP-hard
  - Black-box solutions won't be accepted by power sector
    - E.g., Market clearing
- **Approach two: AI as workflow orchestrator**
  - Leverage existing tools and integrate with agentic AI

WHY WE TURNED TO AGENTIC AI

- **Grid Planning is mostly orchestration of existing tools:** set up cases, run simulations, compare results, write the reports.
- **Each step is well-defined:** the bottleneck is efficiently putting the steps together.
- **Modern Agentic AI is good at this:** it can plan, call tools, read results, and try again (autonomously or semi-autonomously – Human-in-the-Loop).
- **Goes beyond orchestration:** in-depth discussion/analysis with human operators.

Existing power system analysis tools do the math.

The Agentic AI does the routine busywork.

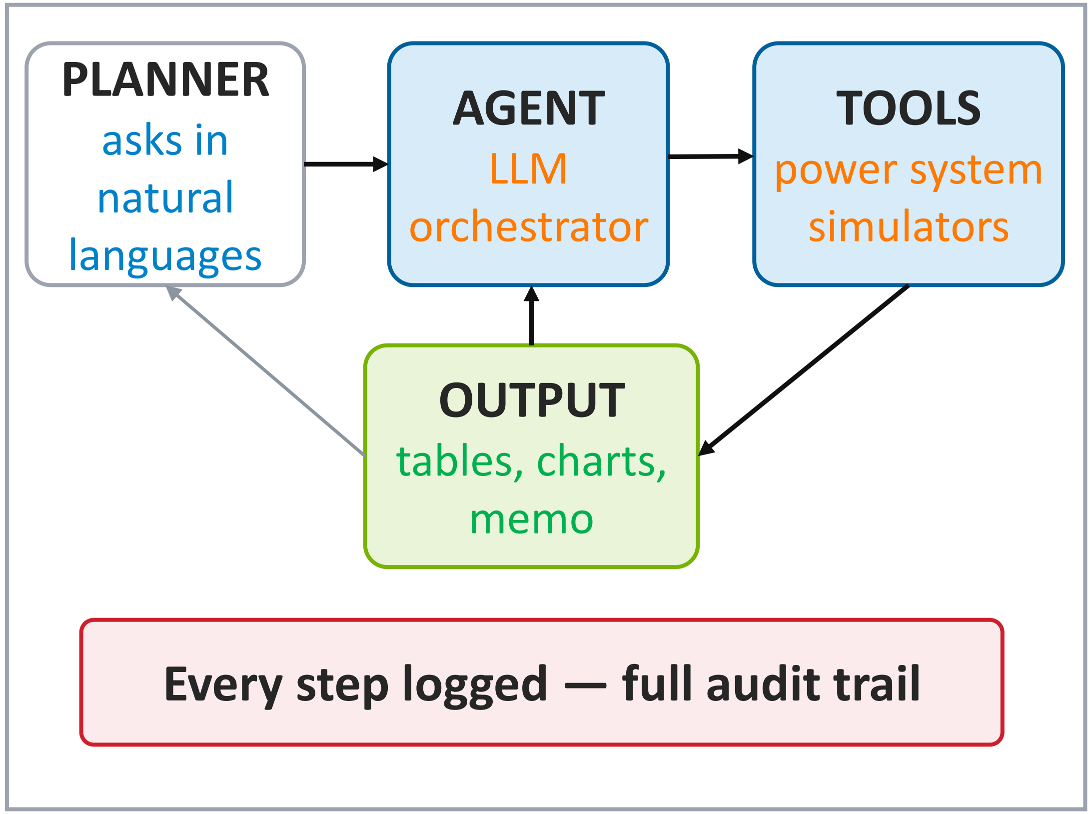
A NEW WAY TO POWER GRID PLANNING

AGENTIC AI FOR POWER GRID STUDIES

GRID PLANNING – TRADITIONAL VS. AGENTIC

- **Traditional Grid Planning**
  - 1. Define study scope manually
  - 2. Collect and prepare data
  - 3. Build study cases manually
  - 4. Run simulation tools
  - 5. Analyze violations manually
  - 6. Develop mitigation plans
  - 7. Prepare reports and presentations
  - 8. Human review and approval
- **Agentic Grid Planning**
  - 1. Planner expresses planning **intent** in natural language
  - 2. Agentic agents **automatically** gather and organize data
  - 3. Agentic agents **generate** study cases automatically
  - 4. Agentic agents **orchestrates** simulation workflows
  - 5. Agentic agents **identifies** risks and root causes
  - 6. Agentic agents **proposes** candidate solutions
  - 7. Agentic agents **generates** decision-ready insights
  - 8. Human-in-the-loop decision making

Agentic Grid Planning



OUR APPROACH VS OTHERS

	Fixed tool-calling / **MCP**	**Fine-tuned** model	**LLM as solver**	Ours
Examples	Harvard - PowerMCP[1]	NLR - eGridGPT[2]	research only	this work
Functionality	do predefined functionalities	give decision suggestions, validate in simulation	compute the answer	**writes code to the native API (full capabilities and compossibilities)**
Extensibility	new coding work for new functions	re-fine-tune	— (re-train?)	**write a new document with instructions**
Math	real solver	real solver, partially	LLM (unreliable)	**real solver**
Trust	highly reliable	black-box, weights	black-box	**inspectable code + validation + sandbox**

[1] Harvard - PowerMCP. https://poweragent.seas.harvard.edu

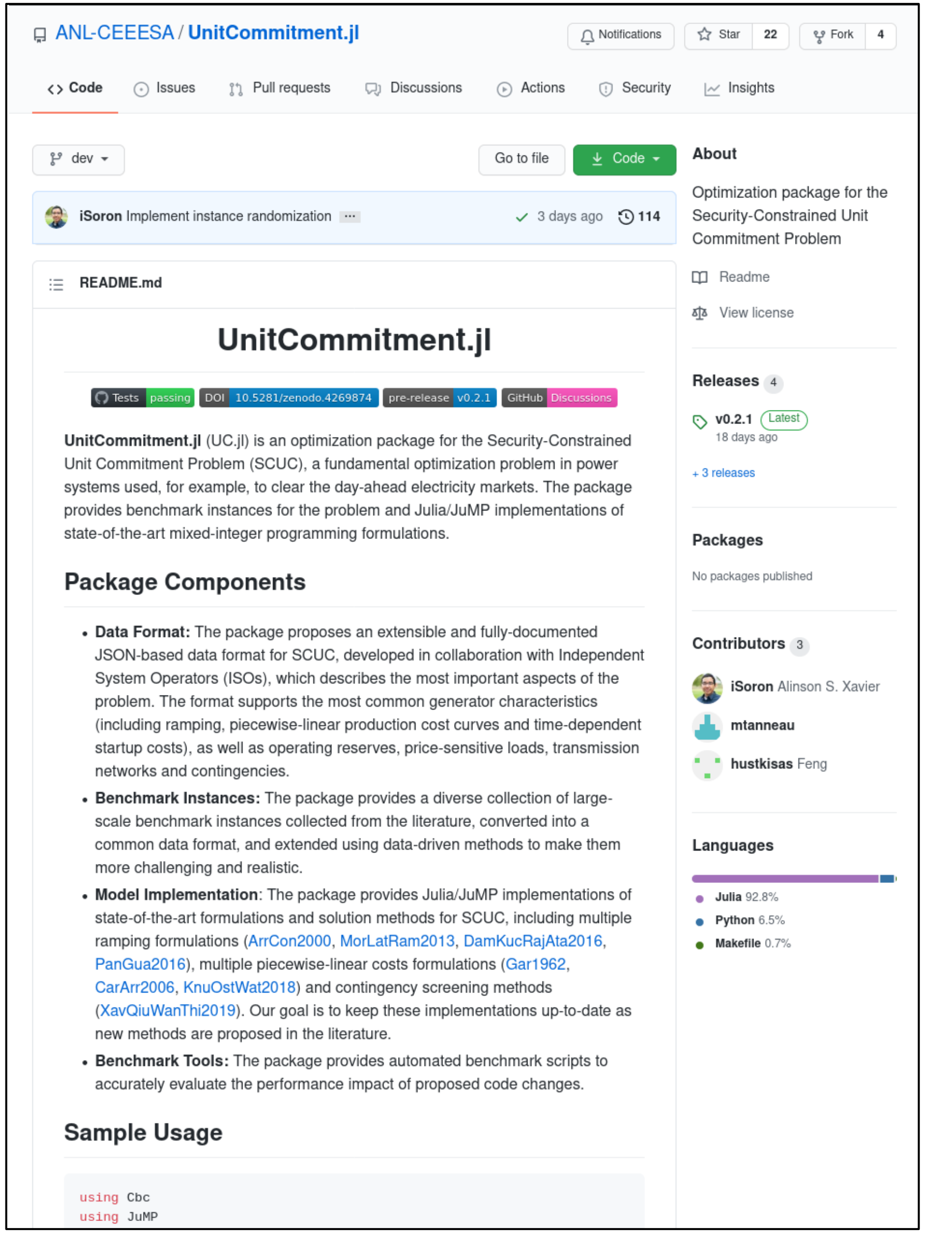
[2] NRL - eGridGPT. https://research-hub.nlr.gov/en/publications/egridgpt-trustworthy-ai-in-the-control-room

DEMO

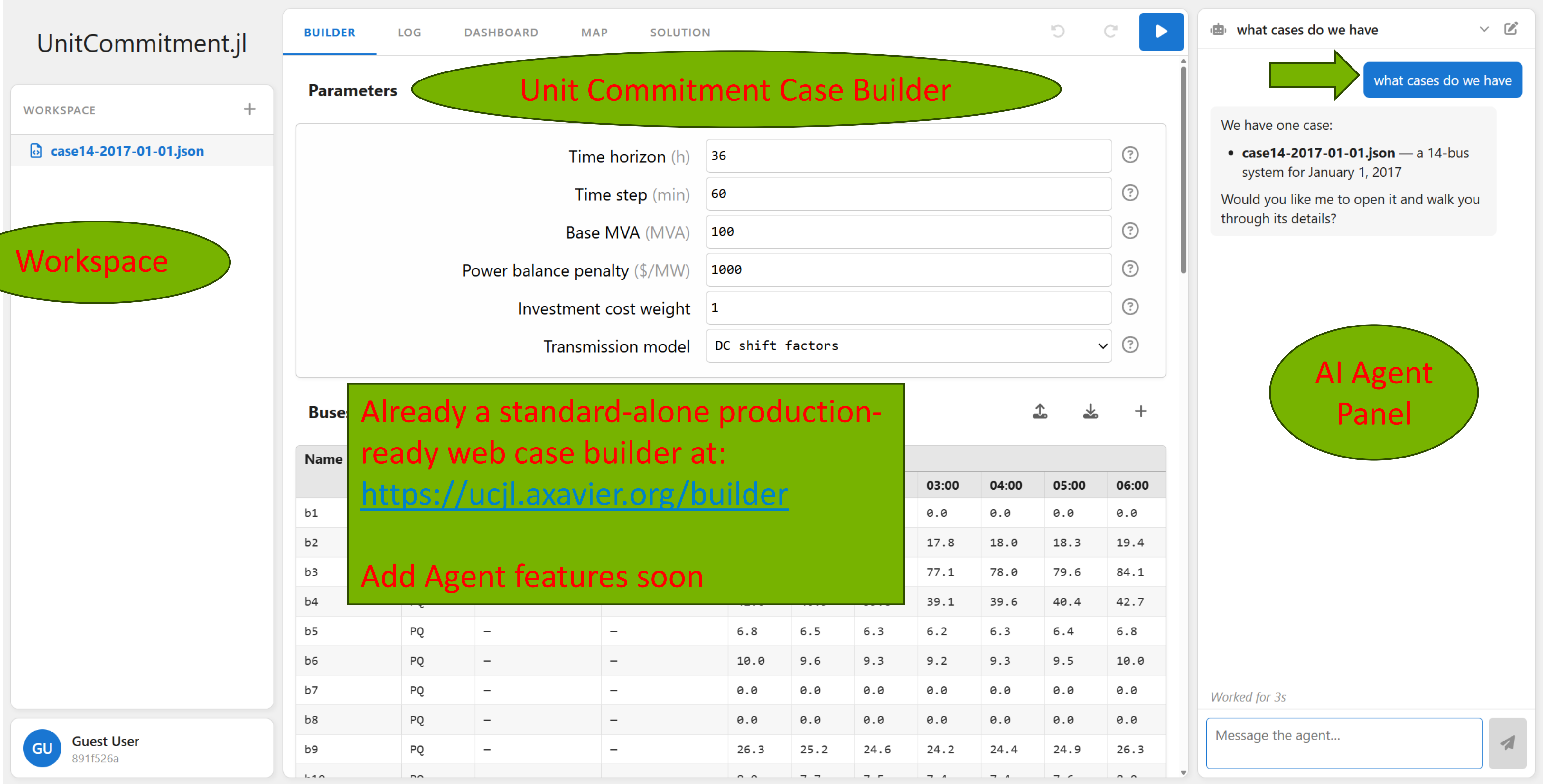
THE FUNDAMENTAL TOOLS

- **UnitCommitment.jl**
  - Source code: https://github.com/ANL-CEEESA/UnitCommitment.jl
  - Documentation: https://anl-ceeesa.github.io/UnitCommitment.jl
  - Benchmark instances: https://axavier.org/UnitCommitment.jl/0.4/instances

- **UnitCommitmentWeb**
  - Web UC case builder: https://ucjl.axavier.org/builder



First Glance



Unit Commitment Case Builder

Workspace

AI Agent Panel

Already a standard-alone production-ready web case builder at: https://ucjl.axavier.org/builder

Add Agent features soon

Ask for a comprehensive summary of the case

what cases do we have

what cases do we have

We have one case:

- **case14-2017-01-01.json** — a 14-bus system for January 1, 2017

Would you like me to open it and walk you through its details?

work on the case14 one, summarize it for me

Here's a summary of the **case14-2017-01-01** case:

Parameters

- **Time horizon:** 36 hours | **Time step:** 60 min (36 intervals)
- **Base MVA:** 100 | **Power balance penalty:** $1,000/MW

Network — 14 Buses, 20 Branches

- Buses **b1–b14**, connected by 20 transmission lines (l1–l20)
- Branch flow limits and reactances are not set (undefined)
- **b1** is the reference/slack bus; **b2–b14** carry load

Generators — 5 Thermal Units

High level summary

what cases do we have

Generators — 5 Thermal Units

Unit	Bus	Ramp (MW/h)	Min Up/Down (h)	Max prod curve point (MW)
g1	b1	231.1	1 / 1	~330 MW
g2	b2	92.6	1 / 1	~132 MW
g3	b3	66.2	1 / 1	~95 MW
g4	b6	66.9	**4 / 4**	~96 MW
g5	b8	65.0	1 / 1	~93 MW

- All units have piecewise-linear production cost curves (5 points each) and 3-level startup costs
- Min/Max power limits are **not set**
- **g4** is the most constrained — 4-hour minimum uptime/downtime

Load Profile

- Load exists on **11 of 14 buses** (b1 is zero-load, likely generator-only)
- **b3 carries the heaviest load** — peaks at ~105 MW around hour 18
- System peak appears around **hour 18** across all buses
- Total system peak is roughly ~270

Worked for 38s

native package APIs

not raw file read

accurate efficient

what cases do we have

g5	b8	65.0	1 / 1	~93 MW

- All units have piecewise-linear production cost curves (5 points each) and 3-level startup costs
- Min/Max power limits are **not set**
- **g4** is the most constrained — 4-hour minimum uptime/downtime

Load Profile

- Load exists on **11 of 14 buses** (b1 is zero-load, likely generator-only)
- **b3 carries the heaviest load** — peaks at ~105 MW around hour 18
- System peak appears around **hour 18** across all buses
- Total system peak is roughly **~270 MW** summed across all loaded buses

Other Components

- **Storage units:** None
- **Price-sensitive loads:** None
- **Shunt devices:** None
- **Reserves/Interfaces:** None defined

In short: A classic 14-bus IEEE-style test case with 5 thermal generators, realistic 36-hour load profiles, but missing branch limits and generator min/max power settings. Want me to fill in any of those missing values or make other edits?

Worked for 38s

Complex CRUD operations

what cases do we have

settings. Want me to fill in any of those missing values or make other edits?

ok, what are the top 3 most average loaded buses

The top 3 most average loaded buses are:

Rank	Bus	Avg Load
🥇 1	**b3**	89.23 MW
🥈 2	**b4**	45.28 MW
🥉 3	**b9**	27.94 MW

b3 is by far the dominant load center — roughly twice the load of b4. Want to do anything with these buses?

can you decrease the load for them by 5~10% randomly, while make the rest increase by 10~20% randomly

Done! Here's a before/after comparison:

Bus	Before	After	Change
b3	89.23 MW	82.72 MW	↓ ~7.3%
b4	45.28 MW	41.93 MW	↓ ~7.4%
b9	27.94 MW	25.82 MW	↓ ~7.6%
b2	20.35 MW	23.69 MW	↑ ~16.4%
b5	7.22 MW	8.28 MW	↑ ~14.7%

Worked for 20s

not single checkout

but composability

what cases do we have

🥉 3	**b9**	27.94 MW

b3 is by far the dominant load center — roughly twice the load of b4. Want to do anything with these buses?

can you decrease the load for them by 5~10% randomly, while make the rest increase by 10~20% randomly

Done! Here's a before/after comparison:

Bus	Before	After	Change
b3	89.23 MW	82.72 MW	↓ ~7.3%
b4	45.28 MW	41.93 MW	↓ ~7.4%
b9	27.94 MW	25.82 MW	↓ ~7.6%
b2	20.35 MW	23.69 MW	↑ ~16.4%
b5	7.22 MW	8.28 MW	↑ ~14.7%
b6	10.65 MW	12.17 MW	↑ ~14.3%
b10	8.57 MW	9.84 MW	↑ ~14.8%
b11	3.33 MW	3.80 MW	↑ ~14.1%
b12	5.81 MW	6.65 MW	↑ ~14.5%
b13	12.86 MW	14.72 MW	↑ ~14.5%
b14	14.19 MW	16.26 MW	↑ ~14.6%

All changes were applied per time step with random factors within the specified ranges.

Worked for 20s

Solve the case using the real solver

what cases do we have

All changes were applied per time step with random factors within the specified ranges.

great, so solve the case and summarize the results for me

Submitted! Job ID is **a45afd8e63e8315a** — waiting for the solver to finish and I'll summarize the results.

The solve (job a45afd8e63e8315a) has finished. Read it with jobStatus('a45afd8e63e8315a') and summarize the result.

Here's the full results summary:

Solver Status

- **Optimal** solution found in **0.15 seconds**
- Optimality gap: **0%**
- Total objective cost: **$361,563**

Generator Dispatch

Unit	Status	Hours On	Production	Avg Utilizat
g1	ON	36/36 h	7,094 MWh	59.7%
g3	ON	36/36	1,758	51.6%

Worked for 32s

Asynchronous for long solving

what cases do we have

summarize the result.

Here's the full results summary:

Solver Status

- **Optimal** solution found in **0.15 seconds**
- Optimality gap: **0%**
- Total objective cost: **$361,563**

Generator Dispatch

Unit	Status	Hours On	Production	Avg Utilizat
g1	ON	36/36 h	7,094 MWh	59.7%
g3	ON	36/36 h	1,758 MWh	51.6%
g2	OFF	0 h	—	—
g4	OFF	0 h	—	—
g5	OFF	0 h	—	—

Only **g1** and **g3** were committed for the full 36-hour horizon. g3 had 1 startup ($5,235). The other 3 generators were not needed.

System Load

Worked for 32s

what cases do we have

full 36-hour horizon. g3 had 1 startup ($5,235). The other 3 generators were not needed.

System Load

- **Peak load:** 288.92 MW | **Minimum load:** 210.54 MW
- **No load curtailment** anywhere — all demand fully served
- **Total load expense:** $396,967

Prices (LMP)

- System-wide LMPs are **uniform across all 14 buses** (no congestion)
- **Off-peak price:** $44.43/MWh (hours 0–6, 24–29)
- **On-peak price:** $45.05/MWh (hours 7–23, 30–35)
- **Average LMP:** $44.84/MWh

Network

- **No branch congestion or overflow** on any of the 20 lines (branch flow limits are unset, so unconstrained)
- **No reserve shortfalls**

In short: A clean, low-cost solve. Two generators handle all demand comfortably, prices are nearly flat, and the network is fully uncongested. Would you like to make any changes or run further analysis?

Worked for 32s

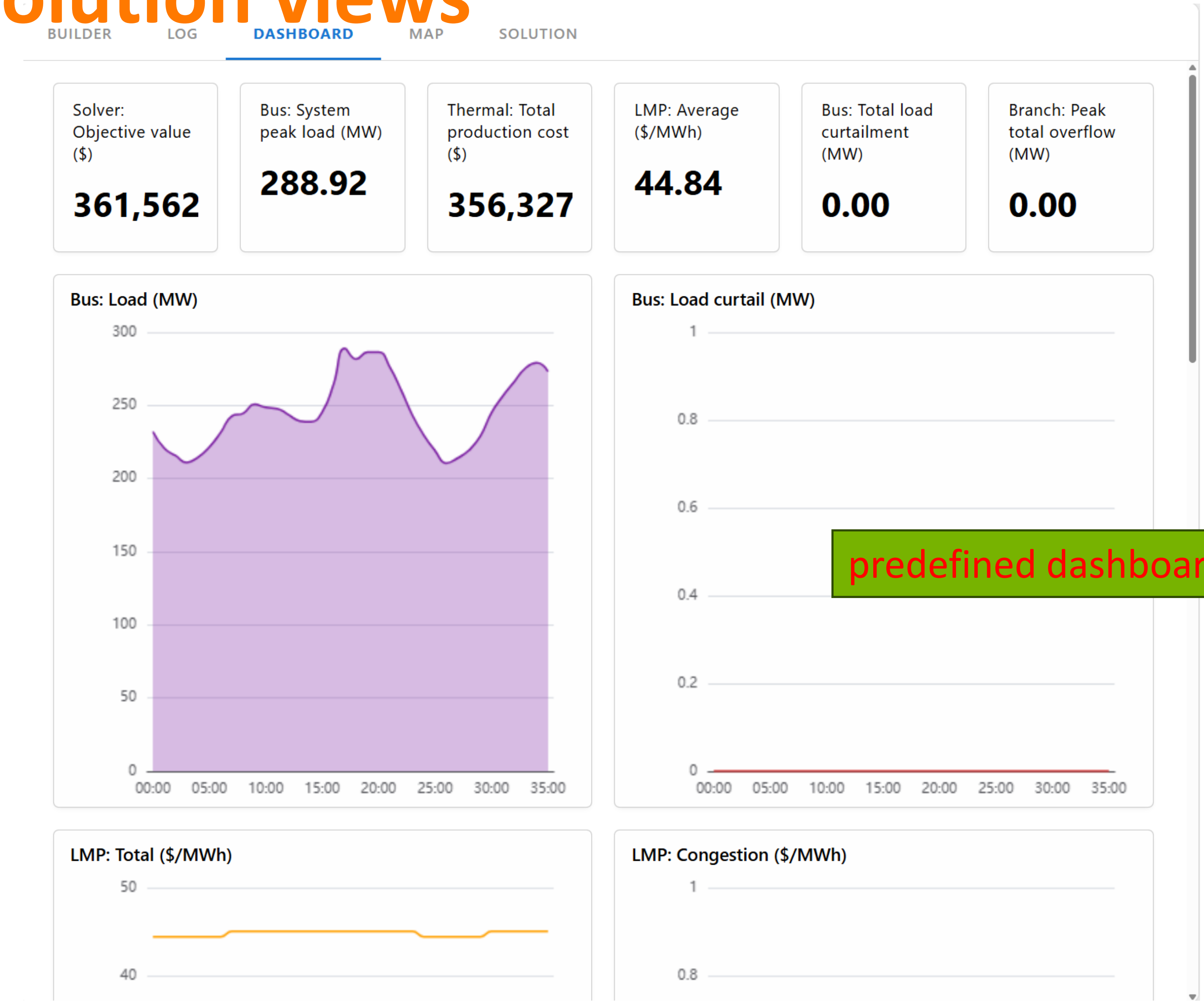
Rich UC solution views

BUILDER LOG DASHBOARD MAP SOLUTION

problem solving logs

```
Running HiGHS 1.14.0 (git hash: 7df0786de3): Copyright (c) 2026 under Apache 2.0 license terms
Using BLAS: blastrampoline
MIP has 4661 rows; 5076 cols; 13960 nonzeros; 1080 integer variables (1080 binary)
Coefficient ranges:
  Matrix  [1e+00, 3e+02]
  Cost    [3e+01, 3e+04]
  Bound   [1e+00, 1e+02]
  RHS     [1e+00, 4e+02]
Presolving model
3795 rows, 2704 cols, 11951 nonzeros 0s
3119 rows, 1982 cols, 10721 nonzeros 0s
3099 rows, 1971 cols, 10895 nonzeros 0s
Presolve reductions: rows 3099(-1562); columns 1971(-3105); nonzeros 10895(-3065)

Solving MIP model with:
   3099 rows
   1971 cols (868 binary, 0 integer, 0 implied int., 1103 continuous, 0 domain fixed)
   10895 nonzeros

Src: B => Branching; C => Central rounding; F => Feasibility pump; H => Heuristic;
     I => Shifting; J => Feasibility jump; L => Sub-MIP; P => Empty MIP; R => Randomized rounding;
     S => Solve LP; T => Evaluate node; U => Unbounded; X => User solution; Y => HiGHS solution;
     Z => ZI Round; l => Trivial lower; p => Trivial point; u => Trivial upper; z => Trivial zero

        Nodes      |    B&B Tree     |            Objective Bounds              |  Dynamic Constraints |
Work
Src  Proc. InQueue |  Leaves   Expl. | BestBound       BestSol              Gap |   Cuts   InLp Confl. | LpIters
Time

         0       0         0   0.00%   0               inf                  inf        0      0      0         0
0.1s
 R       0       0         0   0.00%   361562.745301   361562.949657    0.00%        0      0      0       947
0.1s
         1       0         1 100.00%   361562.745301   361562.949657    0.00%        0      0      0       947
0.1s

Solving report
  Status            Optimal
  Primal bound      361562.949657
  Dual bound        361562.745301
  Gap               5.7e-05% (tolerance: 0.1%)
  P-D integral      7.14470442449e-10
  Solution status   feasible
```

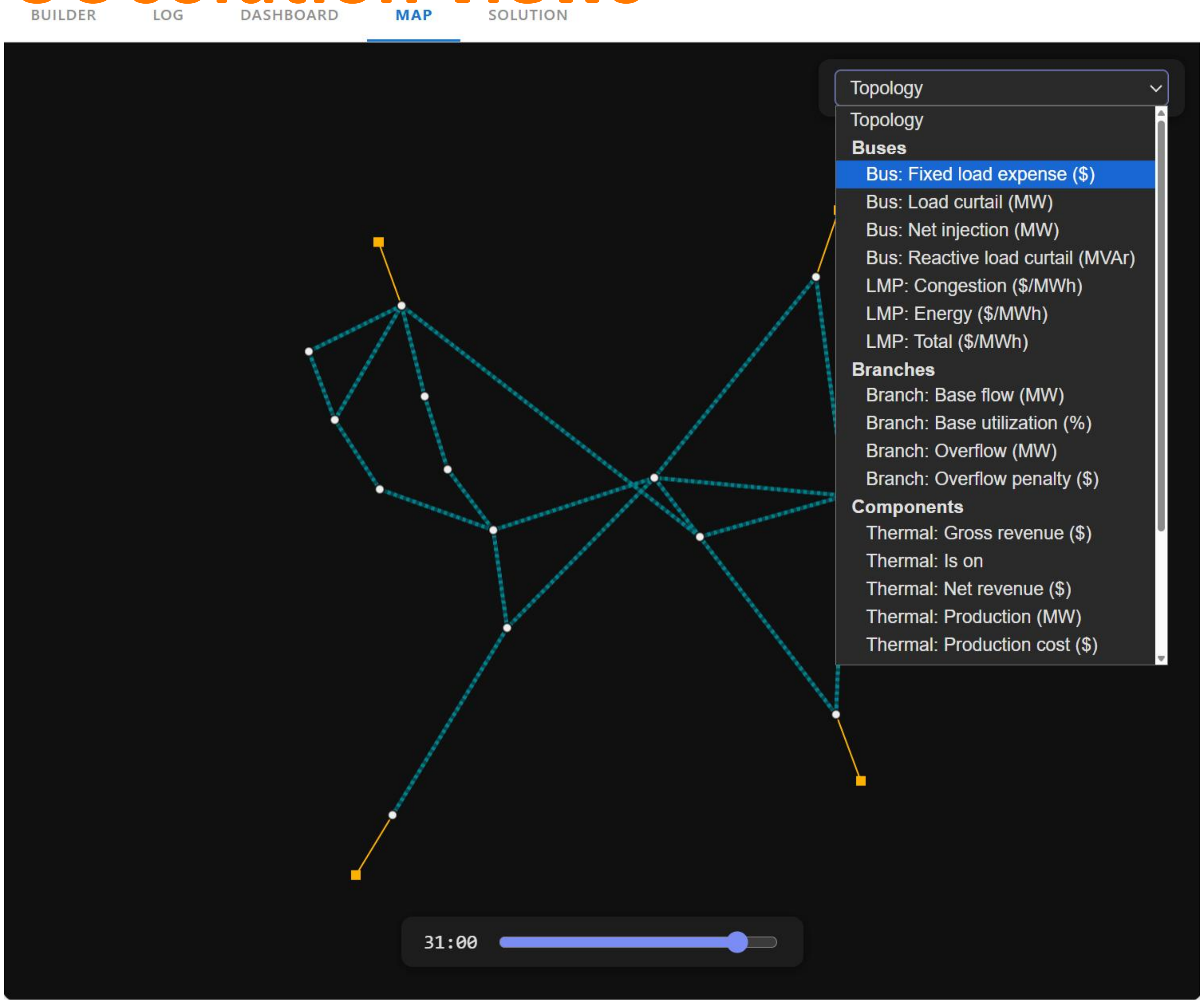
Rich UC solution views



predefined dashboard for key metrics

U.S. DEPARTMENT of ENERGY
Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.
Argonne NATIONAL LABORATORY

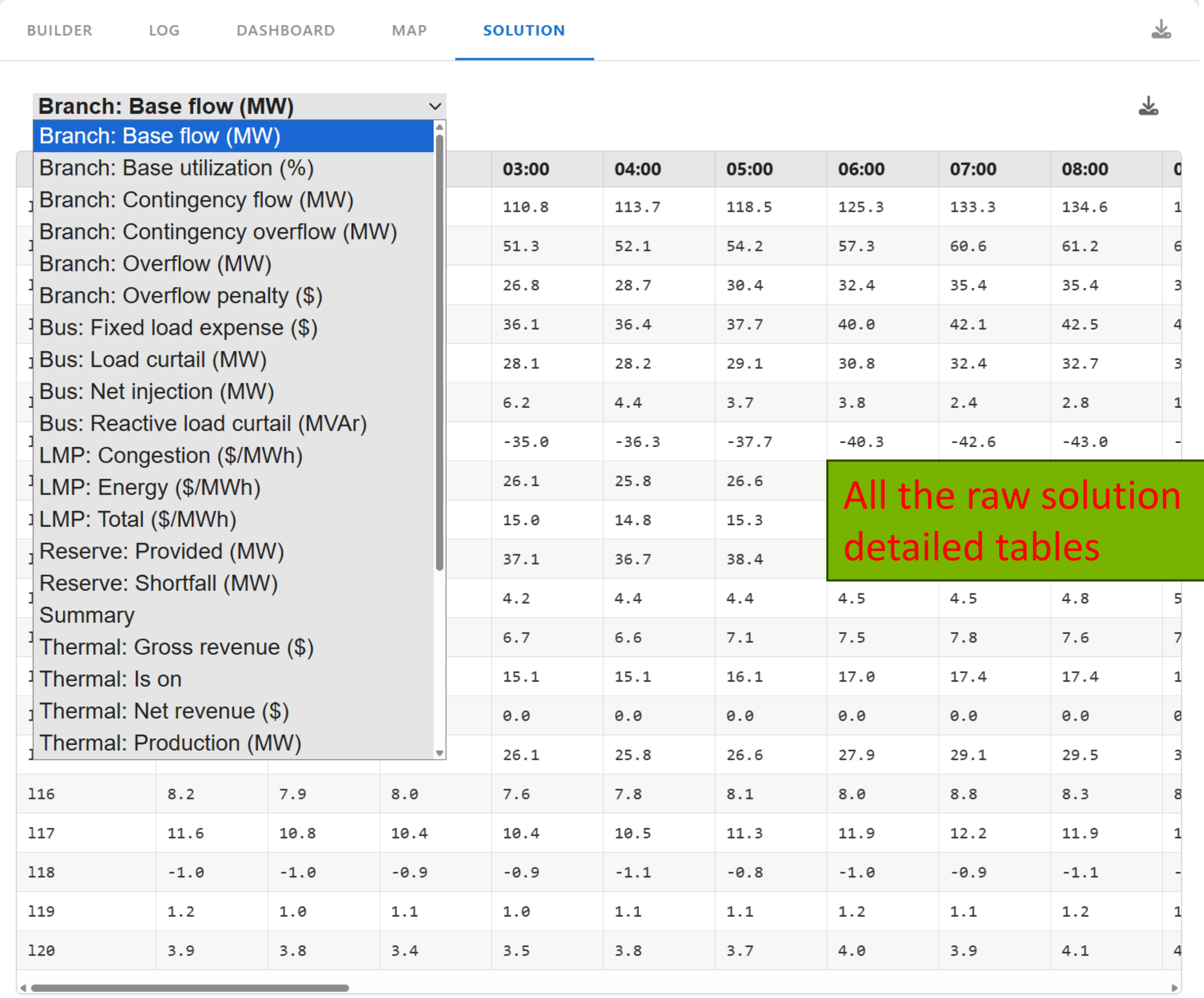
Rich UC solution views



case topology map

system solution states as time evolves

Rich UC solution views



All the raw solution detailed tables

LLM Pretrained Power System Knowledge

what cases do we have

why no branch overflow and load curtailment

The answer is straightforward — it comes down to two things visible from our earlier case inspection:

1. **No branch flow limits set** — all 20 branches have `Normal flow limit (MW)` = `undefined`. Without limits, the solver treats lines as having infini[te] capacity, so overflow is mathematica[lly] impossible.
2. **No generator min/max power limi[ts] set** — all 5 generators have `Minimum power (MW)` and `Maximum power (MW)` = `undefined`. Without an upp[er] cap, g1 and g3 can produce as much power as needed to meet any load level, so load curtailment never occurs.

In short, the case is essentially **unconstrained** on both the network and generation side. If you want more realistic results, we could:

- Set **branch flow limits** on the 20 lines (e.g. based on typical IEEE-14 ratings)
- Set **min/max power limits** on the generators (their cost curves already hint at realistic ranges, e.g. g1 up to ~330 MW, g2 up to ~132 MW, etc.)

Want me to fill those in?

Worked for 6s

Ask possible reasons from LLM based on pretrained knowledge in power system domain

what cases do we have

~330 MW, g2 up to ~132 MW, etc.)

Want me to fill those in?

right, can you set the branch limits for me to make them reasonable with some congestions

[L]et me first check the current branch flows [f]rom the solution to set limits that would [r]ealistically cause some congestion.

[G]ood — now I can see the actual peak [f]lows. I'll set limits that are tight enough [o]n the busier lines to cause congestion, [w]hile keeping lighter lines comfortably [w]ithin limits.

All 20 branch limits are set. Here's what I did:

Strategy: compared each branch's previous peak flow and set limits accordingly:

Lines	Limit	Previous Peak	Intent
l1	130 MVA	165.8 MW	⚠️ Congested (~79%)
l2	60 MVA	74.3 MW	⚠️ Congested (~81%)

Worked for 36s

Then let it make some actions

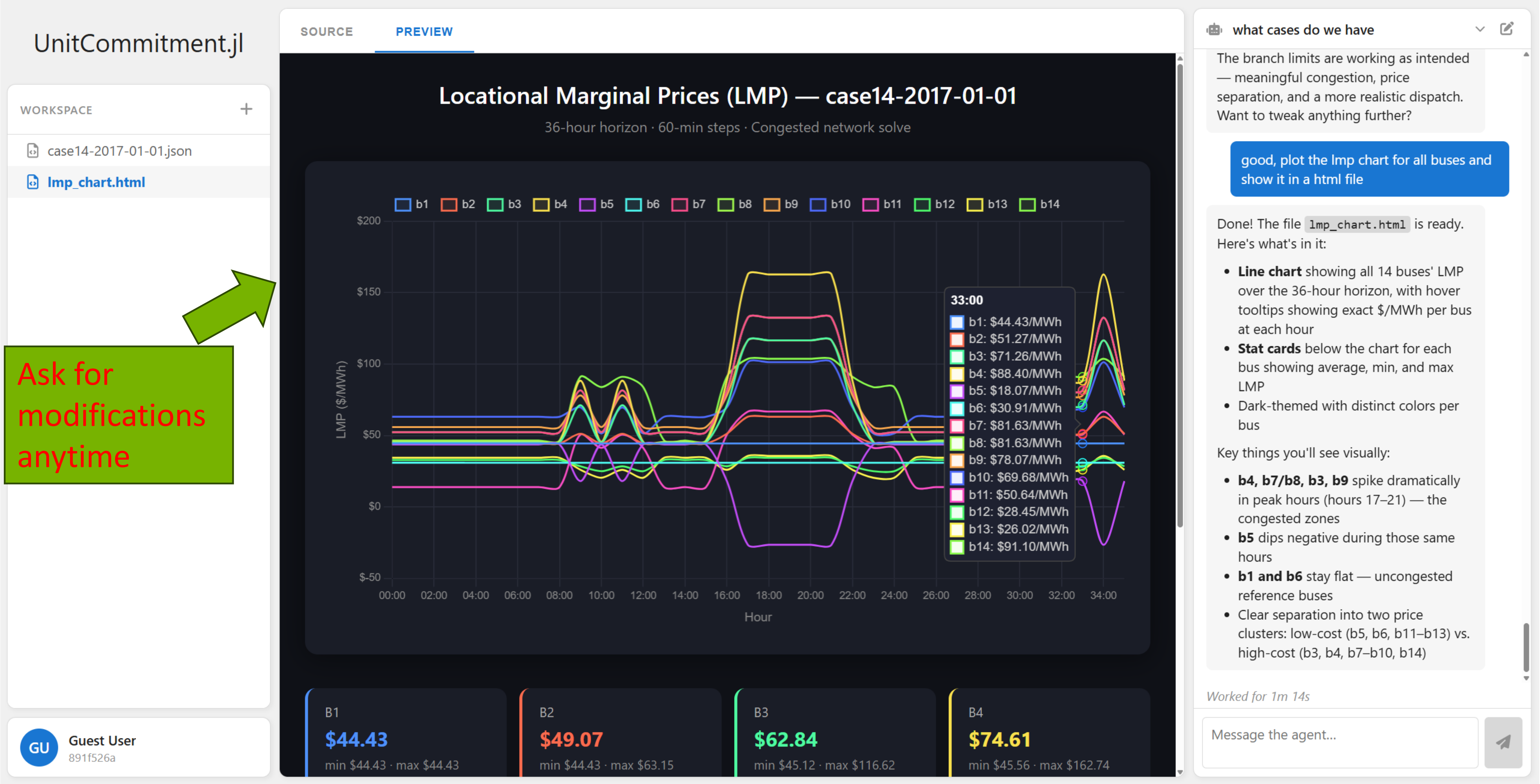
what cases do we have

	MVA	MW	Congested (~79%)
l2	60 MVA	74.3 MW	⚠️ Congested (~81%)
l3	40 MVA	47.2 MW	⚠️ Congested (~85%)
l4	43 MVA	51.3 MW	⚠️ Congested (~84%)
	33 MVA	38.9 MW	⚠️ Congested (~85%)
	45 MVA	54.1 MW	⚠️ Congested (~83%)
l10	42 MVA	49.7 MW	⚠️ Congested (~85%)
l6, l8–l9, l11–l20	10–40 MVA	well below	✅ Comfortable

Six of the busiest lines are deliberately capped below their previous peak flows, which should force the solver to re-dispatch and create meaningful congestion and non-uniform LMPs. Want me to re-solve now?

Worked for 36s

Customized rich visualization



Ask for modifications anytime

U.S. DEPARTMENT of ENERGY
Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.
Argonne NATIONAL LABORATORY

Closed loop – planning proposals

what cases do we have

- Clear separation into two price clusters: low-cost (b5, b6, b11–b13) vs. high-cost (b3, b4, b7–b10, b14)

ok, now, make some proposals to relief the congestions

Let me look at the congested branches in detail to make informed proposals.

Here's a thorough analysis of what's happening and concrete proposals to relieve the congestion:

🔍 Root Cause Analysis

The network has **two distinct congestion clusters**:

Cluster A — b1 export corridor (l1, l2, l4, l5, l7): g1 at b1 is the cheapest and most powerful unit (~330 MW curve), so the solver wants to push as much power from b1 outward as possible. Lines l1→b2, l2→b1→b5, l4→b4, and l7 (b4→b5) are all saturated trying to deliver that energy eastward.

Cluster B — b6 distribution lines (l11, l13, l18, l20): g4 sits at b6, which serves a cluster of lightly-loaded buses (b11–b14) through very thin lines that are hitting 94–100% utilization.

Worked for 40s

what cases do we have

100% utilization.

💡 Proposals

1. 🔌 Activate g2 at b2 *(Redispatch — easiest, no infrastructure change)*
   - **g2 sits idle at b2**, right in the middle of the congested corridor
   - Committing g2 (up to ~132 MW) would reduce how much power needs to flow over l1 (b1→b2) and relieve l4, l5, l7 downstream
   - Trade-off: g2 is slightly more expensive than g1, but avoids congestion penalties and high LMPs at b3/b4/b7–b10

2. ⚡ Activate g5 at b8 *(Redispatch — targets Cluster A downstream)*
   - **g5 sits idle at b8**, deep in the high-LMP zone (b7/b8 avg $72/MWh)
   - Local generation at b8 would reduce imports flowing over the congested b4→b5→b7 path
   - Especially effective during peak hours (17–21) when l7 hits 100%

3. 🏗 Upgrade l7 (b4→b5) capacity *(Infrastructure — most congested line at 100%)*
   - l7 is the single most congested line (100% utilization), acting as a bottleneck between the b1-side and the b5/b7/b8/b9 load cluster
   - Raising its limit from 45 → 70–80 MVA

Worked for 40s

Many possible mitigation proposals based on system condition

what cases do we have

- b3 (peak 97.8 MW) and b4 (peak 49.8 MW) are the heaviest loaded congested buses
- Flattening their load profile by 10–15% during hours 17–21 would reduce peak congestion without any infrastructure investment

Recommended Priority

Proposal	Type	Effort	Impac
Commit g2 @ b2	Redispatch	⭐ Low	High (Clust A)
Commit g5 @ b8	Redispatch	⭐ Low	Mediu (Clust A)
Upgrade l7 (45→75 MVA)	Infrastructure	⭐⭐ Medium	High (Clust A)
Upgrade l13+l20	Infrastructure	⭐⭐ Medium	High (Clust B)
Demand response b3/b4	Operational	⭐⭐⭐ Hard	Mediu

Want me to implement any of these (e.g. force g2/g5 on, or upgrade the line limits) and re-solve?

Worked for 40s

Closed loop – do simulations

what cases do we have

Want me to implement any of these (e.g. force g2/g5 on, or upgrade the line limits) and re-solve?

can you implement all them in parallel and then compare the solutions

I'll set up all 5 variants in parallel, solve them all, then compare. Let me first create the modified case files.

All 5 case files ready. Now submit them all in parallel.

All 5 solves submitted in parallel! Here are the job IDs:

Variant	Description	Job ID
v1	Commit g2 @ b2	`1cdfe548769291`
v2	Commit g5 @ b8	`d47d308b2a4bd5`
v3	Upgrade l1 + l7 (Cluster A)	`b6e44826f0facb`
v4	Upgrade l11/l13/l18/l20 (Cluster B)	`58b323c083b558`
v5	Demand response b3 & b4	`f5359d8c0ee2ba`

Working for 27s

Setup the scenarios

what cases do we have

them all at once when they're done.

The solve (job f5359d8c0ee2bab9) has finished. Read it with jobStatus('f5359d8c0ee2bab9') and summarize the result.

Let me check all 5 jobs at once — some may already be done too.

All 5 completed. Here's the full comparison:

📊 Summary Comparison

Metric	Baseline	v1: g2 ON	v2: g5 ON
Objective ($)	$374,923	$377,731	$389,17
Prod cost ($)	$364,415	$355,840	$370,71
Startup cost ($)	$10,508	$21,891	$18,458
Load expense ($)	$538,982	$459,359	$452,81
Avg LMP ($/MWh)	$60.89	$51.89	$51.15
Peak LMP ($/MWh)	$162.74	$251.90 ❌	$91.10

Worked for 44s

Run the simulations in parallel

Summarize the findings

what cases do we have

- Good targeted fix but doesn't address the dominant bottleneck

✅ **v5: Demand Response b3+b4 — Cheapest to implement**

- **Lowest objective ($366,947)** — saves ~$8k vs baseline
- Load expense drops ($538,982 → $482,871) reflecting reduced consumption
- But congestion pattern is **unchanged** — same 4 branches at ≥95%, same peak LMP of $162.74
- Just less energy flowing through the same bottlenecks, not actually resolving them

💡 **Recommendation**

The best real-world approach is to **combine v3 + v4**:

- **Upgrade l1+l7** to relieve Cluster A (the dominant bottleneck)
- **Upgrade l11/l13/l18/l20** to relieve Cluster B (nearly free win — Cluster B becomes uncongested with just 1 congested branch)
- Together they'd likely reduce peak LMP below $80 and bring congested branches to near zero

Want me to implement the combined v3+v4 and solve it?

Worked for 44s

Closed loop – make reports

UnitCommitment.jl

WORKSPACE +

- case14-2017-01-01.json
- case14-v1-g2on.json
- case14-v2-g5on.json
- case14-v3-upgrade-clusterA.json
- case14-v4-upgrade-clusterB.json
- case14-v5-demand-response.json
- **congestion_relief_report.md**
- lmp_chart.html

SOURCE PREVIEW

Congestion Relief Analysis Report

Case: case14-2017-01-01
Date: 2017-01-01
Horizon: 36 hours · 60-minute time steps
Network: 14 buses · 20 branches · 5 thermal generators

AI Agent writes down everything it did

Human review

1. Executive Summary

This report analyses five congestion-relief proposals applied to a security-constrained unit commitment (SCUC) solve of the IEEE 14-bus test case. The baseline scenario, after load adjustments and branch flow limits were introduced, exhibited significant network congestion across two distinct clusters, with locational marginal prices (LMPs) ranging from **−$26.49 to $162.74 /MWh** — a spread of nearly $190/MWh — and four branches operating at or above 95% utilisation

what cases do we have

Want me to implement the combined v3+v4 and solve it?

ok, can you please write me down a comprehensive report in a md file

The report has been written to `congestion_relief_report.md` (446 lines). Here's what it covers:

Report Structure

1. **Executive Summary** — key findings and top-line recommendations at a glance
2. **Network & Load Overview** — bus loads, generator fleet, branch limits table
3. **Baseline Results** — full stats,

5. Head-to-Head Comparison

5.1 Cost & LMP Summary

Variant	Objective ($)	Prod Cost ($)	Startup ($)	Load Expense ($)	Avg LMP	Peak LMP	Min LMP
Baseline	374,923	364,415	10,508	538,982	$60.89	$162.74	−$26.49
v1: g2 ON	377,731	355,840	21,891	459,359	$51.89	$251.90 ❌	−$11.11
v2: g5 ON	389,171	370,713	18,458	452,811	$51.15	$91.10 ✅	$13.88 ✅
v3: Upgrade l1+l7	**367,283**	356,775	10,508	**423,539**	**$47.85** ✅	$109.02	$13.88 ✅
v4: Upgrade ClsB	372,069	361,561	10,508	531,958	$60.09	$162.74	−$26.49
v5: Demand Resp	**366,947** ✅	356,439	10,508	482,871	$55.09	$162.74	−$26.49

8. Appendix — Solve Performance

Variant	Solve Time (s)	Termination	Optimality Gap
Baseline	7.52	OPTIMAL	0%
v1: g2 ON	5.40	OPTIMAL	0%
v2: g5 ON	9.97	OPTIMAL	0%
v3: Upgrade l1+l7	11.39	OPTIMAL	0%
v4: Upgrade ClsB	7.21	OPTIMAL	0%
v5: Demand Response	7.78	OPTIMAL	0%

All solves reached global optimality. v3 required the longest solve time (11.39 s) due to increased network flexibility creating a larger feasible space for the optimizer to explore.

Report generated from SCUC solver results — case14-2017-01-01

WHAT IT PROVIDES

THE PLATFORM

- **Agent core**
  - All the common supported features like the other agentic platforms, e.g. Claude, ChatGPT
- **Power grid modeling and simulation tools**
  - Steady state and dynamic analysis
- **Well-maintained for power grid studies**
  - System prompts, skills & MCPs
- **A shareable platform for power system communities**
  - Write, test, publish & share their customized workflows with others

KEY TAKEAWAYS & FUTURE WORK

- **Takeaways**
  - Fundamental power system analysis tools and solvers are the foundation.
  - Agentic AI can orchestrate and run the complex workflow using the tools. People stay in charge.
  - Every step in the work is logged, reproducible and audited.
- **Future work**
  - **Modeling capabilities:** Only scenario setup & run supported, add more low-level mathematic modeling capabilities for agents.
  - **More tool integrations:** Add more power grid tools for more power grid analysis, e.g., Argonne's PowerSAS dynamic modeling and simulation tool.
  - **Testing:** We still need a fair test of agent vs. expert.

ACKNOWLEDGEMENT

U.S. DEPARTMENT OF ENERGY
OFFICE OF ELECTRICITY
ADVANCED GRID MODELING PROGRAM

THANK YOU

WEI GAO · ALINSON SANTOS XAVIER · FENG QIU

ARGONNE NATIONAL LABORATORY

Scan for collaborations

Public release coming soon.