AI and Optimization: Synergies and Frontiers

Guest Lecture, NTNU Trondheim

Reda El Makroum

Energy Economics Group (EEG), Technische Universität Wien

elmakroum@eeg.tuwien.ac.at redaelmakroum.dev

About Me

Reda El Makroum

PhD Researcher, Energy Economics Group (EEG), Technische Universität Wien, Austria

Background

- MSc in Sustainable Energy Management, Al Akhawayn University, Morocco
- Started PhD in 2024 at TU Wien, Austria.

Research focus

- Agentic AI and LLMs for autonomous energy management
- Demand-side flexibility of energy systems
- Load scheduling optimization of decentralized energy assets

Agenda

1. Where optimization hits its limits
2. Three ways AI meets optimization
3. AI inside the solver
4. AI as the modeler
5. AI as coordinator
6. Applied case studies
7. OptiMUS demo
8. The road ahead

Optimization Works. Where Does It Stop?

MILP, decomposition, and commercial solvers are the backbone of operational research:

- **Energy system planning**: PyPSA, PRIMES, MESSAGEix[1,2,3]
- **Logistics**: airline scheduling, supply chains, fleet routing
- **Industrial operations**: workforce scheduling, production planning

These tools deliver **provable guarantees** and global optimality for well-defined problems.

But what happens when the problem is not well-defined?

[1] Brown et al., PyPSA, *JORS*, 2018 [2] Capros et al., PRIMES, *Energy Strategy Reviews*, 2018 [3] Huppmann et al., MESSAGEix, *Env. Modelling & Software*, 2019

The Scale of the Problem Is Changing

~600 GW
solar PV installed in 2024 alone[1]

58 million
electric vehicles on the road[2]

25 million
heat pumps in Europe[3]

Each device adds decision variables, coupling constraints, and uncertainty.

An EV fleet of 10,000 vehicles, scheduled over 24 hours at 15-min resolution = **960,000 binary decisions per day**.

[1] IEA-PVPS Snapshot 2025
[2] IEA Global EV Outlook 2025
[3] EHPA Market Data 2024

Where Does Optimization Hit Its Limits? (1/2)

Scalability

MILP-based EV fleet scheduling scales poorly: solve times grow non-linearly with fleet size, and at thousands of vehicles with real-time re-dispatch requirements, they exceed operational deadlines.

The problems that benefit most from optimization are often too large to solve within the required time window.

Information asymmetry

Centralized optimization requires full information from all participants: cost functions, state of charge, departure times, willingness to defer. In practice, EV owners cannot or will not share this data.

[1] Qin et al., “Multi-Agent RL in Energy Networks,” Survey, 2024

Where Does Optimization Hit Its Limits? (2/2)

Behavioral complexity

Optimization assumes rational actors. Real EV owners are not: they override schedules, change departure times, or simply unplug early. Models that ignore this "fail to account for behavioral constraints and user willingness."[1]

A technically optimal charging schedule that users reject is not a useful solution.

The formulation bottleneck

As problems grow more complex, the effort to formulate them mathematically grows faster than the effort to solve them. Adding vehicle-to-grid, battery degradation, or dynamic tariffs means re-deriving constraints, re-validating, and re-testing. Each extension is manual work.

[1] IEA, "Behavioural Interventions in Energy," 2024

So Where Does AI Come In?

These limitations are not new. What is new is that AI – specifically large language models – has reached a point where it can address some of them directly.

Not as a replacement for optimization, but in three roles:

- Making solvers **faster** (learning to prune, branch, warm-start)
- Making formulation **easier** (natural language to MILP)
- Handling problems that **resist formulation** (distributed, behavioral, human-facing)

Before we look at each, a quick look at what makes this feasible now.

AI Costs Are Dropping Fast

1,000×
inference cost drop in 3 years[1]

142×
fewer parameters for same performance[2]

$0.15/M
GPT-4o Mini input tokens today[3]

- GPT-3 equivalent quality: from **$60/M tokens** (2021) to **$0.06/M tokens** (2024)[1]
- Gartner: <5% of enterprise apps use task-specific AI agents today; projected **40% by end of 2026**[4]

[1] Appenzeller, "LLMflation," Andreessen Horowitz, Nov. 2024
[2] Stanford HAI, AI Index Report, 2025
[3] OpenAI GPT-4o Mini pricing, 2025
[4] Gartner Press Release, "40% of Enterprise Apps Will Feature Task-Specific AI Agents by 2026," Aug. 2025

Three Ways AI Meets Optimization

Enhance

AI *inside* the solver

ML-guided branching, warm starts, variable pruning

Interface

AI *as the modeler*

Natural language to MILP formulation and solver code

Coordinate

AI *as decision-maker*

Autonomous agents replacing the optimization loop

These are not competing approaches. They sit on a **spectrum**, and the right choice depends on the problem.

AI Inside the Solver

Hybrid approaches that preserve mathematical guarantees

- **Apollo-MILP** (ICLR 2025): Neural network predicts which integer variables to fix early, narrowing the search space for Gurobi. Reduces the solution gap by **over 50%** and finds better solutions in $\sim$17 min than Gurobi alone in 1 h.[1]
- **ML-augmented Branch & Bound**: ML learns which parts of the search tree to skip. Applied to aircraft routing, this pruned up to **49.2% of variables** without losing optimality.[2]
- **Deep Learning Warm Starts on the ISS** (Stanford, 2025): Neural network predicts initial solutions for trajectory optimization on the Astrobee robot aboard the ISS. **Up to 60% fewer solver iterations** in flight tests.[3]

The solver and its mathematical guarantees remain intact; AI just reduces the time needed to get there.

[1] Liu et al., Apollo-MILP, ICLR 2025
[2] Xu et al., "DL Approach to Accelerate MILP," *Aerospace*, 2025
[3] Banerjee et al., arXiv:2505.05588, 2025

AI as the Modeler

From natural language to optimization, automatically

- **OptiMUS** (Stanford, 2024): Takes a natural language problem description, formulates the MILP, generates solver code, executes it, and debugs iteratively. **88.6% accuracy** on LP problems, nearly 2× the baseline.[1]
- **ORLM** (*Operations Research*, 2025): Fine-tuned 7–8B-parameter open-source model achieves state-of-the-art OR benchmark performance, deployable on a single GPU.[2]
- **OR-LLM-Agent** (2025): Multi-agent framework using reasoning LLMs achieves ∼**83% accuracy** on OR benchmarks, outperforming o3 and Gemini 2.5 Pro by at least 7%.[3]

The bottleneck in OR is often not the solver, but the modeler. LLMs are closing that gap.

[1] AhmadiTeshnizi et al., OptiMUS, arXiv:2407.19633
[2] Huang et al., ORLM, *Operations Research*, 2025
[3] Zhang et al., arXiv:2503.10009

AI as Autonomous Coordinator

Skipping the formulation entirely

OPRO (DeepMind, ICLR 2024): LLMs used directly as black-box optimizers. No gradients, no solver. The task is described in natural language, and the LLM iteratively proposes and improves solutions.[1]

More broadly: **agentic AI systems**

- Autonomous decision-making in dynamic, uncertain environments
- Interpret goals in natural language, assess context, execute multi-step tasks
- Orchestrate specialized sub-agents for parallel execution[2]
- **10–30× cost-efficiency** through composition of distributed expert agents[3]

Instead of formulating the problem mathematically, the user describes the goal and the system coordinates execution.

[1] Yang et al., OPRO, ICLR 2024 [2] Hosseini & Seilani, *Array*, 2025 [3] Belcak et al., arXiv:2506.02153, 2025

Comparing the Three Approaches

	MILP	AI Inside the Solver	AI as Coordinator
Solution quality	Provably optimal	Near-optimal	Near-optimal
Computation time	Seconds to hours	Accelerated	Seconds to minutes
User interface	Technical expertise	Mixed	Natural language
Scalability	Grows with problem size	Improved	Linear (post-training)

Each approach has strengths. The question is not **which is better** but **which fits your problem**.

Well-defined, safety-critical, complete information $\rightarrow$ optimization.
Distributed, uncertain, human-facing $\rightarrow$ AI coordination may be the better fit.

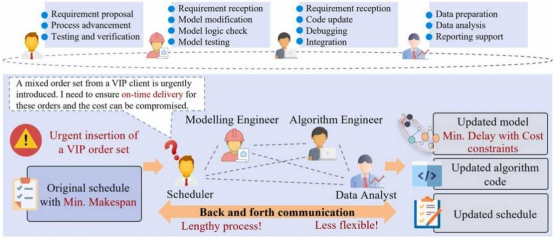
Applied Case Studies (1/2): Smart Manufacturing Scheduling

A4PS (Li et al., *J. Manuf. Syst.*, 2026)[1]

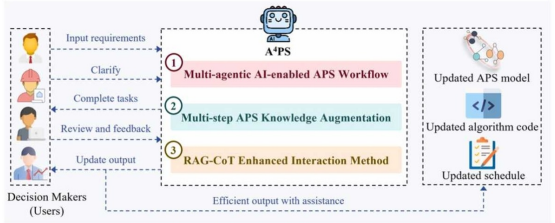
- VIP rush order arrives $\rightarrow$ scheduling objective changes $\rightarrow$ model, constraints, and solver code need rewriting
- Traditional process: weeks of coordination between supervisors, modellers, algorithm developers
- A4PS: **8 LLM agents** follow standard operating procedures, each handling one step (parse request $\rightarrow$ build model $\rightarrow$ write code $\rightarrow$ diagnose)
- Complex cases: modelling success $\sim$**50%** $\rightarrow$ **75.6%**, code executability $\sim$**57%** $\rightarrow$ **90%**

[1] Li et al., *J. Manuf. Syst.*, 85, 207–226, 2026

Applied Case Studies (1/2): A4PS Workflow



(a) Traditional workflow for APS structural updates



(b) New paradigm for APS structural updates via A⁴PS

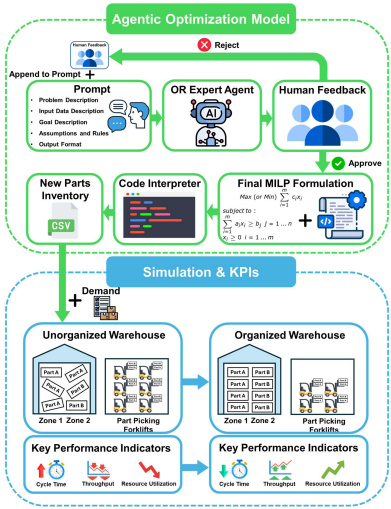
Applied Case Studies (2/2): Warehouse Slotting Optimization

Agentic Optimization Framework (El Baz et al., *Comput. Ind. Eng.*, 2026)[1]

- Real forklift assembly warehouse: 17 one-way aisles, $\sim$2,400 SKUs, 34 pick types
- LLM agent (GPT-4o) generates MILP formulations $\rightarrow$ human validates $\rightarrow$ FlexSim simulation evaluates
- $\sim$10 agentic iterations to reach a stable, constraint-satisfying formulation
- Blocking $-$**33.7%**, throughput $+$**7.8%** vs. baseline

[1] El Baz et al., *Comput. Ind. Eng.*, 214, 111884, 2026

Applied Case Studies (2/2): Agentic Optimization Framework



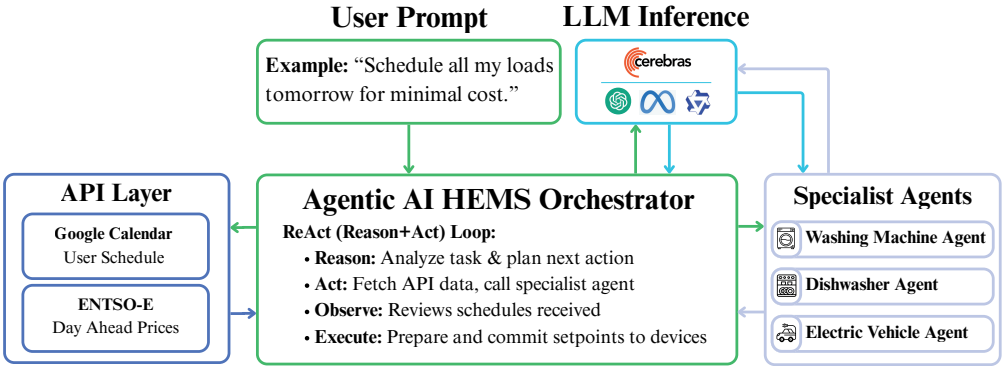
Agentic AI for Home Energy Management

- Same pattern, applied to energy: user says *“charge my EV to 80%, minimize cost”* and a multi-agent LLM system handles price retrieval, constraint checking, and scheduling
- Orchestrator + specialized sub-agents per appliance, using the ReAct (Reason + Act) loop
- Benchmarked against MILP-optimal schedules across 75 trials with real Austrian electricity prices
- Llama-3.3-70B achieves **100% optimality** across all single- and multi-appliance scenarios

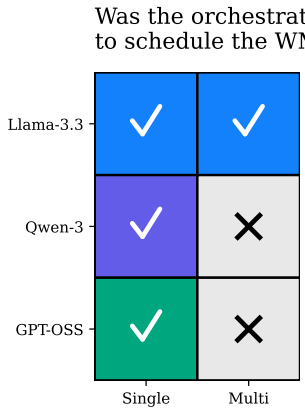
Open source: github.com/RedaElMakroum/agentic-ai-hems Paper: arXiv:2510.26603

El Makroum et al., “Agentic AI HEMS: An LLM Framework for Residential Load Scheduling,” arXiv:2510.26603, 2025

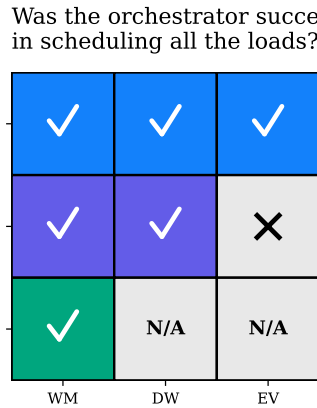
Agentic AI HEMS: System Architecture



How does the Agentic System Perform?



Single-appliance: all models achieve **100% optimality** with similar computational requirements.



Multi-appliance: only Llama-3.3-70B maintains **100% optimality**. Other models fail to coordinate all three appliances.

The Road Ahead

We saw three modes (enhance, interface, coordinate) and three applied cases (manufacturing, warehousing, energy). Where is this going:

- **Not replacement, integration**: Solvers keep their optimality guarantees. AI fills the gaps around them, from translating user intent to writing formulations to orchestrating multi-step workflows.
- **The modelling bottleneck shrinks**: A4PS showed that domain experts can update scheduling models through natural language. The barrier to using optimization is no longer the solver, it is the modeller.
- **Domain-agnostic frameworks, domain-specific knowledge**: The agentic architecture transfers across manufacturing, warehousing, and energy. What changes is the knowledge base.

Key Takeaways

1. **AI and optimization are converging.** Three modes – enhance, interface, coordinate – each suited to different problems. The right answer is often a hybrid.
2. **The bottleneck is shifting from solving to formulating.** LLMs can translate natural language to mathematical programs at high accuracy, making OR tools more accessible.
3. **The same agentic pattern works across domains.** Manufacturing scheduling, warehouse slotting, energy management. The framework transfers; the domain knowledge is what changes.

OptiMUS Demo

Natural language to MILP, automatically

"A factory produces two products (A and B). Product A yields €40 profit, product B yields €30. Each unit of A requires 2 hours of machining and 1 hour of assembly. Each unit of B requires 1 hour of machining and 2 hours of assembly. The factory has 100 hours of machining and 80 hours of assembly available per week. Maximize weekly profit."

Feel free to tag along:

optimus-solver.com

Thank You

Reda El Makroum

Energy Economics Group (EEG), Technische Universität Wien

Email: elmakroum@eeg.tuwien.ac.at

redaelmakroum.dev

github.com/RedaElMakroum/agentic-ai-hems

TU WIEN TECHNISCHE UNIVERSITÄT WIEN

EEG Energy Economics Group