Operations Research Meets AI: From Foundations to New Frontiers

Wenyuan Tang

Department of Electrical and Computer Engineering

NC State University

2026 FREEDM Annual Research Symposium

February 10, 2026

Operations Research (OR) in Power Systems

Source: AI-generated

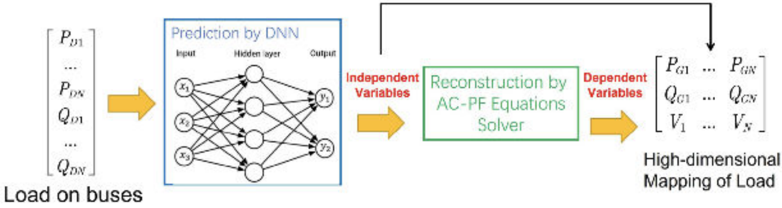
- OR = analytics + modeling + simulation + optimization to support decision-making

AI in Power Systems

Source: AI-generated

- AI enables smarter forecasting, optimization, and control in power systems

Synergy Between OR and AI

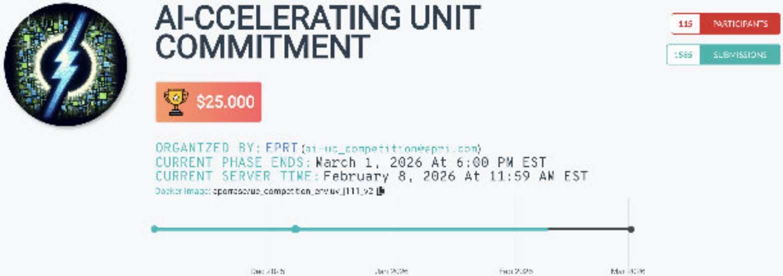


Source: DeepOPF, Pan et al. (2023)

- As the core of OR, optimization serves as the computational foundation of AI
- AI unlocks new frontiers in optimization and the broader field of OR

Today, I'd like to share an anecdote from practice to show how OR and AI complement each other...

Unit Commitment (UC)



Source: https://www.codabench.org/competitions/11178/

- Decision variables: on/off status of generating units (binary); production of generating units, charging or discharging power of storages (continuous)
- Constraints: unit-level and system-level operational constraints (usually linear)
- Objective: minimizing total operating cost (usually linear or convex piecewise linear)
- UC is formulated as a large-scale mixed-integer linear program (MILP), which is non-convex and belongs to the class of NP-hard problems

AI-ccelerating Unit Commitment (AI-UC) Competition

Source: AI-generated based on test system used

- Traditional optimization methods are challenged by scale, speed, and uncertainty
- Even for a small test system, a time horizon of 72 hours is sufficient to pose a computational challenge for off-the-shelf solvers (about 100 seconds per instance)
- Machine learning (ML) techniques may significantly improve computational performance, aiming for near-optimal solutions in about 1 second per instance

Do AI/ML Outperform Traditional Methods?

Workflow of AI/ML methods

- 2620 instances are solved exactly and provided to participants
- Participants may optionally augment the dataset to increase their training data
- Participants train their AI/ML models (input: demand, solar, and wind over the 72-hour time horizon; output: binary UC decisions) using the available data
- The trained models are evaluated on 504 unseen instances, scored on solution quality and computational speed

- While trained AI/ML models offer near-instant inference, the primary challenge is ensuring feasibility, often necessitating corrective post-processing
- Post-processing can be complex due to the strong interdependence of variables and constraints, where over-adjustment can compromise solution optimality
- At first, I believed traditional methods, such as greedy algorithms with efficient heuristics, could outperform AI/ML, since they are fast and guarantee feasibility
- Contrary to initial expectations, my greedy algorithm yields an optimality gap of about 10%, whereas some AI/ML methods achieved higher-quality solutions

Are Traditional Methods Still Valuable?

strange solution regarding startup limit #57

Answered by iSoron tangwenyuan asked this question in Help & FAQ

tangwenyuan on Nov 26, 2025

Hi,

I found the solution is strange (not optimal) for a minimal example here.
case1.json

Essentially I'd like to fix the commitment status for the only generator so that we solve an economic dispatch problem for a horizon of 4 slots. The minimum power output is 63, the maximum is 101, the startup limit is 90, the minimum uptime is 1; the load and the other limits (ramp up, ramp down, shutdown limit, etc.) are large enough.

The following is the result (case2, case3, case4 can be obtained easily by changing the commitment status):

case1 [1, 0, 0, 0]: the dispatch is [75, 0, 0, 0], and I think the optimal one should be [90, 0, 0, 0].
case2 [1, 1, 0, 0]: the dispatch is [90, 90, 0, 0], and I think the optimal one should be [90, 101, 0, 0].
case3 [1, 1, 1, 0]: the dispatch is [90, 101, 90, 0], and I think the optimal one should be [90, 101, 101, 0].
case4 [1, 1, 1, 1]: the dispatch is [90, 101, 101, 101], which is optimal.

Source: https://github.com/ANL-CEEESA/UnitCommitment.jl/discussions/57

- While the greedy algorithm is unlikely to beat the top AI/ML methods, it surprisingly produced a valuable byproduct: it uncovered a bug in `UnitCommitment.jl` (v0.4.1), the optimization package used to generate optimal solutions as training data
- The bug was identified when `UnitCommitment.jl` incorrectly flagged a solution produced by the greedy algorithm as infeasible; by systematically narrowing down the issue, I constructed a minimal reproducible example that exposes the bug

Tracing Back the Bug

- The bug stems from an incorrect formulation of a constraint in the UC model, originating from a typo in Knueven et al. (2020), *On Mixed-Integer Programming Formulations for the Unit Commitment Problem*, INFORMS Journal on Computing
- When the minimum uptime is 1, the generator can start up at t and shut down at $t+1$, with its power output at t constrained by both the startup and shutdown limits

$$p^l(t) \leq (\overline{P}^l - \overline{P}^{l-1})u(t) - C^v(l)v(t) \tag{47a}$$

$$p^l(t) \leq (\overline{P}^l - \overline{P}^{l-1})u(t) - C^w(l)w(t+1) \tag{47b}$$

- The constraints are then tightened to strengthen the LP relaxation, thereby accelerating the branch-and-cut process by reducing the search tree size

$$p^l(t) \leq (\overline{P}^l - \overline{P}^{l-1})u(t) - C^v(l)v(t) - [C^v(l) - C^w(l)]^+ w(t+1) \tag{48a}$$

$$p^l(t) \leq (\overline{P}^l - \overline{P}^{l-1})u(t) - C^w(l)w(t+1) - [C^w(l) - C^v(l)]^+ v(t) \tag{48b}$$

- However, the signs in the red expressions are erroneously inverted
- This bug occurs in a special case and is too subtle to be noticed (now fixed in v0.4.2)

Beyond the Fundamentals

Empirical aspects of OR: tightness vs. compactness

- Tightness: the degree to which the LP relaxation recovers the convex hull
- Compactness: the efficiency of the model in terms of variable and constraint count
- A tight formulation provides a better bound at the cost of larger LP subproblems, whereas a compact formulation offers faster node evaluations at the cost of a weaker relaxation and a larger search tree
- Modern MILP solvers incorporate sophisticated heuristics and cut generation routines, and hence it is important to evaluate UC formulations empirically

AI builds on OR, opening new opportunities...

- AI can achieve significant speedups while maintaining near-optimal solutions
- AI training may rely on exact solutions produced by traditional OR methods
- Traditional OR methods provide the rigor to reveal issues that AI alone might miss
- In hindsight, AI-based proof systems could have detected the derivation error
- We look forward to deeper industry collaboration: data fuels AI in power systems