Solving Large-Scale Security-Constrained Unit Commitment Problems Using Machine Learning Approaches

Backgrounds:

The Security-Constrained Unit Commitment (SCUC) problem is a critical optimization task in power system operations. It determines the optimal schedule of power generation units while ensuring system reliability under contingencies, such as transmission line failures or generator outages. Traditional SCUC methods employ mixed-integer linear programming (MILP) models and optimization solvers to obtain solutions. Decision variables include binary variables indicating generator operational status and continuous variables representing power production levels at each time slot. Core constraints governing SCUC formulations encompass resource-level constraints and system-wide constraints. Resource-level constraints represent the physical operating characteristics of generators and other resources, including capacity limits, ramp rate limits, minimum run times, and minimum down times. These constraints may link binary and continuous variables across multiple time intervals. System-wide constraints primarily consist of power balance constraints, reserve requirement constraints, and transmission constraints. Transmission constraints are enforced to meet reliability standards under transmission line failures. They typically associate with continuous variables and are often decoupled by time intervals. State-of-the-art formulation variants of SCUC include multiple ramping formulations [1,2], multiple piecewise-linear costs formulations [3,4], and startup costs formulation [5]. Traditional methods encounter computational bottlenecks when applied to large-scale power grids with high renewable penetration (e.g., exceeding 1000 buses).

The SCUC problem has been repeatedly solved in real-world applications, often several times daily, with only incremental changes to input data. Fundamental system parameters such as generation unit characteristics and grid topology exhibit minimal variation between consecutive solutions. Recent advancements in machine learning (ML)—particularly reinforcement learning (RL), graph neural networks (GNNs), and physics-informed neural networks (PINNs)—offer promising alternatives to expedite SCUC solving. ML can learn from historical operational data, approximate complex constraints, and accelerate decision-making, making it a compelling approach for real-time power system optimization. Some existing ML-based methods are listed below:

Identify Redundant Constraints: Predict, based on statistical data, which constraints are necessary in the formulation and which constraints can be safely omitted.

Warm Start: Construct, based on a large number of previously obtained optimal solutions, a partial solution that is likely to work well as a warm start.

Fix Partial Solutions: Identify, with very high confidence, a smaller-dimensional affine subspace where the optimal solution is likely to lie, leading to the elimination of a large number of decision variables and significantly reducing the complexity of the problem.

Predicting Integer Solutions: This method predicts the binary decisions first, and then the MILP-SCUC problem is transformed into a continuous linear programming problem while fixing the integer variables.

Technical Requirements:

1. Develop a machine learning-based solution framework for the SCUC problem, where either partial machine learning components or fully end-to-end learning architectures may be employed.
2. Computational acceleration using either GPU or multi-core CPU platforms is suggested to be incorporated in the solution implementation.

Dataset:

SCUC Configurations: The open-source package UnitCommitment.jl [10] provides 42 SCUC configurations. These configurations are JSON-based and describe the most common generator characteristics, including ramping, piecewise-linear production cost curves, and time-dependent startup costs. Additionally, they encompass operating reserves, price-sensitive loads, transmission networks, and contingencies.

MIP Formulations: UnitCommitment.jl extends the default formulation with five more types of ramping formulations, three more types of piecewise-linear costs formulations, one more type of startup costs formulation, and one more type of transmission formulation. This combination yields 96 (6 * 4 * 2 * 2) types of formulations.

Dataset Generation: Utilizing the aforementioned 42 configurations and 96 formulations, 4032 MIP instances can be generated.

Baseline:

Solver: Gurobi, version 12.0.0 or above

Configuration: The solving time of obtaining a feasible solution with a 1% Gap under a 7200s time limit.

- model.Params.TimeLimit = 7200
- model.Params.MIPGap = 0.01

Hardware: Intel X86 with CPU Frequency >= 2.60GHz

Goal:

Optimality Metric: Obtain a solution within 0.5% optimality gap compared to the solution obtained by Gurobi according to the definitions provided in the baseline descriptions, i.e., $\left|\frac{solution_{proposed\_met} \quad - solution_{baseline}}{solution_{baseline}}\right| <= 0.5\%$.

Performance: the performance is benchmarked by calculating the geometric mean score (GMS) of all the 4032 instances under the proposed methods. The objective is to achieve a 10X speedup over the baseline method under the above dataset while maintaining the above optimality metric, i.e., $GMS_{proposed\_method} <= 0.1 * GMS_{baseline}$.

Hardware Acceleration: Utilize GPU or multi-core CPU acceleration whenever possible.

Deliverables:

- Source code for training and testing the machine learning model.
- Source code and the model file of the ML-based solution for evaluation.

References:

[1] Arroyo, J. M., & Conejo, A. J. (2000). Optimal response of a thermal unit to an electricity spot market. IEEE Transactions on Power Systems, 15(3), 1098-1104.

[2] B. Wang and B. F. Hobbs, “Real-Time Markets for Flexiramp: A Stochastic Unit Commitment-Based Analysis,” in IEEE Transactions on Power Systems, vol. 31, no. 2, pp. 846-860, March 2016.

[3] Carrión, M., & Arroyo, J. M. (2006). A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Transactions on Power Systems, 21(3), 1371-1378.

[4] Knueven, B., Ostrowski, J., & Watson, J. P. (2018). Exploiting identical generators in unit commitment. IEEE Transactions on Power Systems, 33(4), 4496-4507.

[5] Morales-España, G., Latorre, J. M., & Ramos, A. (2013). Tight and compact MILP formulation for the thermal unit commitment problem. IEEE Transactions on Power Systems, 28(4), 4897-4908.

[6] Ramesh A V, Li X. Machine learning assisted model reduction for security-constrained unit commitment[C]. 2022 North American Power Symposium (NAPS). IEEE, 2022: 1-6.

[7] Baker K. Learning warm-start points for AC optimal power flow[C]. 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2019: 1-6.

[8] Xavier Á S, Qiu F, Ahmed S. Learning to Solve Large-Scale Security-Constrained Unit Commitment Problems. INFORMS Journal on Computing, 2021, 33(2): 739-756.

[9] Tang X, Bai X, Weng Z, et al. Graph Convolutional Network-Based Security-Constrained Unit Commitment Leveraging Power Grid Topology in Learning. Energy Reports, 2023, 9: 3544-3552.

[10] https://github.com/ANL-CEEESA/UnitCommitment.jl