




PowNet 2.0: A production cost modeling framework for large-scale power systems

Phumthep Bunnak¹[¶], Hisham Eldardiry¹, Matija Pavičević², Jerry Zhuoer Feng¹, and Stefano Galelli¹

¹ Cornell University, USA ² Argonne National Laboratory, USA [¶] Corresponding author

DOI: [10.21105/joss.08035](https://doi.org/10.21105/joss.08035)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Arfon Smith](#)  

Reviewers:

- [@ijbd](#)
- [@praka](#)

Submitted: 26 October 2024

Published: 15 July 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

Managing the complex network of power stations and transmission lines that deliver electricity across large spatial domains (e.g., country and continental scale) requires a variety of mathematical models. Among them, Production Cost Models (PCM) are commonly used for planning the short-term operation of power grids ([Garver, 1962](#)). Specifically, PCMs schedule the production of electricity by generators considering transmission constraints with planning horizons that range from a few hours to a few days.

PowNet 2.0 is a Python-based PCM framework that simulates and analyzes the most cost-effective way of meeting electricity demand using various electricity sources (e.g., coal, natural gas, and renewables) such that sufficient electricity is generated at each hour while considering factors like fuel prices, availability of renewables, and regulatory requirements. The framework also allows us to study the integration between hydropower and power systems.

The framework is designed to be accessible to a wide range of users, especially those with only basic Python knowledge. To use PowNet 2.0, a user can supply data for power stations, transmission lines, and electricity demand as spreadsheets, and run a script to handle the complex calculations and to produce modeling outputs as either data frames or spreadsheet files. While a basic user does not need to modify the codebase, an advanced user can leverage the software's flexible and modular design for more complex tasks, like modeling the interaction between water and power systems or exploring customized optimization algorithms.

Statement of need

PCMs share similar features ([Oikonomou et al., 2022](#)) and are widely used in both industry and academia. While proprietary options like PLEXOS and PROMOD offer comprehensive features and user-friendly interfaces, they do not follow the Findable, Accessible, Interoperable, and Reusable (FAIR) principles ([Wilkinson et al., 2016](#)), which help improve the transparency and applicability of models. Furthermore, the cost of proprietary PCMs can be prohibitive for many researchers and institutions. In response to these limitations, the open-source community has developed PCMs in various programming languages. A detailed taxonomy and comparison of different PCMs can be found in ([Hoffmann et al., 2024](#); [Oberle & Elsland, 2019](#)). Notable examples include SIENNA ([Lara et al., 2021](#)), written in Julia, and Python-based frameworks like PyPSA ([Brown et al., 2018](#)), Grid Operations ([Akdemir et al., 2024](#)), and PowNet ([Chowdhury et al., 2020](#)). Building upon its predecessor, PowNet 2.0 represents a significant leap forward, particularly as not all existing PCM frameworks have a full suite of features for addressing specific needs within the power systems domain, namely (1) the flexibility to incorporate diverse mathematical formulations and solution algorithms, (2) the capacity to generate stochastic input data (e.g., load, solar availability) to support uncertainty analyses, and (3) the ability to

integrate hydropower and power system models. All these features are seamlessly integrated in our software.

PowNet 2.0 has a few key functionalities that address the aforementioned needs:

Mathematical Formulations: Selecting a set of computationally efficient mathematical formulations is an ongoing research effort (Chen et al., 2022; Tejada-Arango et al., 2019). The set of formulations implemented in PowNet 2.0 was chosen based on thorough benchmarking exercises that compare the runtime of several different formulations (Hörsch et al., 2018; Knueven et al., 2020). PowNet 2.0 also allows a user to experiment with different formulations, such as representing the direct-current power flow with either the voltage-angle formulation or the Kirchhoff formulation.

Solution Method: PowNet 2.0 supports both Gurobi and HiGHS as the mathematical solver. While Gurobi is a powerful commercial solver, its free usage is limited to an academic license. Consequently, HiGHS was chosen as an alternative due to its competitive performance among open-source solvers (Parzen et al., 2022). A user can also explore solution methods like rounding heuristics (Bunnak et al., 2025).

Stochastic generation of input variables: A user can automatically generate stochastic time series of load, solar, wind, and hydropower availability. These time series are created by the SARIMAX model, which requires as input deterministic time series. Through this functionality, the user can then explore the impact of uncertainty in weather and climatic conditions on power system performance.

Reservoir simulation: The reservoir module simulates the operation of hydropower reservoirs, thus providing time series of available hydropower. It requires dam design specifications and inflow as input data. This feature makes the model more detailed, such as avoiding the use of static capacity factors, while reducing reliance on external hydrologic models.

Water-power system coupling: The reservoir simulation module can be either soft- or hard-coupled with the 'core' module simulating the unit commitment and economic dispatch problem. In the former case, the hydropower availability is passed as input (unidirectional information flow). In the latter case, the two modules are run in parallel, ensuring that dam release decisions follow the exact needs of the entire power system – representing a tighter integration between water and power systems (Koh et al., 2022).

Software Design and Implementation

PowNet 2.0 offers an end-to-end modeling framework through functionalities from generating synthetic time series of external forcings (e.g., load, solar, and wind availabilities) to plotting the modeling outputs. PowNet 2.0 leverages the Gurobipy package for building instances of the optimization problem. This package was chosen because of its proven performance when compared to other Python-based modeling frameworks as shown in (Broihan, 2023; Hofmann, 2023). Furthermore, Gurobipy allows a user to leverage the Python ecosystem by using popular libraries like Pandas, SciPy, and NetworkX.

The modeling workflow of PowNet 2.0 is illustrated in Figure 1. For each modeling task represented by a rectangle, there is a corresponding module to accomplish the task. This modular design facilitates unit testing of individual class objects and functions, ensuring code reliability and maintainability. It also enables future extension and customization, empowering users to adapt PowNet 2.0 to their specific needs and contribute to its ongoing development. Current development efforts focus on modeling power purchase agreements, regional electricity trading schemes, and computational algorithms.

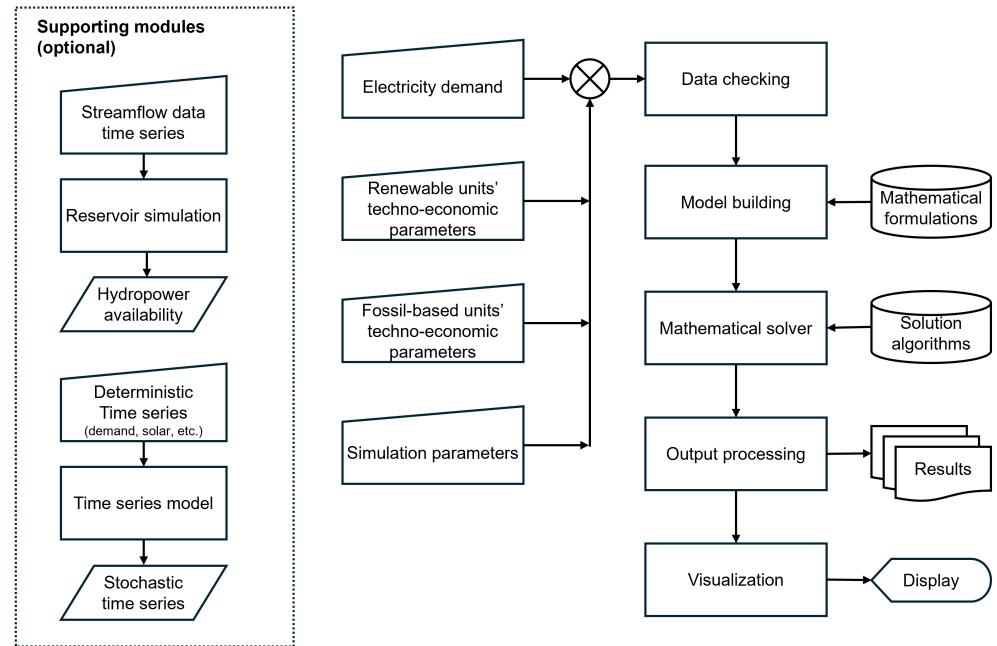


Figure 1: A simplified diagram of PowNet 2.0 modeling workflow. Each process is responsible by a dedicated Python module.

Acknowledgement

Development of the reservoir module was supported by the US Department of Energy (DOE) Water Power Technologies Office as a part of the HydroWIRES “HydroCHiPPs” project (53165).

References

- Akdemir, K. Z., Oikonomou, K., Kern, J. D., Voisin, N., Ssembatya, H., & Qian, J. (2024). An open-source framework for balancing computational speed and fidelity in production cost models. *Environmental Research: Energy*, 1(1), 015003. <https://doi.org/10.1088/2753-3751/ad1751>
- Broihan, J. (2023). *Performance in optimization models: A comparative analysis of GAMS, pyomo, GurobiPy, and JuMP*. <https://www.gams.com/blog/2023/07/performance-in-optimization-models-a-comparative-analysis-of-gams-pyomo-gurobi-py-and-jump/>
- Brown, T., Hörsch, J., & Schlachtberger, D. (2018). PyPSA: Python for power system analysis. *Journal of Open Research Software*, 6(4, 1). <https://doi.org/10.5334/jors.188>
- Bunnak, P., Coniglio, S., & Galelli, S. (2025). Bridging theory and practice: Efficiently solving the unit commitment problem in production cost models. In *Energy* (No. 5; Vol. 322, p. 135454). Elsevier. <https://doi.org/10.1016/j.energy.2025.135454>
- Chen, Y., Pan, F., Qiu, F., Xavier, A. S., Zheng, T., Marwali, M., Knueven, B., Guan, Y., Luh, P. B., Wu, L., & others. (2022). Security-constrained unit commitment for electricity market: Modeling, solution methods, and future challenges. *IEEE Transactions on Power Systems*, 38(5), 4668–4681. <https://doi.org/10.1109/TPWRS.2022.3213001>
- Chowdhury, A. K., Kern, J., Dang, T. D., & Galelli, S. (2020). PowNet: A network-constrained

- unit commitment/economic dispatch model for large-scale power systems analysis. *Journal of Open Research Software*, 8(1).
- Garver, L. L. (1962). Power generation scheduling by integer programming-development of theory. *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, 81(3), 730–734. <https://doi.org/10.1109/aieepas.1962.4501405>
- Hoffmann, M., Schyska, B. U., Bartels, J., Pelser, T., Behrens, J., Wetzels, M., Gils, H. C., Tang, C.-F., Tillmanns, M., Stock, J., & others. (2024). A review of mixed-integer linear formulations for framework-based energy system models. *Advances in Applied Energy*, 100190. <https://doi.org/10.1016/j.adapen.2024.100190>
- Hofmann, F. (2023). Linopy: Linear optimization with n-dimensional labeled variables. *Journal of Open Source Software*, 8(84), 4823. <https://doi.org/10.21105/joss.04823>
- Hörsch, J., Ronellenfisch, H., Witthaut, D., & Brown, T. (2018). Linear optimal power flow using cycle flows. *Electric Power Systems Research*, 158, 126–135. <https://doi.org/10.1016/j.epsr.2017.12.034>
- Knueven, B., Ostrowski, J., & Watson, J.-P. (2020). On mixed-integer programming formulations for the unit commitment problem. *INFORMS Journal on Computing*, 32(4), 857–876. <https://doi.org/10.1287/ijoc.2019.0944>
- Koh, R., Kern, J., & Galelli, S. (2022). Hard-coupling water and power system models increases the complementarity of renewable energy sources. *Applied Energy*, 321, 119386. <https://doi.org/10.1016/j.apenergy.2022.119386>
- Lara, J. D., Barrows, C., Thom, D., Krishnamurthy, D., & Callaway, D. (2021). PowerSystems.Jl—a power system data management package for large scale modeling. *SoftwareX*, 15, 100747. <https://doi.org/10.1016/j.softx.2021.100747>
- Oberle, S., & Elstrand, R. (2019). Are open access models able to assess today's energy scenarios? *Energy Strategy Reviews*, 26, 100396. <https://doi.org/10.1016/j.esr.2019.100396>
- Oikonomou, K., Tarroja, B., Kern, J., & Voisin, N. (2022). Core process representation in power system operational models: Gaps, challenges, and opportunities for multisector dynamics research. *Energy*, 238, 122049. <https://doi.org/10.1016/j.energy.2021.122049>
- Parzen, M., Hall, J., Jenkins, J., & Brown, T. (2022). Optimization solvers: The missing link for a fully open-source energy system modelling ecosystem. *Mars*. <https://doi.org/10.5281/zenodo.12703174>
- Tejada-Arango, D. A., Lumbreras, S., Sánchez-Martín, P., & Ramos, A. (2019). Which unit-commitment formulation is best? A comparison framework. *IEEE Transactions on Power Systems*, 35(4), 2926–2936. <https://doi.org/10.1109/TPWRS.2019.2962024>
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Silva Santos, L. B. da, Bourne, P. E., & others. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1), 1–9. <https://doi.org/10.1038/sdata.2016.18>