

CapacityExpansion: A capacity expansion modeling framework in Julia

Lucas Elias Kuepper¹, Holger Teichgraeber¹, and Adam R. Brandt¹

¹ Department of Energy Resources Engineering, Stanford University

DOI: [10.21105/joss.02034](https://doi.org/10.21105/joss.02034)

Software

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

Editor: [Kathryn Huff](#) ↗

Reviewers:

- [@sjpfenninger](#)
- [@ali-ramadhan](#)

Submitted: 14 November 2019

Published: 05 February 2020

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

CapacityExpansion is a Julia (Bezanson, Edelman, Karpinski, & Shah, 2017) implementation of an input-data-scaling capacity expansion modeling framework. It provides an extensible, multi-carrier, simple-to-use generation and transmission capacity expansion model that allows to address a diverse set of research questions in the area of energy systems planning and can be used to plan and validate energy systems at scales ranging from quarters to entire global regions. CapacityExpansion provides simple integration of (clustered) time-series, geographical, cost, and technology input data. The software features a modular model setup and an investment and dispatch optimization that uses the JUMP modeling language (Dunning, Huchette, & Lubin, 2017). An interface is provided between the optimization result and further analysis.

Infrastructure planning in the energy sector

Energy systems convert different energy resources to meet desired demands like electric and thermal energy demands. Political, economical, and technological changes require expansions of the infrastructure of energy systems. Expanding the infrastructure has to balance multiple political, environmental, and economical objectives. Capacity expansion planning can be an important tool during the planning process (Gacitua et al., 2018).

Capacity expansion planning is used to compute cost-optimal energy system designs under given sets of constraints from the perspective of a central planner. The resulting cost-optimal energy system design can be used to inform policy decisions that incentivize the industry to invest in this design (Johnston, Mileva, Nelson, & Kammen, 2013). Similarly, cost-optimal energy system designs can be used by companies for their investment strategies.

Aspects of the energy system design that capacity expansion planning aims to answer are what the optimal technology mix is in regards of location, time, and installed generation, conversion, storage, and transmission capacities. The design optimization is done while using an integrated dispatch formulation to ensure that supply can equal demand at all nodes and time steps. The model determines the costs, emissions, power generation, energy storage, and power flows based on the installed capacities.

Capacity expansion planning is formulated as a mathematical optimization problem. Like any optimization problem, capacity expansion planning has certain degrees of freedom, consists of constraints, and consists of an objective function that is minimized: Typical degrees of freedom (also called decision variables) are installed capacities, power generation, energy storage, and power flows. Some constraints ensure that the model is physically consistent in itself and following the rules of thermodynamics, e.g. energy balances ensure energy conservation over time. Other constraints restrict the solution space to external conditions like costs, demands,

available energy resources as well as other political, environmental, economical, or technological constraints. The objective function to determine cost-optimal investment can be total system cost, the net present value, or another business-oriented cost measure.

Dispatch planning formulations are very similar to capacity expansion formulations. However, the installed capacity is no degree of freedom, but introduced as an external constraint.

Package features

The model class of `CapacityExpansion` is capacity expansion planning and it combines generation and transmission capacity expansion planning. The model is setup as a linear optimization model that models energy systems based on the provided input data. Multiple energy carriers can be modeled, which makes the software suitable for research of sector coupling technologies. Technologies can be defined that belong to dispatchable or non-dispatchable generation, conversion, storage, transmission, or demand. The decision variables of the model are investment and dispatch and the total system costs are minimized.

The following key features are provided by `CapacityExpansion`. The usage and mathematical formulation is explained in detail within the software's documentation.

- *Modeling language:* The implementation in Julia enables the usage of a high-level programming language, and short times for data handling at the same time. The implementation in Julia also allows the usage of the mathematical optimization language JUMP, which enables the optimization formulation close to mathematical writing (Dunning et al., 2017). The integration into the Julia ecosystem further allows to handle the process from data preparation to figure export within one programming language and software.
- *Provided input data:* The input data for single and multi-node representations of electricity systems are provided for California (USA) and Germany (EU). The input data provides hourly time-series input data for multiple years, aggregated geographic information about existing generation and transmission, cost information including the monetary and life cycle assessment costs, and necessary technology data. This input data can be used to address many research questions.
- *The generalized import of input data:* Modeling other energy systems is possible by adjusting the input data. The package extracts all information needed to model a specific energy system based on a few standardized input files. The time-series, geographic, and cost data can be edited as tables and integrated using the `.csv`-file format. The technology input data can be edited like a tree structure and integrated using the `.yaml`-file format.
- *Integration of TimeSeriesClustering:* Time-series data like the demand, available solar factors, and available wind factors are used as an input to model the temporal variance of the energy system. Aggregating the time-series input data is commonly done to reduce the computational complexity of the optimization. `CapacityExpansion` is well integrated with the Julia package `TimeSeriesClustering` (Teichgraeber et al., 2019). This integration allows using the typical time-series aggregation methods (Teichgraeber & Brandt, 2019), integrated testing of the temporal resolution, and an integrated feedback loops between the optimization result and time-series aggregation.
- *Seasonal storage:* A recent seasonal storage formulation from Kotzur is implemented to allow time-series aggregation and modeling of seasonal storage at the same time (Kotzur, Markewitz, Robinius, & Stolten, 2018).

- *Modular model setup:* The model setup is flexible and based on a modular setup. Depending on the configuration different technology groups can be activated or deactivated, a green or brown field study performed, emissions limits can be enforced, different storage models can be used, and an integrated investment and dispatch or pure dispatch optimization can be run.

CapacityExpansion within the broader ecosystem

CapacityExpansion is the first package to provide capacity expansion planning in Julia.

Multiple other software tools exist that support energy system planning both in Julia and other programming languages. We provide an overview of the broader ecosystem for orientation.

The [Jouliu](#) package in Julia provides a modeling framework for electricity systems that optimizes dispatch (Weibezahn & Kendzierski, n.d.). The generation, storage, and transmission capacities have to be provided as exogenous parameters.

The [anyMOD](#) package in Julia provides a modeling framework for energy system models with a focus on multi-period capacity expansion. Each energy carrier can be modeled on separate geospatial and temporal sets.

The [InfrastructureModels](#) package in Julia is a combination of multiple steady-state network optimization models to model electricity, gas, and water networks.

The [Calliope](#) package in Python provides a framework to develop energy system models (Pfenninger & Pickering, 2018). An experimental back end for JUMP exists with the package [CalliopeJUMP](#). However, the CalliopeJUMP back end is called from within the Calliope framework in Python.

The [URBS](#) package in Python provides a framework for capacity expansion planning that uses a single .xlsx-file including the input data in form of tables and that contains information for the model configuration.

The [Switch](#) package in Python provides a framework for capacity expansion planning of electricity systems. The switch framework allows modular optimization model setup and the integration of custom modules.

The model [DIETER](#) package provides a framework for capacity expansion planning using the proprietary GAMS software for optimization and .xlsx-files for data import and export.

Applications

CapacityExpansion can be applied to plan and validate a variety of energy systems. The focus on time-series aggregation, storage modeling, and integration of multiple energy carriers makes it especially valuable for the planning and validation of future energy systems with higher shares of non-dispatchable generation and sector coupling technologies. The scale of the modeled energy system can range from quarters to entire global regions and is only restricted by the computational complexity of the model.

CapacityExpansion has been used in academic research. It was used to analyze and improve the impact of time series aggregation methods on low emission energy systems (Kuepper, 2019).

Furthermore, CapacityExpansion has been used as an educational tool. It is used for modeling exercises in the Stanford University course “Advanced Methods in Modeling for Climate and Energy Policy”. The teaching material is also provided open-source in the package.

References

- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1), 65–98. doi:[10.1137/141000671](https://doi.org/10.1137/141000671)
- Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2), 295–320. doi:[10.1137/15M1020575](https://doi.org/10.1137/15M1020575)
- Gacitua, L., Gallegos, P., Henriquez-Auba, R., Lorca, Negrete-Pincetic, M., Olivares, D., Valenzuela, A., et al. (2018). A comprehensive review on expansion planning: Models and tools for energy policy analysis. *Renewable and Sustainable Energy Reviews*, 98(October 2017), 346–360. doi:[10.1016/j.rser.2018.08.043](https://doi.org/10.1016/j.rser.2018.08.043)
- Johnston, J., Mileva, A., Nelson, J. H., & Kammen, D. M. (2013). SWITCH-WECC: Data, assumptions, and model formulation, (October), 1–75.
- Kotzur, L., Markewitz, P., Robinius, M., & Stolten, D. (2018). Time series aggregation for energy system design: Modeling seasonal storage. *Applied Energy*, 213(October 2017), 123–135. doi:[10.1016/j.apenergy.2018.01.023](https://doi.org/10.1016/j.apenergy.2018.01.023)
- Kuepper, L. E. (2019). *Investigation of time-series aggregation methods for infrastructure optimization of low emissions energy systems* (MS thesis). Stanford University, RWTH Aachen University, Aachen.
- Pfenninger, S., & Pickering, B. (2018). Calliope: a multi-scale energy systems modelling framework. *Journal of Open Source Software*, 3(29), 825. doi:[10.21105/joss.00825](https://doi.org/10.21105/joss.00825)
- Teichgraeber, H., & Brandt, A. R. (2019). Clustering methods to find representative periods for the optimization of energy systems: An initial framework and comparison. *Applied Energy*, 1283–1293. doi:[10.1016/j.apenergy.2019.02.012](https://doi.org/10.1016/j.apenergy.2019.02.012)
- Teichgraeber, H., Kuepper, L., & Brandt, A. (2019). TimeSeriesClustering: An extensible framework in Julia. *Journal of Open Source Software*, 4(41), 1573. doi:[10.21105/joss.01573](https://doi.org/10.21105/joss.01573)
- Weibezahn, J., & Kendzioriski, M. (n.d.). Illustrating the benefits of openness: A large-scale spatial economic dispatch model using the julia language. *Energies*, 12(6). doi:[10.3390/en12061153](https://doi.org/10.3390/en12061153)