The Journal of Open Source Software

Osier: A Python package for multi-objective energy system optimization

Samuel G. Dotson [1¶] **and Madicken Munk** [1]

1 Department of Nuclear, Plasma, and Radiological Engineering, University of Illinois Urbana-Champaign, USA ¶ Corresponding author

DOI: 10.21105/joss.06919

Software

- Review
- Repository
- Archive

Editor: Prashant Jha

Reviewers:

- @fredshone
- @victoraalves

Submitted: 05 April 2024
Published: 06 December 2024

Summary

Transitioning to a clean energy economy will require expanded energy infrastructure. An equitable, or just, transition further requires the recognition of the people and communities directly affected by this transition. However, public preferences may be ignored during decision-making processes related to energy infrastructure due to a lack of technical rigor or expertise (Johnson et al., 2021). This challenge is further complicated by the fact that people have and express preferences over many dimensions simultaneously. Multi-objective optimization offers a method to help decision makers and stakeholders understand the problem and analyze tradeoffs among solutions (Liebman, 1976). Although, to date, no open-source multi-objective energy modeling frameworks exist. Open-source multi-objective energy system framework (`osier`) is a Python package for designing and optimizing energy systems across an arbitrary number of dimensions. `osier` was designed to help localized communities articulate their energy preferences in a technical manner without requiring extensive technical expertise. In order to facilitate more robust tradeoff analysis, `osier` generates a set of solutions, called a Pareto front, that are composed of a number of technology portfolios. The Pareto front is calculated using multi-objective optimization using evolutionary algorithms. `osier` also extends the common modeling-to-generate-alternatives (MGA) algorithm into N-dimensional objective space, as opposed to the conventional single-objective MGA. This allows users to investigate the near-optimal space for appealing alternative solutions. In this way, `osier` may aid modelers in addressing procedural and recognition justice.

Statement of Need

There are myriad open- and closed-source energy system optimization models (ESOMs) available (Pfenninger et al., 2022). ESOMs can be used for a variety of tasks but are most frequently used for prescriptive analyses meant to guide decision-makers in planning processes. However, virtually all of these tools share a fundamental characteristic: Optimization over a single economic objective (e.g., total cost or social welfare). Simultaneously, there is growing awareness of energy justice and calls for its inclusion in energy models (Pfenninger et al., 2014; Vågerö & Zeyringer, 2023). Two well known open-source ESOMs, Calliope (Pfenninger & Pickering, 2018) and Python for Power Systems Analysis (PyPSA) (Brown et al., 2018), partially address equity issues by implementing MGA, but this does not resolve the limitations of mono-objective optimization. Some studies incorporate local preferences into energy system design through multi-criteria decision analysis (MCDA) and community focus groups (Bertsch & Fichtner, 2016; McKenna et al., 2018; Zelt et al., 2019). But these studies rely on tools with pre-defined objectives which are difficult to modify. Without the ability to add objectives that reflect the concerns of a community, the priorities of that community will remain secondary to those of modelers and decision makers. A flexible and extensible multi-objective framework that fulfills this need has not yet been developed. `osier` closes this gap.

The Journal of Open Source Software

Design and Implementation

The fundamental object in `osier` is an `osier.Technology` object, which contains all of the necessary cost and performance data for different technology classes. `osier` comes pre-loaded with a variety of technologies described in the National Renewable Energy Laboratory's (NREL) Annual Technology Baseline (ATB) dataset (National Renewable Energy Laboratory, 2023) but users are also able to define their own. In order to run `osier`, users are required to supply an energy demand time series and a list of `osier.Technology` objects. Users can optionally provide weather data to incorporate solar or wind energy.

A set of `osier.Technology` objects, along with user-supplied demand data, can be tested independently with the `osier.DispatchModel`. The `osier.DispatchModel` is a linear programming model implemented with the `pyomo` library (Hart et al., 2011). For investment decisions and tradeoff analysis, users can pass their portfolio of `osier.Technology` objects, energy demand, and their desired objectives to the `osier.CapacityExpansion` model, the highest level model in `osier`. The `osier.CapacityExpansion` model is implemented with the multi-objective optimization framework, `pymoo` (Blank & Deb, 2020). Figure 1 overviews the flow of data through `osier`.

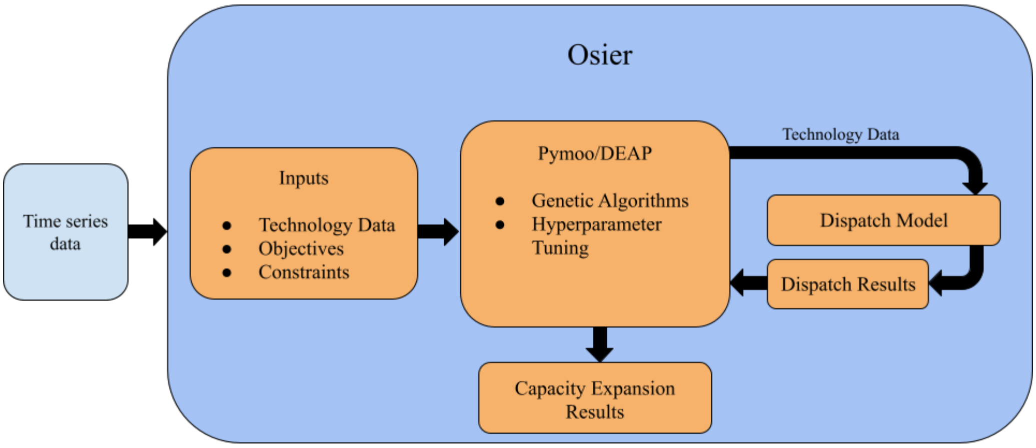

Figure 1: The flow of data into and within `osier`.

Key Features

In addition to being the first and only open-source multi-objective energy modeling framework, `osier` has a few key features that further distinguishes it from other modeling frameworks. First, since `osier.Technology` objects are Python objects, users can modify values and assumptions, or assign new attributes to the tested technologies. Second, contrary to conventional energy system models, `osier` has no required objectives. While users may choose from a variety of pre-defined objectives, they may also declare their own objectives based on any quantifiable metric. The requirements for a bespoke objective are:

1. The first argument must be a list of `osier.Technology` objects.
2. The second argument must be the results from an `osier.DispatchModel`. But this may be a simple placeholder with a default value of `None`.
3. The function must return a single numerical value.
4. The final requirement, is that all `osier.Technology` objects possess the attribute being optimized.

These two features acknowledge that a modeler cannot know *a priori* all possible objectives or parameters of interest. Allowing users to define their own objectives and modify technology objects (or simply build their own by inheriting from the `osier.Technology` class) accounts for this limitation and expands the potential for incorporating localized preferences. Lastly, in order to account for unmodeled or unmodelable objectives, `osier` extends the conventional MGA algorithm into N-dimensions by using a farthest-first-traversal in the design space.

Sample Results and Interpretation

When solving a multi-objective problem, `osier` generates a set of co-optimal solutions rather than a global optimum, called a Pareto front. Figure 2 shows a Pareto front from a problem that simultaneously minimizes total cost and lifecylce carbon emissions.

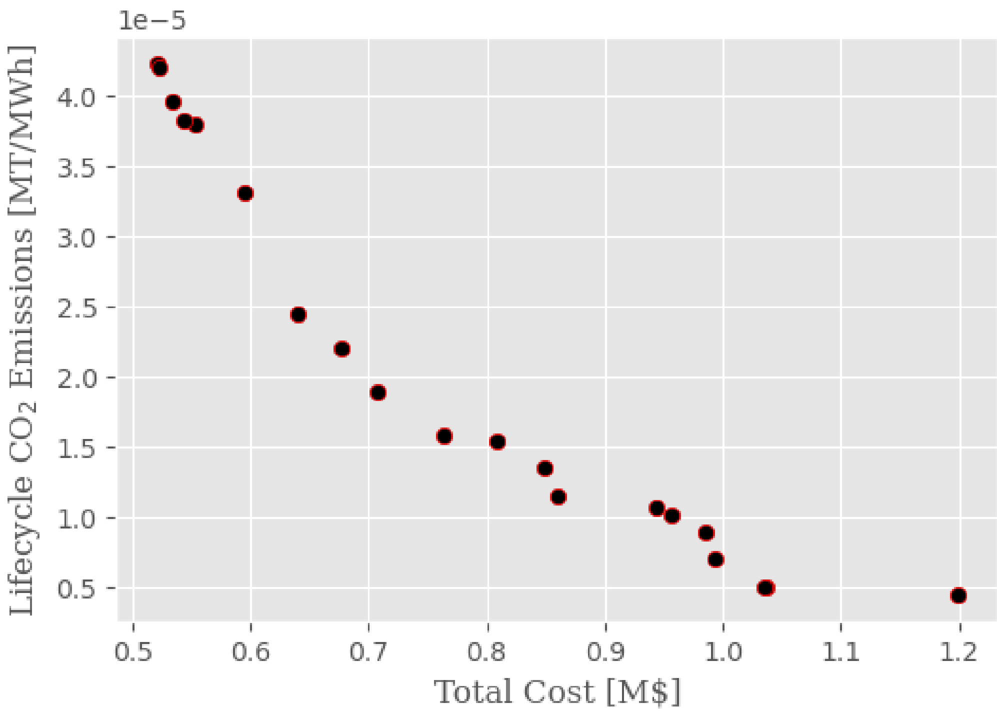

Figure 2: A Pareto front generated by`osier`.

Each point on this Pareto front represents a different technology portfolio (i.e., different combination of wind, natural gas, and battery storage). Figure 3 illustrates the variation in solutions from the Pareto front in Figure 2. In this case, the range of wind capacity is wider than the range of capacities for natural gas and battery storage.

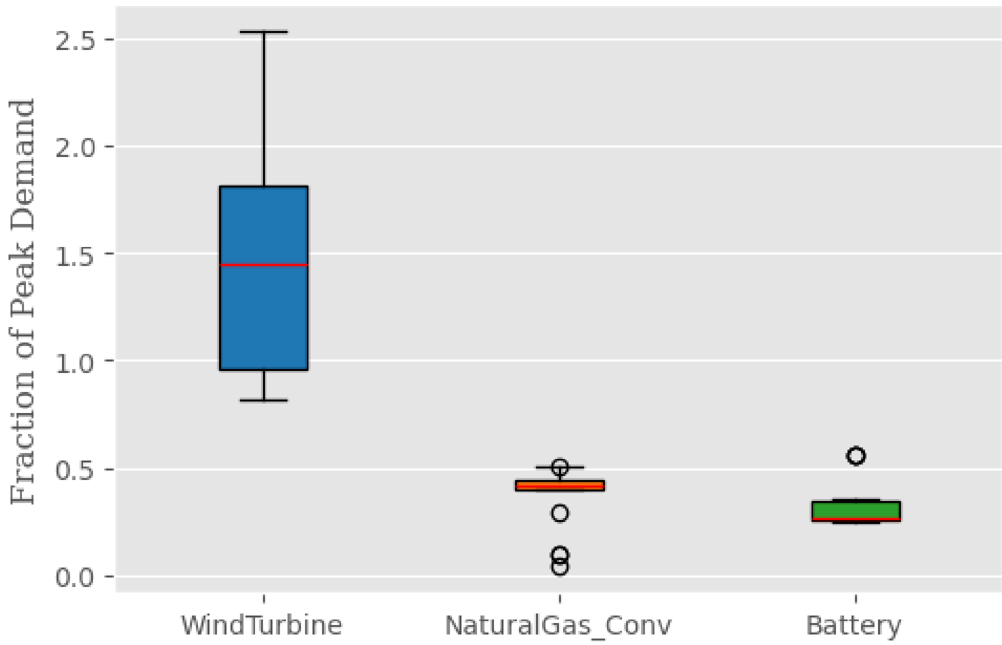

Figure 3: The variance in technology options along a Pareto front.

Documentation

`osier` offers robust documentation with detailed usage examples at osier.readthedocs.io.

Acknowledgements

Samuel Dotson, the corresponding and lead author of this publication is responsible for the conceptualization of `osier`, developing `osier` as a software, in preparing this manuscript for publication, and for performing analysis to validate `osier`. Madicken Munk provided resources and supervision for the work, as well as assisted in the review and editing of the manuscript. Samuel Dotson was supported by the Nuclear Regulatory Commission Fellowship program. This research was part of the Advanced Reactors and Fuel Cycles (ARFC) group in the Department of Nuclear, Plasma, and Radiological Engineering (NPRE) at the University of Illinois Urbana-Champaign. To that end, the authors would like to acknowledge ARFC members Oleksander Yardas, Luke Seifert, Nathan Ryan, Amanda Bachmann, and Sun Myung Park for their contributions in reviewing pull requests supporting the creation of osier. Additionally, Samuel Dotson was supported by the Felix T. Adler Fellowship through NPRE. Finally, the authors would like to thank the JOSS reviewers for their time and commentary in reviewing this manuscript.

References

Bertsch, V., & Fichtner, W. (2016). A participatory multi-criteria approach for power generation and transmission planning. *Annals of Operations Research*, *245*(1), 177–207. https://doi.org/10.1007/s10479-015-1791-y

Blank, J., & Deb, K. (2020). Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, *8*, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567

Brown, T., Hörsch, J., & Schlachtberger, D. (2018). PyPSA: Python for Power System

Analysis. *Journal of Open Research Software*, *6*(1), 4. https://doi.org/10.5334/jors.188

Hart, W. E., Watson, J.-P., & Woodruff, D. L. (2011). Pyomo: Modeling and solving mathematical programs in Python. *Mathematical Programming Computation*, *3*(3), 219–260. https://doi.org/10.1007/s12532-011-0026-8

Johnson, M. F., Sveinsdóttir, A. G., & Guske, E. L. (2021). The Dakota Access Pipeline in Illinois: Participation, power, and institutional design in United States critical energy infrastructure governance. *Energy Research & Social Science*, *73*, 101908. https://doi.org/10.1016/j.erss.2021.101908

Liebman, J. C. (1976). Some Simple-Minded Observations on the Role of Optimization in Public Systems Decision-Making. *Interfaces*, *6*(4), 102–108. https://doi.org/10.1287/inte.6.4.102

McKenna, R., Bertsch, V., Mainzer, K., & Fichtner, W. (2018). Combining local preferences with multi-criteria decision analysis and linear optimization to develop feasible energy concepts in small communities. *European Journal of Operational Research*, *268*(3), 1092–1110. https://doi.org/10.1016/j.ejor.2018.01.036

National Renewable Energy Laboratory. (2023). *2023 Annual Technology Baseline (ATB)*. https://atb.nrel.gov/

Pfenninger, S., Hawkes, A., & Keirstead, J. (2014). Energy systems modeling for twenty-first century energy challenges. *Renewable and Sustainable Energy Reviews*, *33*, 74–86. https://doi.org/10.1016/j.rser.2014.02.003

Pfenninger, S., & Pickering, B. (2018). Calliope: A multi-scale energy systems modelling framework. *Journal of Open Source Software*, *3*(29), 825. https://doi.org/10.21105/joss.00825

Pfenninger, S., Schlect, I., Trondle, T., & Brown, T. (2022). Openmod - Open Energy Modelling Initiative. In *openmod-initiative*. https://www.openmod-initiative.org/.

Vågerö, O., & Zeyringer, M. (2023). Can we optimise for justice? Reviewing the inclusion of energy justice in energy system optimisation models. *Energy Research & Social Science*, *95*, 102913. https://doi.org/10.1016/j.erss.2022.102913

Zelt, O., Krüger, C., Blohm, M., Bohm, S., & Far, S. (2019). Long-Term Electricity Scenarios for the MENA Region: Assessing the Preferences of Local Stakeholders Using Multi-Criteria Analyses. *Energies*, *12*(16), 3046. https://doi.org/10.3390/en12163046