Energy system modelling is of high importance to investigate different scenarios in their technical, economical and environmental feasibility. The interplay of different technologies and energy flows in respective models can be represented as directed graphs in a generic but comprehensible formalism. However, additional effort is needed to create specific models and to derive an optimal sizing or operation of components. To tackle this problem, oemof.solph facilitates the formulation of (mixed-integer) linear programs from a generic object-oriented structure. Its structure allows to create models on different levels of detail by means of predefined components and an optional formulation of additional expressions and constraints. With its open and documented code base, extensive collection of examples and an active community it is useful across many levels, from simple applications to advanced modelling.
1. About solph and oemof

The solph package [1] is part of the open energy modelling framework (oemof), which has been developed to model integrated energy systems [2]. Its basis is a graph structure consisting of buses and components connected by directed edges representing the flow of energy carriers and resources, their conversion and consumption. This structure allows to model different sectors of the energy system equivalently. Whereas the graph holds information on the topology and relationships between the nodes, solph converts this graph into an optimisation model.

It is implemented in Python based on the optimisation package pyomo [3]. In combination with the object-oriented graph based data model, modelling does neither require deep knowledge about mathematical optimisation nor an (algebraic) modelling language to implement linear (LP) and mixed-integer linear (MILP) problems. Instead, it can be undertaken by comprehensive domain-specific, in this case energy system, knowledge.

Within the workflow, different versions of an energy system model are built:

1. The oemof graph version is created.
2. The associated pyomo optimisation instance is constructed.
3. The model is passed to an external solver (for example using the LP file format).

The solver optimises for minimal costs that can be economic, environmental, technical or any other type of cost. Currently, users can create economic dispatch and unit commitment models with an additional investment optimisation. Due to the modular structure of the package, the API and underlying mathematical constraints of new components can be developed and reviewed individually. This allows to review, discuss, improve and add components in a collaborative process within the community and beyond. A comparison with other (linear) optimisation tools [4,5] has proven consistency of results.

2. Contribution to the scientific community

The research area of energy system analysis faces different challenges which lie in an increasing complexity, higher levels of uncertainty and more interdisciplinary properties of energy systems [6]. Moreover, the challenges of energy system modelling [7] have brought up the demand for suitable tools to better address scientific standards [8,9]. The latter were among the main drivers for the design of the oemof framework and the solph package, respectively [2]. Based on the previous experience with building energy system models, the initial development of oemof aimed to contribute to the scientific community through the following features:

Supporting reusability

A strong motivation to build oemof.solph is to avoid double-work. It is intended to relieve modellers from programming utility code and let them focus on their core task: modelling energy systems. To facilitate this, the source code has to be well tested and documented. Validation, including unit testing and usage within a community and many use cases, reduces the likeliness of bugs and creates a tested code base. Finally, a detailed documentation with many examples lowers the entry barrier for new users in the field of energy modelling. Therefore, oemof.solph is also suitable for projects and studies in which modelling is not the main focus or projects with a limited time budget. Beyond citations in scientific publications, usage itself remains almost invisible. However, a number of studies has been published, not only in scientific journals. Solph\(^1\) is used by a number of other open source projects on GitHub [10,11]. Additionally, we get support questions from companies in the consulting and energy sector work with oemof.solph.

Building a developer community

Since the very beginning of oemof’s development it was a strongly pursued goal to create a developer community that is open for members from various institutions. Such research groups value the existing code base as well as the possibility to discuss extensions in the community and to get a review for new components or additions to oemof.solph from other modellers. Since the beginning, new research groups from other institutions joined the developer team. For example, researchers of the projects Quarree100 [12] and EnAQ [13]. Both users at different levels and new developers, attend the oemof developer and user meetings that are held on a regular basis.

Supporting open science principles and transparency

Using solph it is possible to publish studies following Open Science and transparency rules. The permissive open source license facilitates users to build an open science tool chain. Models for the electricity sector [14–16] and for district heating grids [17] are show-casing how to implement a model chain incorporating open data from OpenStreetMap. Even without building a fully automated tool chain, the permissive license of the oemof.solph’s code facilitates to publish the model alongside with the data. This was done for a study about the validation of exit strategies for lignite and coal in Brandenburg [18].

Integrating a modelling toolbox

While oemof is meant to be a software cosmos, suitable for building model-chains and providing easy-to-use libraries and tools for energy system modelling, as of today, solph is the most recognised library of the framework. Thus, most studies citing oemof also use solph. However, there are also examples that use exclusively specific other oemof packages such as demandlib [19] for load curve generation [20], TESpy [21] for thermal engineering systems or the windpowerlib [22] for wind power feed-in calculation based on weather data.

3. Specific impact on research

The oemof.solph library is used for a wide variety of research questions. Its generic design allows to model energy systems in any possible combination of sectors. For the electricity sector, example publications discuss adding storage to existing supply [23,24] – in one case in addition to electrification of agricultural machines [25]. Others analyse the possible vulnerability for future energy systems [26], or optimise the utilisation of pumped energy storage in Switzerland [27]. The possible integration of more renewable energy has been investigated for the case of Italy using both, a fixed model [28,29] and linear invest optimisation showing pathways to more renewable energies [30]. Other uses include the layout of a complete mini-grid [31–33], or providing the technical side of study social and ecological factors in energy system modelling [34–36]. For the heat sector, studies exist that determine the optimal size of heat and cold storage [37,38], check the use of district heating for demand side management [28], do a life cycle analysis [39], or increase the model accuracy using a pre-calculated fluctuating temperature supply [40]. Others use the solph for dispatch optimisation

\(^1\) Note that some dependencies still point to the oemof and not to the oemof-solph repository.
inside a non-linear size-optimisation heuristics for electricity and heat supply [41], or compare different mobility options [42].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

U.K. and P.S. acknowledge funding by the Federal Ministry for Economic Affairs and Energy (BMWi), Germany and the Federal Ministry of Education and Research (BMBF) of Germany (grant number 03SBE113B and 03SBE111, respectively).

References


