SpineOpt: A flexible open-source energy system modelling framework

Maren Ihlemann, Iasonas Kouveliotis-Lysikatos, Jiangyi Huang, Joseph Dillon, Ciara O’Dwyer, Topi Rasku, Manuel Marin, Kris Poncelet, Juha Kiviluoma

Working Paper Series No. ESIM2021-11
Last update: September 29, 2021
SpineOpt: A flexible open-source energy system modelling framework

Maren Ihlemanna, Iasonas Kouveliotis-Lysikatob, Jiangyi Huangc, Joseph Dillon, Ciara O’Dwyer, Topi Raskuc, Manuel Marinb, Kris Ponceleta and Juha Kiviluomac

aKU Leuven, Celestijnenlaan 300, Box 2421, B-3001 Leuven, Belgium
bKTH-Royal Institute of Technology, Brinellvägen 8, 114 28 Stockholm, Sweden
cEnergy Reform Ltd., 22 Avoca Drive, Bray, Co. Wicklow, Ireland
dUniversity College Dublin, Belfield, Dublin 4, Ireland
eVTT Technical Research Centre of Finland Ltd, Box 1000, FI-02044 VTT, Espoo, Finland

ARTICLE INFO

Keywords:
Open source tool
Energy system modelling
Energy system analysis
Integrated energy systems
Investment planning
Sector coupling

ABSTRACT

The transition towards more sustainable energy systems poses new requirements on energy system models. New challenges include representing more uncertainties, including short-term detail in long-term planning models, allowing for more integration across energy sectors and dealing with increased model complexities. SpineOpt is a flexible, open-source, energy systems modelling framework for performing operational and planning studies, consisting of a wide spectrum of novel tools and functionalities. The most salient features of SpineOpt include a generic data structure, flexible temporal and spatial structures, a comprehensive representation of uncertainties, and model decomposition capabilities to reduce the computational complexity. These enable the implementation of highly diverse case studies. SpineOpt’s features are presented through several publicly-available applications along with an illustrative case study.

1. Introduction

Energy system models serve decision making in both public and private domains. Energy policy and regulation influence investment decisions made by different public and private entities. In addition, energy system models guide the operation of energy systems in conjunction with sector specific models. Energy system models are used for various research-related purposes ranging from the analysis of emission reduction pathways to the investigation of costs and benefits of particular technologies. In this article, we will introduce a new energy system model that is more feature-complete than existing alternatives and can therefore contribute to serving the purposes listed above.

The importance of transparency and reproducibility has been growing both in academia and in public decision making. Commercial models are typically not fully transparent and their license costs add further barriers for reproducing and improving results. SpineOpt is fully open source, including the software it runs on. SpineOpt is developed in Julia, an emerging multi-platform programming language for high performance computing, and utilises its mathematical optimisation package JuMP.

The importance of operational details in planning future energy systems has been rapidly increasing. One driver for this is the increasing role of variable power generation, which requires that the energy system models consider higher temporal granularity, longer time series and new aspects in power system stability. Another important driver is a consequence of sector coupling through the electrification of transport, space heating and industries. As the other sectors become more integrated with electricity, they need to be modelled at a granularity that is sufficient for describing the potential flexibility they can bring to the power system, which is dominated by variable and uncertain power generation. SpineOpt can represent different energy sectors using representation typically available only in sector specific models. Helistö et al. [1] conceptualise different approaches on how operational details can be included in energy system models. SpineOpt is built in a way that allows to use any of those modelling approaches depending on the requirements and available resources. To reduce the computational requirements, SpineOpt has inbuilt capabilities for decomposing investment problems.
Many energy system models have been built assuming a specific type of energy system. This results in hard-coding of model features that may need to change when the future reveals new modelling challenges. SpineOpt has been built with adaptability in mind. The temporal and stochastic structures are highly flexible, new sectors and commodities can be added as input data, different conversion and transfer methods can be applied through data-driven decisions, and new constraints can be added when needed. The modelling decisions are left to the user, but with an attempt to guide the user into providing appropriate data for the chosen modelling method.

The novelty of SpineOpt lies in the way it puts together a rich set of features. Most of these features are available in some other open source models, but we are not aware of any model that offers such a diverse combination. Moreover, to the best of our knowledge, the temporal and stochastic structure of SpineOpt is one of a kind in its adaptability. Different parts of the same model, e.g. different locations, sectors, or technologies, can have different temporal resolution as well as different stochastic structures. The stochastic structure allows for the branching and merging of scenarios. SpineOpt also has an unmatched representation of sector specific modelling detail in a single energy system model.

The paper is structured as follows: In Section 2, SpineOpt is positioned among other existing open-source modelling tools, focusing on the key features of the different models, that each have different functionalities. The general architecture of SpineOpt and its specific features are presented in Section 3. Applications of SpineOpt and an illustrative case study are presented in Section 4. Section 5 concludes the paper giving a perspective for future developments and enhancements.

2. Comparison of SpineOpt with state-of-the-art modelling tools

In this section, SpineOpt is positioned in the domain of open source energy system models. We preclude commercial models as their methodologies are not fully transparent. SpineOpt is positioned as a bottom-up, partial-equilibrium model. In the forthcoming subsections we compare SpineOpt with other open-source energy models belonging in the same category. In [2], a large number of open-source modelling tools are compared according to their features. PyPSA [3], OSeMOSYS [4], Switch [5] and TEMOA [6] are identified as top performing open-source tools, based on various comparison benchmarks. The modelling framework TIMES [7] is also included in the comparison, as its conceptual approach is relevant to SpineOpt. Numerous open-source modelling tools and frameworks exist, and the interested reader is referred to the excellent reviews in [2, 8, 9] and [10].

In the following, we compare the structural elements of these models, i.e. the data structure, the temporal representation, the representation of uncertainties and the optimisation algorithms. Section 2.5 gives an overview of SpineOpt’s features in comparison to the other models.

2.1. Data structure

The data structure refers to the way the data is organised, managed and stored. For an energy system model, this comprises the data that defines the structure of an optimisation model, and the data that represents a specific technical system. It encompasses all available information, which uniquely and consistently define a particular model (such as the system’s topology and the input data), together with other meta-parameters, required for instantiating and optimising the model. PyPSA’s data structure [3] is based on a specific list of fundamental components, which are connected and customised by selecting values for their predefined properties. The list of available components include buses, generators, loads, lines, transformers, storage units, the energy carrier, etc. All components are attached to a bus, the role of which is to enforce the energy balance constraint. PyPSA offers specialised model-building components, which mostly focus on modelling Power Systems. PyPSA is implemented using the open-source python package Pyomo. The OSeMOSYS model [4] separates the energy system model into a series of component blocks, with each block forming the specification for different aspects of the optimisation model, i.e., the objective function, costs, storage parameters, capacity adequacy, the energy balance, constraints and emissions. In OSeMOSYS, the block modelling approach greatly simplifies the process of building long-term energy models, but has limited flexibility for designing or modifying the lower-resolution operational problem. Originally written in the free software GNU MathProg, OSeMOSYS is now also available in Pyomo, PuLP and GAMS. In the TEMOA project [6], the model is structured using commodities and their associated processes. Commodities can signify input resources, while the processes represent the potential conversion technologies between them. The TEMOA concept includes the basic categories of resources, transformations and demands, but custom relationships between the components are not directly provided. TEMOA is implemented using the python package Pyomo. The Switch platform [5] uses individual
modules for modelling policies, energy sources, transmission lines, etc. When the input data is parsed, the spatio-
temporal power system model is generated with the various constraints and objective terms. Unlike the other compared
models, core Switch only represents the power system sector. Switch is implemented using the python package Pyomo.
The TIMES energy economy [7] uses (and combines) three fundamental types of entities: Technologies (also called
processes), which represent transformations such as energy conversions, Commodities which are generally produced
and/or consumed by processes, and Commodity flows representing the actual flows of a specific commodity. The data
structure of TIMES resembles that of SpineOpt, allowing the specification of different geographical regions and the
linking of technologies to them. In TIMES the user interface is commercial, and using it without an interface is not
straightforward. Also it relies on commercial software for generating the optimisation program (GAMS).

SpineOpt has a generic structure, which is complemented by a wide range of technology specific methods and
parameters. In contrast to Times, the graphical user interface is open-source, using Spine Toolbox, an open-source
workflow management system [11] to define, manage, and execute energy system models. The data structure of
SpineOpt is also based on the Spine Toolbox database design, which uses entity classes, entities and their parameters.
SpineOpt uses this structure to implement a multi-commodity, flow-based model of a generic energy system, along
with methods and parameters to implement specific technologies and sector specific physics. The fundamental building
blocks of SpineOpt are: nodes, commodities, units and connections, and will be further discussed in Section 3.1.
These generic building blocks are used to model flows, conversions and balances in any energy sector. SpineOpt
is implemented in Julia, using the domain-specific modelling language JuMP [12] for mathematical optimisation.

2.2. Temporal structure

The temporal dimension of a model is employed to capture the temporal variability of the input data and the
dynamics of the system. In regard to energy models, the temporal scope can differ significantly. Short-term
optimisation models usually concern unit-commitment and dispatch decisions, while long-term optimisation models
aim at calculating the optimal investment decisions in significantly larger time horizons, often spanning several
decades. PyPSA traditionally optimises total system costs on a yearly basis. PyPSA enables high-detail operational
resolutions with sequential time steps. The default operational resolutions of PyPSA is hourly, with evenly-spaced,
continuous time series as inputs. Recently, support for multi-period investments has been added to PyPSA. The
implementation of temporal aggregation methods seems to be still ongoing. OSeMOSYS, TIMES and TEMOA are
designed to optimise time horizons of up to decades by employing temporal aggregation techniques to limit the resulting
model’s dimension. The temporal structures of these models are based on the concept of disjoint representative time
slices. However, in order to be able to represent the seasonal behaviour of long-term storage systems, they enable the
user to define different time slice levels, in order to capture the inter-seasonal dynamics of the storage constraints. The
core structures of OSeMOSYS and TEMOA are not designed for optimising models with inter-temporal constraints
precluding adequate assessment of flexibility issues. TIMES and TEMOA allow in a limited way tracking of different
commodities and technology productions, respectively, on different resolutions. In Switch, both, disjoint and sequential
time slices are supported. Chronological sequences of time slices are realised through time series. As a circular
condition is imposed on time series, inter-seasonal dynamics of long-term storage systems cannot be captured for
disjoint time series [5].

SpineOpt supports all of the above options for defining the temporal structure of a model with significant additional
capabilities, further increasing the model’s flexibility and efficiency. SpineOpt can represent both sequential or disjoint
time slices. However in SpineOpt the temporal resolution can vary in spatial, sectoral, and technological dimensions.
This is useful where certain commodities’ or units’ dynamics and/or data are such that it is efficient to treat them at
lower temporal resolutions, reducing the size of the resultant model [13]. Investment, flow and online status decision
variables can also be modelled with individual, independent resolutions varying on a per node, unit, or connection
level. The temporal resolution for all model decisions can also evolve over time. Depending on the accuracy of certain
input parameters, e.g., a wind forecast, a high resolution might be beneficial at the beginning of the optimisation, which
can be gradually reduced for the longer look-ahead decisions. Representative periods are supported with weighting and
optional ordering to support modelling of investments and long-term storage systems. The architecture of the temporal
structure of SpineOpt will be discussed at length in Section 3.3.

2.3. Representation of uncertainties

To address the impact of uncertainty and variability, stochastic, scenario-based approaches are commonly used.
Apart from PyPSA, which does not explicitly provide the possibility for scenario-based optimisation [3], other tools
usually provide some functionalities to account for non-deterministic parameters. Although OSeMOSYS does not support different alternatives as a core functionality, it comes with an extension that handles uncertainties by generating potential disturbances for the stochastic parameters, thus representing the realisation of different scenarios. TEMOA, although not directly handling stochastic parameters, is able to explore highly variable design solutions, by employing the heuristic method Modelling to Generate Alternatives (MGA). This algorithm provides near optimal solutions, thus producing significantly differing alternative energy system design solutions, to be further evaluated by the analyst [14]. Scenarios for different system setups also exist in Switch, which provides a scenario-solving system and the option to run several separate scenarios at once [5]. TIMES, on the other hand, offers the possibility to define a multi-stage stochastic energy system model, by representing the probability distributions of the uncertain parameters on an event tree [7]. In SpineOpt, both the alternative scenarios and multi-stage stochastic formulations are supported. The impact of variations in input data can be captured by defining data scenarios containing alternative values for any input parameter. This approach results in one outcome per scenario and is often used to model different high-level assumptions and/or policy decisions. Additionally, a generic stochastic data structure provides the user with the flexibility to define a multi-stage stochastic program which minimises total expected costs considering the probability distributions of any input parameter. This combination enables the comprehensive consideration of a wide variety of uncertainties. Moreover, the stochastic structure of SpineOpt, similar to the temporal structure, gives the possibility to model the stochastic inputs with different spatial dimensions independently i.e., belonging to different regions or sectors. The stochastic structure will be described at length in Section 3.4.

2.4. Model Linking & Decomposition

Linking of models can be used to combine functionalities of individual models with specific functionalities. If this happens unidirectional, we refer to this process as soft-linking, e.g. passing the results of an investment model to an operational model. In principle, each of the presented models can be soft-linked, but this can be challenging in practice as it often involves significant data transformations and complex workflows. To facilitate this process, SpineOpt is integrated within the Spine Toolbox application, specifically designed to seamlessly handle such data exchanges and process orchestration. Note that this functionality of Spine Toolbox is model-agnostic and can be used with any model. Soft-linking, however, does not allow for full recursion between models, and convergence to an optimal solution cannot be guaranteed. To achieve this, decomposition is often used where one large problem (e.g., a combined operations and planning problem) is decomposed into an equivalent set of smaller problems, reducing the computational complexity and maintaining tractability. A rolling horizon optimisation decomposes a single long duration problem into a series of (usually) overlapping, shorter-duration problems which are solved sequentially using a rolling window. The term rolling horizon is typically used in the context of decomposing dispatch models, while myopic is typically used for decomposed investment models with limited foresight. Among the compared models, the rolling horizon on a dispatch level is only supported by SpineOpt, while myopic optimisation is supported by both, TIMES and SpineOpt. While short-term rolling horizon optimisation can be useful to reduce the complexity of operational models, longer-term decisions, such as investments in seasonal storage systems, cannot be captured. As a solution to this problem, SpineOpt uses the Benders decomposition technique [15] to divide a single large scale optimisation problem, with operations and investment decisions, into a master problem that optimises investment decisions, and a sub-problem that optimises operational decisions (which may itself utilise rolling horizon optimisation).

2.5. Overview of SpineOpt’s features compared to other models

In the following Table 1, an overview of the diverse features and flexibility options of SpineOpt is presented in comparison to the other models. We stress that we are not aiming to present a complete overview of all individual features, but focus on the aspect of versatility and adaptability. The choice of a certain modelling tool will always depend on the modelling task of interest and its requirements. A manifold of different models and their features have been compared in [2, 8, 9] and [10]. Under Data Structure & Scope, a model is marked with flexible spatial scope if there is an option to model diverse geographical scopes under different levels of detail. Varying resolution under Temporal Concept captures the extent to which temporal granularity can be varied in specific dimensions: temporal (T) corresponds to varying temporal resolutions over the modelling horizon; technological varying resolution (E) to independent resolutions for flow and unit commitment variables of individual units; sectoral varying resolution (C) to independent resolutions for different flow commodities; spatial varying resolution (Z) to independent resolutions for flow variables node-wise. Whether or not the temporal resolution of the energy system is customisable is indicated by user-defined resolution. The hierarchy of a temporal structure represents the relationships between overlapping
Table 1

Features of SpineOpt compared to other reviewed energy system models. The () refers to a partial support of the functionality. Other specific notations: Z-Spatial, C-Sectoral, T-Temporal, E-Technological, O-/PF-Optimal-/Power Flow, SC-Security-Constrained, PG-Pressure-driven Gas transfer, TH-Thermal, HD-Hydro

<table>
<thead>
<tr>
<th>Data Structure &amp; Scope</th>
<th>TIMES</th>
<th>TEMOA</th>
<th>Osemosys</th>
<th>PyPSA</th>
<th>Switch</th>
<th>SpineOpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sector coupling</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Multi-period investments</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Temporal Concept</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Varying resolution</td>
<td>(ZCT)</td>
<td>(CT)</td>
<td>(T)</td>
<td>–</td>
<td>(T)</td>
<td>ZCTE</td>
</tr>
<tr>
<td>User-defined resolution</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Flexible hierarchy</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Representation of Uncertainties &amp; Design Alternatives</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design Alternatives</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Stochastic programming</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Varying Scenarios</td>
<td>CE</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>ZCTE</td>
</tr>
<tr>
<td>Flexible stochastic branching</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Model Reduction, Linking &amp; Decomposition</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rolling-horizon</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Representative days</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Decomposition algorithm</td>
<td>–</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Technical detail</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit commitment</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Commodity Physics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC-OPF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC-PF,DC-SCOPF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DC-SCOPF,PG,TH,HD</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reserves</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ramping Constraints</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Piece-wise linear efficiency</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>Development &amp; Community</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Free of charge and fully open-source</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Documentation &amp; Tutorials</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Data management toolkit</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

3. Architecture of SpineOpt

In the following, the core structure and main features of SpineOpt are presented, each of which aim to facilitate cross-sectoral modelling and the integration of operational detail in large-scale optimisation problems. First, the generic data structure of SpineOpt is presented, followed by the flexible spatial, temporal and stochastic structures. Lastly, the decomposition and model linking functionalities are introduced.

3.1. SpineOpt Building Blocks

The SpineOpt data structure is based on a generic Entity – Attribute – Value design, with Objects, Classes and Relationships. This data structure is inherited from Spine Toolbox, an integrated software platform for data acquisition and processing, and for the orchestration of complex energy system modelling workflows. The Julia package SpineInterface [16] allows for a direct link between the toolbox’s data store and SpineOpt. The abstract object
classes of Spine Toolbox and the relationships between them facilitate the creation of an easily extendable structure to support the specification of multi-energy system models. In SpineOpt, this is achieved by creating specific classes to represent generic energy system model components. The building blocks of SpineOpt, as illustrated in Figure 1, can be divided in model components, i.e., classes related to the structure of the optimisation problem, and system components, i.e., classes representing the physical energy system.

The core element of the model components is the model object itself, which holds general information about the optimisation problem at hand (e.g., start/end of the optimisation, solution algorithm etc.). The temporal block objects hold information about the temporal resolution of specific parts of the model. The stochastic scenarios in combination with the stochastic structure hold information about the available scenarios and their branching structure.

The central entity of the system components is the node, i.e. a notional point where energy flows are balanced, and which may have a state, allowing storage and/or transport dynamics to be considered. The commodity object can be used to confer a physical meaning on energy flows to and from nodes e.g., electricity, heat, oil, gas, water, etc. The unit objects and connection objects enable interactions between nodes. Unit objects represents arbitrary flow conversions between any number of input and output nodes and have operational status variable. Connection objects enable energy transport flows between nodes and the supposition of specific transport physics (e.g., a gas-pipeline, an electricity line, a river, a means of transportation, etc.).

**Relationships classes**

To define the structure of an energy system model, relationships are used. Classes of relationships define which objects can be interlinked, and associate specific functionalities with these relationships, i.e., triggering variables or data parameters. For example, the relationship class unit__to_node allows a unit to be related to a node. Creating an instance of this relationship between a specific unit and a specific node creates a flow variable from the unit to the node, while parameters defined on this relationship, unit_capacity, for example, pertain to this particular flow. Figure 1 illustrates how the different objects are interlinked.

![Figure 1: Model and system components of SpineOpt. Object classes are presented as graphical symbols, and possible relationships between them through grey semi-transparent line segments.](image1)

As illustrated in Figure 1, the model object, which contains general information about the optimisation problem, is connected to two structural elements, the temporal block class and the stochastic structure class 1. By introducing relationships with a model object, temporal blocks and stochastic structures are included in the energy system model. The temporal resolution and the uncertainties related to a certain set of nodes are defined through a relationship between nodes, temporal blocks and stochastic structures. Each node is associated with exactly one commodity. The flows associated with units and connections inherit the temporal structure from the node they are connected to.

**3.2. Flexible spatial structure**

The general design philosophy of SpineOpt follows the following paradigm: *detail where needed, speed-up where possible*. This is also reflected in the design of the spatial structure of SpineOpt. The spatial structure of SpineOpt is non-hierarchical, meaning that a node can represent any spatial location, i.e., it can be a bus, a city, a region, a country. For simplicity and clarity, only one object of each kind is depicted (there can be a manifold), and some relationships are kept hidden. All possible relationship types between object classes are detailed in the online documentation [17].

---

1For simplicity and clarity, only one object of each kind is depicted (there can be a manifold), and some relationships are kept hidden. All possible relationship types between object classes are detailed in the online documentation [17].
etc. This design allows the user to have a detailed representation of the region of interest, while simultaneously allowing for a coarser spatial resolution for other areas of the model. An additional tool specifically for reducing electricity grids into multi-area equivalents has also been implemented for SpineOpt. The tool is based on the generalised Radial Equivalent Independent (REI) methodology [18], and is presented in a case study in [19].

The design of SpineOpt is commodity agnostic, meaning that the model does not inherently assume a specific type of energy to be modelled. This facilitates the functionality of sector coupling. An example of this approach is the implementation of the so-called \textit{fix\_ratio\_out\_in} constraint, which is fully described in the Supplementary Material and the online documentation [17]. This constraint is triggered through a relationship between a unit object and two nodes (or node groups), together with the parameter \textit{fix\_ratio\_out\_in}. The abstract formulation of this constraint has multiple uses; in its purest sense, it represents the ratio between incoming and outgoing flows of a unit. It can thus detail conversion efficiency, a separation or synthesising process, and more. In Figure 2, the concept of sector coupling (Regions A and B) and the concept of spatial aggregation (Regions A and C) are illustrated.

### 3.3. Flexible temporal structure

The paradigm used for the temporal structure developed in SpineOpt aims at enabling a detailed resolution where necessary, as well as allowing speed-up through temporal aggregation where possible. The temporal structure of SpineOpt has a flexible hierarchical structure. While the model object spans the entire optimisation horizon, a temporal block can represent any fraction within this optimisation window, from minutes up to decades.

Temporal blocks can either represent a continuous time series over a given period, or a disjoint fraction, i.e., representative periods. The supplementary package \textit{SpinePeriods.jl} provides a tool within the Spine environment to determine representative periods [20]. For models with seasonal storage systems, a sequence of representative days is determined to represent the seasonal variations, as described in [21]. Each of the temporal blocks is associated with a resolution parameter, i.e., the duration of the underlying individual time slices. It should be noted that this allows us to entirely decouple the resolution(s) of the model from the resolution of time-dependent parameter values. The temporal block object is used to identify the different resolutions of the decision variables, both operational and planning related. In terms of operational detail, we further differentiate the resolutions of the energy flow and unit-commitment variables.

Through a relationship between a node object and a temporal block object, the temporal resolution of the energy flows in a specific part of the model can be defined. As different nodes can be linked to different temporal blocks, the modeller can choose a high temporal resolution for a particular region of interest, while applying a coarser resolution for other regions. This also holds for different energy vectors, as some commodities might have other time constants. For instance, the fluctuations in power system-related variables might be more volatile and should thus be captured at a higher resolution than those of the gas network. To the best of our knowledge, this flexible temporal differentiation between different sectors and different parts of the model is unique to SpineOpt.

As stated before, not only can the resolution of energy flows be varied, but also through a relationship between units and temporal blocks, the resolution of the unit-commitment variables can be decoupled from the resolution of the energy flow variables, which is inherited from the node. Unit-commitment decision variables are usually of binary or integer type and can therefore significantly increase the computational burden. In practice, by decreasing the temporal resolution of the binary variables, the number of the integer variables in the model is decreased. Depending on the level of detail that is of interest to the modeller, a lower resolution can be selected for the dispatch decisions than for the associated energy flows, providing a trade-off between accuracy and computational complexity.

Temporal blocks are also used to determine the temporal resolution of the investment decisions. All system components, i.e., nodes, units, and connections, can individually have a relationship with temporal blocks to signify the resolution of their investment decision variables. Another useful functionality of the temporal block, that increases the value added by SpineOpt, is its capability to instantiate different resolutions, not only in the spatial, sectoral, or technical dimension, but also in the time dimension. For instance in hydro-thermal scheduling, where long horizons are required to capture the future value of stored water, decreasing the temporal resolution towards the end of the horizon is useful. This way, hourly resolution for the operations in the immediate future can be preserved, while long-term resolution can be reduced (and therefore complexity).

All of these temporal relationships and associated resolutions of the decision variables are evaluated by SpineOpt. The relationships between time slices are automatically determined within the code, i.e., which time slices are within another time slice, which time slices are directly adjacent to each other and which time slices overlap with each other. These relationships between the time slices are then directly accessed within the constraints, to dis-/aggregate variables when different resolutions come together. As an example, the resolution of energy flows in two regions may be different.
When these two regions are coupled, the energy flows of the region with the higher temporal resolution is aggregated for the linking constraint with the region of a lower temporal resolution, using the time slice relationships. This allows the user to swiftly change and evaluate different model setups, finding an applicable resolution for the different parts of the energy model, by only changing the resolution parameter of the corresponding temporal block.

### 3.4. Flexible stochastic structure

As stated before, the representation of uncertainties is gaining more and more importance for energy system models. Scenario-based, stochastic programming is a widely accepted methodology in order to address uncertain parameters in such models, but is not always readily available in open-source implementations, as addressed in Section 2.

To realise this functionality, the stochastic structure and the stochastic scenario objects are introduced. The stochastic scenarios represent different realisations of uncertain parameters. The stochastic structure objects hold the information about the sequence of the stochastic scenarios and their relative weights to each other. These are introduced through relationships between the stochastic structure objects and the scenario objects, in combination with designated parameters.

Similarly to the temporal concept, the stochastic structure provides the required information regarding which parts of the model correspond to which scenarios. This is achieved by using relationships between nodes and stochastic structure objects, allowing for different sequences of stochastic scenarios to be associated with different regions, technologies, or sectors of the model. This enables the user to represent uncertainties accurately where needed, while reducing the amount of scenarios and thus the computational burden where possible. For instance, the uncertainties associated with the power sector might be more pronounced than the uncertainties associated with the gas sector. The stochastic structure of the unit-commitment decisions can also be decoupled from the stochastic structure of the energy flow variables. As the stochastic structure can also be realised as one deterministic scenario, the user can, for instance, model the unit-commitment decisions in a deterministic way, under the expected realisation of uncertain energy flows.

Through the designated relationships with nodes, units, and connections, a stochastic structure can also be associated with the investment decisions. A typical use case for this functionality would again be a two-stage investment problem, where investment decisions are associated with a deterministic structure, while energy flows are modelled with multiple scenarios under a stochastic formulation [22].

Unique to the flexible stochastic concept of SpineOpt is also the way the sequence of stochastic scenarios can be defined. Through relationships between stochastic scenarios and structures, together with relationships between stochastic scenarios, a reduced stochasticity can be modelled. In contrast to a traditional stochastic tree, this flexible stochastic structure allows combining or merging of redundant constraints of certain scenarios, hence reducing the computational complexity. In practice, the modeller might need to generate constraints only for the scenarios with meaningful existing stochastic data. An example of the flexible stochastic structure is illustrated in Figure 3, where two scenarios are merged.

![Figure 3: Example for a realisation of the stochastic tree in SpineOpt.](image)

### 3.5. Model linking and decomposition

In this subsection, the possibilities to link and to decompose models are discussed. Modellers are often interested in linking different models, to combine features, compare results or iterate between them. Significant benefits can be gained from the decomposition of one large model into an equivalent set of smaller problems, in order to increase computational efficiency and maintain tractability.

Linking different models to iterate between tools, or compare results can be challenging as models often rely on different ontologies and programming languages. As SpineOpt is integrated into Spine Toolbox, the communication between other models and SpineOpt is greatly facilitated. Spine Toolbox provides a graphical user interface where models, tools and data can be combined. Moreover, it allows for advanced data import, export and data transformation.
To decompose large problems and reduce the computational burden, SpineOpt implements the Bender’s decomposition method specifically for expansion planning problems. The original, possibly large problem, is divided into a master problem, representing only the investment decisions, and associated sub-problems, representing only the operational constraints. By exploiting the structure of such problems it makes possible to separate them into multiple simpler sub-problems. It should be noted that, even though this specific decomposition addresses a commonly encountered case, the generic data structure of SpineOpt enables users to implement additional decomposition approaches.

Another way to decompose the problem (although not strictly equivalent to the original problem) is the rolling horizon method. Readily available in SpineOpt, this technique can significantly reduce the complexity of optimisation problems with a large time horizon. The method facilitates independently solving the operational problem for a shorter time period (the window), and rolling it forward until the total time period of the optimisation model is considered. Within the rolling horizon technique, the optimal solution of each window is carried forward to the next one.

An important characteristic of SpineOpt’s rolling horizon functionality is its efficient implementation that only creates updates to the mathematical problem if the time-varying parameters are involved in the window’s constraints. Further, SpineOpt’s rolling horizon functionality is highly adaptable, allowing for different look-ahead and roll forward configurations. Moreover, it enables the combination of rolling and varying resolutions (spatial and temporal), as shown in Figure 4.

4. Applications of SpineOpt

4.1. Overview of performed case studies

SpineOpt has been applied to a wide range of case studies, with a specific focus on combining different sectors or markets. Several case studies, from hydro power models of the Nordic region, to pressure-driven gas transfer models, are publicly available. An overview of the different case studies implemented in SpineOpt is given in Figure 5.

The publicly accessible case studies in the Spine project repository at the time of writing include:

- A replication of a gas transmission system with pressure-driven gas transfer [23]. The gas transmission system is linked to the electricity system.
- A simulation of one week of operation of the Skellefte river, which includes fifteen power stations in the Swedish hydro power system [24], and a detailed hydro power model for Sweden and Norway [13]. For this first case study, an elaborate tutorial on getting started with SpineOpt and Spine Toolbox is available in the same source.
- An electricity case study on seasonal storage systems using representative days [25], in combination with SpinePeriods.jl.
- A study on how the flexible coupling of the transport sector can increase the overall efficiency of the energy system [26]. The computational advantage of co-optimising energy sectors using different temporal resolutions in SpineOpt was demonstrated.
- An implementation of a Transmission Expansion Planning (TEP) problem for the power system of the Nordic region, to investigate the optimal planning decisions under operational uncertainty [27].
Table 2

Alternative variations of the model structure for the case study. For this table, the following abbreviations are used: \textit{tvar} – time-varying, \textit{stoch} – stochastic, \textit{det} – deterministic.

<table>
<thead>
<tr>
<th>Gas flow variable</th>
<th>Electricity flow variable</th>
<th>Unit-commitment variable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15m</td>
<td>30m</td>
</tr>
<tr>
<td>Base case</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Case A</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Case B</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Case C</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Case D</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Case E</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Case F</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Case G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case H</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Case I</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Case J</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

4.2. Electricity and gas network case study

In the following, we present the results from a designated case study related to the co-optimised operation of electricity and gas networks. The aim is to demonstrate how the concepts introduced in Section 3 can be leveraged to swiftly generate and evaluate different model setups, and to identify an applicable resolution providing a trade-off between level of detail and computational complexity. The case study uses the data of the IEEE 24-bus test system together with a 12-node natural gas system, as presented in [28] and [29]. The proposed case study spans one day. The electricity grid is modelled using lossless DC power flow equations and the gas grid with a linearised pressure-driven gas flow model. For the gas flow modelling, the non-linear Weymouth equation, that describes the steady-state gas flow between two nodes, is approximated through a Taylor series expansion, around fixed-pressure points, as in [29]. Unit commitment constraints, such as minimum operating points, minimum up- and downtimes, are imposed for the 12 electricity generators (of which 7 are gas fired). Wind and demand variability are taken into account by considering 10 equiprobable scenarios with a data resolution of 15 minutes.

The different variations of the model structure are given in Table 2. For the Base case, all variables are modelled at a quarter-hourly resolution that is consecutively reduced to a half-hourly and hourly resolutions in Cases A and B, respectively. A varying resolution is introduced in Case C, lowering the resolution from 15-min to 1-hour to 3-hours. Cases D and E lower the resolutions of the gas sector and the electricity, respectively. Subsequently, the unit-commitment decision variables are modelled at an hourly resolution (Case F). Starting from the Base case, these alternative model setups can quickly be realised by simply changing only the resolution parameter of the relevant temporal blocks. Modelling a two-stage stochastic optimisation problem, similar to [30], assuming the unit commitment variables as the first stage variables, is represented by Case G. Analysing the impact of the stochastic representation of the two sectors, Cases H and I juxtapose deterministic and scenario-based variables. These different model settings can easily be realised in SpineOpt, by simply adding the relevant relationships with the different stochastic structure objects. Finally, analysing the computational performance of a rolling horizon optimisation, Case J optimises two consecutive days.

4.3. Results and discussion

An overview of the computation performance and the differences in the objective function of the different cases is given in Table 3. As stated here, the total system costs stem almost solely from fuel costs, while the startup costs are rather insignificant. The optimality gap for all cases was set at 0.1%. As Cases G and I did not converge within 36 hours of computational time, the optimality gap was increased to 1% for these two cases. The difference in the optimal value of the objective function varies between -1.22% and 3.33%. While these variations are limited, the computational times differ significantly. In the following, an in-depth analysis of the considered alternatives is presented, elaborating on the specific differences. Figure 6 shows the relative errors in the aggregated portfolio of the electricity and gas sector, respectively, compared to the Base case.

As shown in Table 3, the computational time decreases significantly with a decreased resolution for the Cases A, B, and C. Already cutting the resolution in half, results in a drastic speed-up. At the same time, the generation costs decrease with a coarser resolution. This is partly due to the fact that more renewable generation can be incorporated, as the capacity factor of the wind generators is averaged over coarser resolutions. While renewable generation is curtailed...
Table 3
Performance evaluation of the compared test cases.

<table>
<thead>
<tr>
<th></th>
<th>comp. time [s]</th>
<th>Performance comp. time diff [%]</th>
<th>memory [GiB]</th>
<th>objective [10^{10}\text{€}]</th>
<th>Total costs objective diff [%]</th>
<th>Cost allocation Fuel costs [%] Start-up costs [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base case</td>
<td>44814</td>
<td>-1.555</td>
<td>99.52</td>
<td>0.48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case A</td>
<td>3599</td>
<td>-92</td>
<td>107</td>
<td>1.553</td>
<td>-0.08</td>
<td>99.51</td>
</tr>
<tr>
<td>Case B</td>
<td>811</td>
<td>-98</td>
<td>60</td>
<td>1.549</td>
<td>-0.34</td>
<td>99.52</td>
</tr>
<tr>
<td>Case C</td>
<td>684</td>
<td>-98</td>
<td>64</td>
<td>1.542</td>
<td>-0.83</td>
<td>99.53</td>
</tr>
<tr>
<td>Case D</td>
<td>1126</td>
<td>-97</td>
<td>152</td>
<td>1.560</td>
<td>0.36</td>
<td>99.5</td>
</tr>
<tr>
<td>Case E</td>
<td>25039</td>
<td>-44</td>
<td>133</td>
<td>1.536</td>
<td>-1.22</td>
<td>99.52</td>
</tr>
<tr>
<td>Case F</td>
<td>9887</td>
<td>-78</td>
<td>211</td>
<td>1.559</td>
<td>0.27</td>
<td>99.51</td>
</tr>
<tr>
<td>Case G</td>
<td>877</td>
<td>-98</td>
<td>64</td>
<td>1.576</td>
<td>1.39</td>
<td>99.43</td>
</tr>
<tr>
<td>Case H</td>
<td>26742</td>
<td>-40</td>
<td>119</td>
<td>1.591</td>
<td>2.36</td>
<td>99.55</td>
</tr>
<tr>
<td>Case I</td>
<td>83944</td>
<td>+87</td>
<td>131</td>
<td>1.606</td>
<td>3.33</td>
<td>99.3</td>
</tr>
<tr>
<td>Case J*</td>
<td>68371</td>
<td>+53</td>
<td>298</td>
<td>3.152</td>
<td>5.36</td>
<td>99.53</td>
</tr>
</tbody>
</table>

Figure 6: Aggregated portfolio deviations between all compared cases, (a) for the electricity, and (b) gas production.

in the Base case due to power line congestion or mismatches with the load profiles, Cases A, B and C do not capture these short-term effects completely. While the portfolio deviations due to short-term effects are negligible for Case A, they become more and more pronounced for the coarser resolutions of Cases B and C.

To analyse the impact of varying resolutions across sectors, we reduce the resolution of the gas and the electricity sector into two hours in Cases D and E, respectively. While for Case D the changes in total system costs and in the portfolio are relatively small, the portfolio deviations are more noticeable in Case E, as depicted in Figure 6. This can be explained by the fact that the electricity and the gas sector have different time constants. The main sources of volatility are introduced in the electricity sector i.e., renewable production and a higher dependency on time-dependent demand. With a coarse resolution in the electricity sector, the portfolio contribution of renewable generation is again overestimated (similar to Case B, C) and demand peaks are smoothened. For Case D, the volatility of the electricity sector can still be captured accurately. Moreover, Case D results again in a significant decrease in the computational time with only a limited impact on the objective function. The impact of a reduced resolution on the unit commitment variables is presented in Case F. As shown in Table 3, the reduced resolution of the integer variables results in a significant reduction in computational times, while the system costs only differ slightly from the Base case. The reduced flexibility, due to the fact that units can only be brought online at full hours, results in moderate changes in the portfolio and slight increase of the total costs.

While the above cases describe the impact of varying temporal structures across the model combined with 10 independent scenarios, the flexibility of changing the stochastic structures in parts of the model is explored in Cases G, H, and I. Case G implements a two-stage stochastic problem, where the unit commitment variables represent the first stage (deterministic) variables, while the energy flow decisions represent the second stage variables, i.e., the unit commitment decisions are the same for all scenarios. As these unit commitment decisions have to comply with the feasibility of each scenario, this results in higher costs than assigning unit commitment variables for each scenario individually. The impact of stochasticity on both sectors is analysed in Cases H and I, with deterministic gas and electricity production decisions, respectively. In order to maintain feasibility across all electricity scenarios for Case H, the dispatch decision for the deterministic gas sector are sub-optimal compared to the Base case. This manifests in
portfolio deviations from non-gas-fired units, towards more gas-fired units. To maintain the feasibility of all scenarios of the electricity sector, the deterministic decisions for the gas sector in Case H need to comply with the worst-case scenario, resulting in a noticeable increase in gas production and an increase in total costs. For the deterministic electricity sector (Case I), we observe that the overall production is higher, which can be simply explained by the choice of the deterministic realisation scenario, which does not coincide with the mean of all other scenarios. This also translates into increased total production costs.

Lastly, Case J illustrates SpineOpt’s rolling window decomposition technique. In contrast to the previous cases where one day was optimized, we now optimise two consecutive days, using a rolling horizon, which is why the objective function value of Case J is roughly doubled compared to the Base case. Naturally, as the Base case scenario corresponds to the first sub-problem of Case J, computational times of J are expected to be twofold compared to the Base case. However, thanks to the efficient implementation of the rolling horizon optimisation in SpineOpt, computational times only increase by 53%.

The main conclusion drawn from the simulated alternatives is that a different solution methodology should be chosen depending on the individual characteristics of the considered case study and its aim, accounting for a trade-off between accuracy and computational complexity. In general, sectors with a high variability (e.g., renewable energy sources) are less suited to simplification in terms of their temporal representation and the representation of their uncertainties. Reducing the unit commitment constraints reduces the computational burden and represents a more conservative cost estimate. SpineOpt comprises a flexible structure and a set of features that enable the user to easily assess different setups, according to their needs. It offers a high level of flexibility in manipulating the temporal / spatial resolution, the stochastic structure, in order to alleviate the computational burden of the model, according to the scope and the specific requirements of the case study.

5. Conclusion

In this paper, we have presented the open-source multi-energy system modelling framework SpineOpt. SpineOpt adds to the existing menagerie of energy system models by including a unique degree of modelling choices and flexibility. SpineOpt contains a generic data structure, a flexible spatial and temporal structure, an efficient representation of uncertainties, sector-specific modelling methods, and an implementation of decomposition methods. These features enable the modelling of a variety of energy system models, while providing the means to reduce computational complexity. The capabilities of SpineOpt have been tested and showcased using multiple publicly available case studies. In addition, the article presents a case study that highlights how different modelling trade-offs can affect model performance and accuracy. The case study shows how SpineOpt can be easily tuned for the modelling task at hand. Future development of SpineOpt will focus on improving its sector-specific modelling capabilities and to introducing additional methods to represent different sectors, e.g., AC load flows. Additional extension plans include the support for a flexible mixed-integer and LP representation of the model (able to relax the integer variables for specific parts of the model in an automated manner).

Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N. 774629.

References

SpineOpt: A flexible open-source energy system modelling framework


Energy Systems Integration & Modeling Group (ESIM):

Web-page: [https://www.mech.kuleuven.be/esim](https://www.mech.kuleuven.be/esim)

LinkedIn: [https://www.linkedin.com/company/esimrg](https://www.linkedin.com/company/esimrg)

Latest preprints and accepted publications: [https://www.mech.kuleuven.be/esim-publications](https://www.mech.kuleuven.be/esim-publications)