Time indexes in energy system optimisation: The example of oemof.solph

P. Schönfeldt[1], **M. Hillen**[2,3], **E. Schischke**[4], **U. Krien**[5]

[1]German Aerospace Center (DLR), Institute of Networked Energy Systems, Oldenburg, Germany
[2]German Aerospace Center (DLR), Institute of Solar Research, Jülich, Germany
[3]RWTH Aachen University, Chair of Solar Components, Aachen, Germany
[4]Fraunhofer Institute for Environmental, Safety and Energy Technology UMSICHT, Oberhausen, Germany
[5]Fraunhofer Institute for Manufacturing Technology and Advanced Materials IFAM, Bremen, Germany

Abstract

In energy system modelling, time is typically discretised, i.e. handled in the form of time steps. While the choice of a time grid is a model parameter, it can affect the results of a simulation or optimisation to the same extent as physical parameters. This fact underlines the importance of design decisions by modelling frameworks, as these decisions define the options for modellers.

In `oemof.solph`, a system is implemented that allows a user-defined length for each single time-step. In the present contribution, we cover a detailed description of the status quo of the implementation, including two additional time index features, namely time-series aggregation and pathway planning, along with their experimental implementations. We then discuss options for harmonising the implementations, making them both, more user-friendly and easier to maintain. We also explain the implications of possible design choices on the capability to model the specifics of energy systems and provide examples for modelling and data-analysis with a focus on the use of time indexes.

1 Introduction

In the real world, time is continuous. In a model, however, it is convenient to discretise time. Actually, we do not know of a single example of an optimisation model that allows continuous time [1]. While event-based simulation that works with non-discretised time has been applied for non-optimising models, rare publications naming 'event-based optimisation' in the energy sector (like [2]) still work with time steps. Whereas the choice of a time grid is a model parameter, it can affect the results of a simulation or optimisation to the same extent as physical parameters, e.g. the power of an electricity producer [3, 4].

Despite the fact that almost all models discretise time, terminology related to time indices varies across the literature. This lack of consistency often complicates discussions and leads to misunderstandings, as there is no shared conceptual framework. While constraints and equations have already been reviewed and compared between different implementations [5], details on the discretisation of time are often omitted. The goal of this paper is to clarify the topic, to establish a common language for both new and experienced modellers, and to indicate advantages and disadvantages of different methods. To achieve this, this paper is structured in the following:

In Sec. 2, we describe our understanding of the different contexts in which time indices are applied in energy system modelling and propose an appropriate terminology. As implementation details of the formulation of time have an impact on the way costs are considered, we include a discussion on the definition of operational and capital-related costs. In this context, it should be noted, that also cost specifications have a significant impact on the optimisation results [6]. We then illustrate these concepts in Sec. 3 using examples applying different methods in `oemof.solph` [7]. Finally, we compare and discuss the results in Sec. 4, which also concludes this paper.

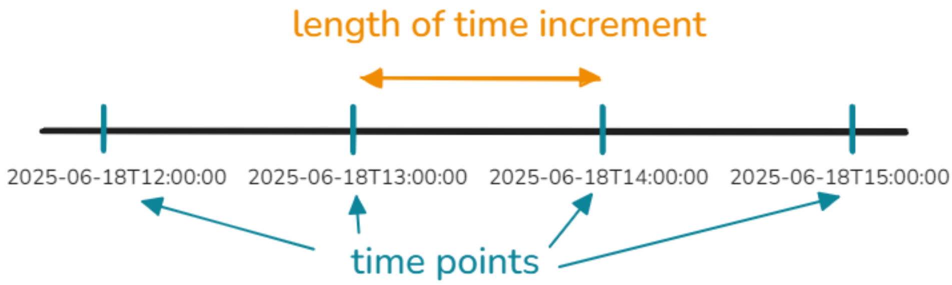

Figure 1: Time index with discretised time steps.

2 Definitions and Conceptual Framework

2.1 Time Index

When interpreting time-discretised information, the fundamental question is: when did something actually happen? Information can be defined either for specific points in time or for time intervals. For example, a temperature measurement timestamp typically implies that the temperature was measured at that particular time. However, a value for precipitation would represent the amount in a time interval, often indexed by the time at which the interval ends. This is referred to as a right-indexed interval. In energy market data, however, the timestamp indicates when the energy price changes to a particular value, and the new price becomes valid after the given time. This is referred to as a left-indexed interval.

The way `oemof.solph` requires the user to provide a time index is a list of points in time

$$\mathfrak{T} = [t_0, t_1, \dots, t_N]. \tag{1}$$

This can be any list of strictly monotonous numbers, that will be interpreted as points in time given in hours after an undefined reference time. Typically, a `pandas.DatetimeIndex` is used, that according to `pandas` nomenclature for a non-leap year has 8761 periods: The index needs to extend from midnight on the first day of the year to midnight on the first day of the following year. As this was not intuitive for some users, the parameter 'infer_last_timeinterval' offers the option of automatically setting the final point. In the code [8], to avoid misunderstandings, we talk about time steps when referring to the discrete nature, about the time increment τ when the individual length of the time step is addressed, and about the time interval if every time step has the same length. A schematic can be seen in Fig. 1. The length of a time increment is calculated from the time index as

$$\tau_m = t_{n+1} - t_n. \tag{2}$$

Note that this definition implies that there are N points in time but only $N-1$ time increments, and that time intervals are left-indexed. To reflect this, we also introduce the counting indexes

$$\mathfrak{N}_{\mathrm{TP}} = [0, 1, \dots, N] \tag{3}$$

for the time points, and

$$\mathfrak{N}_{\mathrm{TS}} = [0, 1, \dots, N-1] \tag{4}$$

for the time steps. For the units, t_n could be, e.g., '2026-02-12T15:00:00', while the τ_m are given in hours (e.g., '$0.25\,\mathrm{h}$'). For variables that change with time, energy-like quantities are defined at points in time while power-like quantities are defined for time increments, i.e., the units could be $[W_n] = 1\,\mathrm{kWh}$, and $[P_m] = 1\,\mathrm{kW}$. This choice allows for values independent of the time resolution.

The definitions in (1) and (2) are very generic. While typically constant time intervals are used (e.g. $\tau_n = 0.25\,\mathrm{h}\ \forall\ n \in \mathfrak{N}_{\mathrm{TS}}$), you can also use non-equidistant time steps. This can be advantageous if you want to capture the actual shape but also want to keep the number of overall variables low to reduce calculation time [9]. A possible step towards reducing the number of time steps without compromising accuracy is to increase the step size during periods of lower relevance. This is called segmentation, and multiple possible approaches exist [10]: Periods of particular relevance can be identified by evaluating the magnitude of the gradient between two consecutive time steps. To this end, a reference variable, such as photovoltaic production, can be employed, as can a combination of several variables, such as photovoltaic production, electricity demand, and heat demand. Another approach is to calculate local variance using a moving window. Beyond mathematical descriptions, specific events based on knowledge of the system can also be utilised. Notable events could include the arrival or departure of an electric vehicle, or an exceptionally high or low residual load. The relevant events can be defined depending on the type of energy system and the research question. This creates a time index with intervals of varying length. Consequently, the final interval must necessarily conclude with a final timestamp. An example can be seen in Fig. 2.

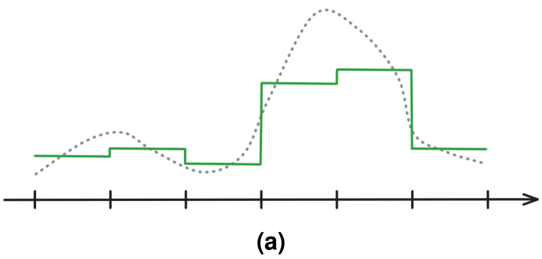

Figure 2: Equidistant (a) and non-equidistant (b) discretisation of continuous information. Note that the latter increases temporal resolution but only locally.

To reduce the number of time steps in a model, also other time series aggregation (TSA) methods than the comprehension of adjacent time steps can be used. For simplicity, consider a situation where time steps are not interlinked. This can be the case in particular if there is no energy storage, and every controllable device is flexibly usable independent from its previous state (esp. no startup time or slow ramps). If this is the case, you just need to decide how many time steps you want to have in total. For example, you might want to have six time steps to represent a whole day. Instead of splitting a day into six $4\,\mathrm{h}$ long increments, you can also choose six $1\,\mathrm{h}$ long increments that each represent multiple hours of the day. This is typically done by clustering similar hours into one representative hour. The representative time increment is usually called (aggregation) period or typical period. As one can see in Fig. 3, the peaks are still adequately represented in the aggregated version, but the chronological sequence of the time series is lost. The fact that one time step after aggregation can represent multiple original time steps is considered by weighting the time steps.

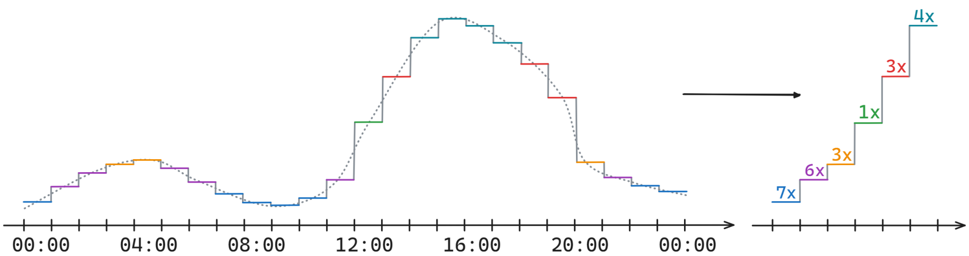

Figure 3: Hourly discretisation and clustering into six aggregation periods (illustrative sketch).

In the previous example, each period consists of only one time step. If the original time steps are not independent, e.g., because a storage system must be considered that might be full or empty depending on previous time steps, this is no longer useful. In that case, aggregated series with a limited set of representative periods, each consisting of multiple time steps (e.g., typical days or weeks), can be used. The periods capture recurring patterns in demand and renewable energy source availability and can be combined with segmentation [11]. In optimisation models, each representative period is weighted in the cost function to reflect how often it occurs in the original data. For clustering-based TSA this weight typically equals the cluster cardinality, i.e., the number of original periods represented by the cluster centroid [12]. For further reading, Hoffmann et al. [13] present a review of different TSA options, including an overview of the terminology and possible synonyms.

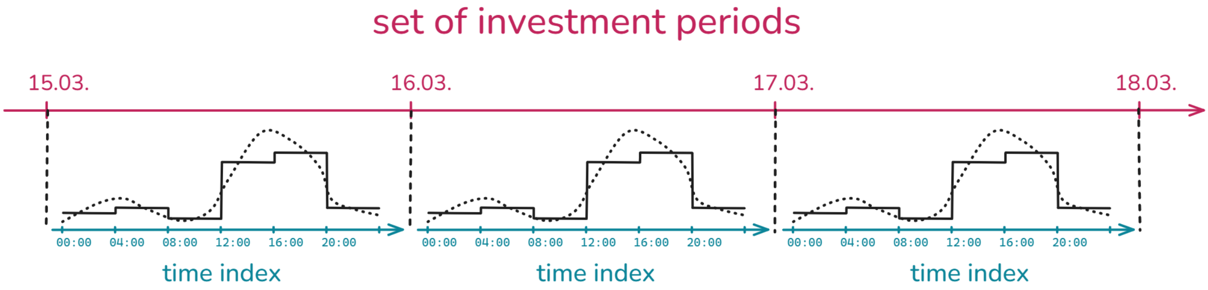

Figure 4: Illustration of the relation between a set of investment periods and a time index for a time series of three days. The set of investment periods consists of three periods of one day, and each day consists of six time steps of four hours. Note that all investment periods are the same here, but could also be different.

A fundamental drawback of standard TSA is the loss of chronological information, because the original sequence of periods is not considered in the optimisation model, which can distort the operation of storage, particularly the behaviour of seasonal storage and other path-dependent components [12]. Renaldi et al. [14] propose multi-resolution formulations to reduce computational runtimes in general optimisation models while preserving seasonal behaviour. These approaches apply different temporal granularities to different system components, using coarser time steps for long-term storage and finer resolutions for short-term operation, with limited accuracy loss in many applications. Gabrielli et al. [15] introduce sequence-aware TSA approaches that preserve temporal coupling by linking representative periods through additional storage balance constraints while maintaining the remaining constraints on the aggregated time layer. Kotzur et al. [12] combine these ideas in a two-layer storage formulation that distinguishes intra-period states (charging/discharging within a typical period) from inter-period states (period boundaries between typical periods). The storage state of charge is represented by a superposition of both; meanwhile, all non-storage components are modelled on the aggregated typical-period layer as usual. This formulation is implemented in `oemof.solph`. It is important to note that the intra-period storage state and the inter-period storage state are of different nature. The intra-period storage state describes the change within a typical period. This delta can take both positive and negative values, reflecting charging and discharging, respectively, and should not be interpreted as a persistent state. In contrast, the inter-period storage state represents the actual storage level at the boundaries between two typical periods, encapsulating the state of charge at the transition points. As ‘intra’ and ‘inter’ can be hard to distinguish, especially in spoken language, there have been discussions in the oemof community about clearer naming. For the time being, however, we will stick to the original naming convention.

Lastly, there is another time scale to consider in energy system modelling. It is needed if you also want to optimise the point in time an investment is made, taking into account the transformation of an energy system and evolving system conditions. This is called dynamic investment [16, 17], pathway planning[18], multi-year planning [19], multi-stage investment [20], or multi-period investment planning [21]. To implement this, a new set of time increments must be introduced that are linked to the times for which each investment is available. For example, a 20-year time horizon can be divided into four 5-year increments. The capacities of the units can then differ for each 5-year time increment. In `oemof.solph`, this set is historically called ‘periods’, but it will be renamed ’investment periods’ or ’capacity periods’ to avoid confusion with the aggregation periods discussed previously. One argument for `investment_periods` is that PyPSA also uses this term [18]. The latter name simply communicates the fact that capacities are defined within these periods. In this way, `capacity_periods` might more clearly indicate that capacities can also change without investment optimisation, for example if the fadeout of a technology is modelled. For the time being, we will use the terms as synonyms.

Each investment period can, in turn, consist of the time index variations discussed above. In our example, the 20 years could be divided into four 5-year investment periods. Each 5-year investment period could be represented by a typical year (using time series aggregation) in an equidistant hourly resolution. You can also imagine dividing each 5-year investment period into a monthly equidistant resolution, resulting in $12 \times 5 = 60$ time increments in each investment period and $4 \times 60 = 240$ time increments in total.

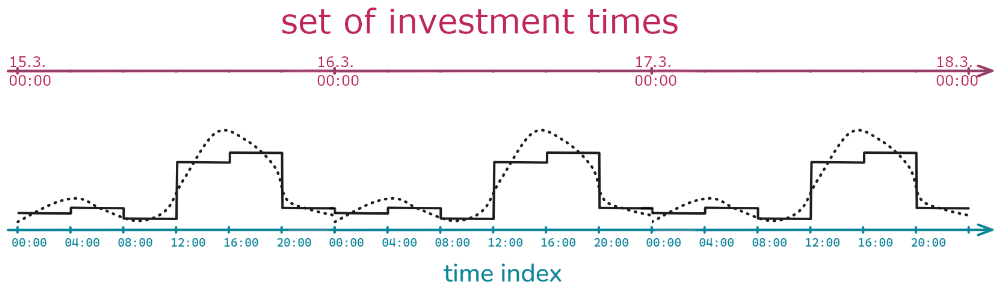

Figure 5: Illustration of the specification of investment times on a continuous time axis. If the multiple time indexes in Fig. 4 are aligned, both formulations are equivalent.

There are multiple ways in which capacity periods can be defined. In particular, they can be seen as isolated time horizons, possibly without continuity between them. In that case, a list of multiple time indexes, as defined in (1), can be given to define a new multi-period index, as illustrated in Fig 4

$$\mathfrak{T}_{\text{MP}} = [[t_{0,0}, t_{0,1}, \ldots, t_{0,N}], \ldots [t_{P,0}, t_{P,1}, \ldots, t_{P,N}]] \tag{5}$$

with P being the last element in the set of the investment period indexes

$$\mathfrak{N}_{\text{invest}} = [0, 1, \ldots, P]. \tag{6}$$

This is the way PyPSA handles multiple periods [18] and the manner in which the original implementation draft in `solph 0.5` defined periods.

In `solph 0.5`, it was possible to use the argument `time_increment` directly define the time increments τ_m. To have full time stamps in the results, additionally the `time_index` as defined in (1), has to be given as a concatenation of all defined indexes from (1)

$$\mathfrak{T}_{\text{MP,solph 0.5}} = [t_{0,0}, t_{0,1}, \ldots, t_{0,N-1}, t_{1,0}, \ldots].$$

Note that according to (2) for example the time index $[\ldots$, 2025/12/31 23:00, 2026/01/01 00:00, 2030/01/01 00:00, $\ldots]$ would produce a time step of four years and one hour, while it is meant to represent a time step of one hour plus a gap of four years. Thus, using `time_increment` was crucial to overwrite the calculated values. This argument, however, was removed with `solph 0.6` to guarantee consistency between the time indexes. A current way to implement gaps will be shown in Section 3.3.

A different way to provide the capacity periods assumes continuous time, as defined in (1). Then, a set of 'investment time points' $t_{\text{invest},i}$ can be established to define points in time when the capacity might change. This is illustrated in Fig 5. So, if two time steps lie in the same capacity period, a capacity C is the same for both time steps, or mathematically, when without loss of generality $t_m < t_n$,

$$\nexists\, t_{\text{invest},i} \text{ with } t_m < t_{\text{invest},i} \leq t_n \quad \Rightarrow \quad C(t_m) = C(t_n) \tag{7}$$

To allow defining costs per investment period as an annuity, also the end of the period needs to be given, so

$$\mathfrak{T}_{\text{invest}} = [t_{\text{invest},0}, \ldots, t_{\text{invest},P+1}], \tag{8}$$

which has one more entry compared to $\mathfrak{N}_{\text{invest}}$. Note that this definition allows investment times to be freely chosen, i.e. they do not have to be part of the time index $\mathfrak{T}$. In particular, this definition allows to give an investment annuity while only optimising operation for shorter times, e.g. a single week. To allow upfront invest without explicitly defining investment times, a default could be $[t_{\text{invest},0}, t_{\text{invest},1}] \overset{\text{upfront}}{=} [t_0, t_N]$, where t_0 is the first and t_N is the last time point of the time index $\mathfrak{T}$ defined in (1). Also note that $\mathfrak{T}_{\text{MP}}$ as defined in (5) contains all information defining the time. Thus, it can replace the original time index $\mathfrak{T}$. In contrast, $\mathfrak{T}_{\text{invest}}$ just adds the information when capacities might change and is used in combination with $\mathfrak{T}$.

Taking into account the discussed implementation details and the ambiguity of the term 'period', we prefer the term 'pathway planning' to the term 'multi-period investment', which was originally used in solph since before time-series aggregation was implemented. Finally, it should be noted that capacities exist over periods of time. Thus, the corresponding counting index $\mathfrak{N}_{\text{invest}}$ follows the definition of (6) regardless of whether the definition (5) or (8) is used for the capacity periods.

2.2 Costs

In economic terms, costs are usually categorised as either capital-related costs (CAPEX) or operational costs (OPEX). For `solph`, they are typically defined using the parameter `variable_costs` of a `Flow` and/or by giving an `Investment` object and its associated costs as the `nominal_capacity` of any class (e.g. `GenericStorage` or `Flow`). Other types, such as `storage_costs` for the costs of using a storage or `activity_costs` for binary costs of using a flow, will not be discussed here.

OPEX are composed of contributions that are

- variable (e.g. energy costs in $€/\text{kWh}$),
- capacity related per time (e.g. maintenance in $€/\text{kW}/\text{a}$ for Flows or in $€/\text{kWh}/\text{a}$ for storage units), or
- fixed per time (e.g. chimney sweeper in $€/\text{a}$).

CAPEX are expressed using

- capacity related contributions (e.g. component costs in $€/\text{kW}$ for Flows or in $€/\text{kWh}$ for storage units), and
- an offset (e.g. cost to start construction in €).

When CAPEX are given per time, e.g. in order to reflect the time scale of the model rather than the lifetime of the units, distinguishing them from the OPEX is not a strict mathematical necessary.

The `variable_costs` in `solph` correspond to the variable OPEX. They are implemented generically and the software takes care of the proper handling of the time index. The total variable costs $c_{\text{var,tot}}$ are calculated by summing over the products of all variable costs c_{var} multiplied by the corresponding time increments τ and the values P of the corresponding Flows

$$c_{\text{var,tot}} = \sum_{(i,o)\in\mathfrak{F}} \sum_{t\in\mathfrak{N}_{\text{TS}}} \tau_t \times P_{i,o,t} \times c_{\text{var},i,o,t}, \tag{9}$$

where $\mathfrak{F}$ is the set of Flows, indexed by the input and the output Node it connects.

Investment optimisation is implemented by providing an `Investment` object for the `nominal_capacity` attribute of any `solph` class. In contrast to the case of variable costs, there are multiple implementation options for investment optimisation. This is due to the fact that the time horizon of the optimisation is typically smaller than the lifetime of the units the model can invest in. If just upfront invest is considered (i.e. investing at the beginning of the optimisation time horizon for the complete horizon), typically (cf. [18]), the optimised capacities C are multiplied with the corresponding capacity related costs c_{invest} (called `ep_costs` for equivalent periodical costs), already scaled to the time horizon of the optimisation

$$c_{\text{invest,upfront}} = \sum_{(i,o)\in\mathfrak{F}} C_{i,o} \times c_{\text{invest},i,o}. \tag{10a}$$

If the time horizon is a year, an annuity is given as `ep_cost`, where the units of $c_{\text{invest},i,o}$ could be for example $€/\text{kW}/\text{a}$. This is how upfront invest optimisation is implemented in `solph`. As noted above, the annualised offset of an investment and the fixed OPEX typically have the same units ($€/\text{kW}/\text{a}$). Thus, in upfront investment models using `solph` both are currently given using the parameter `offset` of the `Investment`.

A way to handle investments more directly instead, is lifetime tracking of the units invested into. This way, total absolute costs and the remaining value are used. The method was implemented for multi-period investment in `solph v0.5`, but is also used e.g. in the optimisation framework REMIX [22]. An advantage is that the absolute value of the costs (e.g. in $€/\text{kW}$ and € for an offset) can be used without calculating annuities, making the model more intuitive. The costs are then attributed to the point in time the investment is made

$$c_{\text{invest,tot}} = \sum_{(i,o)\in\mathfrak{F}} \sum_{p\in\mathfrak{N}_{\text{invest}}} C_{i,o,p} \times c_{\text{invest},i,o,p}, \tag{10b}$$

where the costs $c_{\mathrm{invest},i,o,p}$ are discounted depending on the investment period. The capacity is tracked using

$$C_p = C_{p-1} + C_{\text{invest}}(p) - C_{\text{decommission}}(p) \quad \forall\, p \in \mathfrak{N}_{\mathrm{invest}}. \tag{11}$$

At the end of the time horizon, a remaining value can be subtracted from the original costs. So only the share of investment costs incurred within the time horizon is added to the objective value.

For multi-period investment, also fixed OPEX are considered. This is done using the keyword `fixed_costs`, which accepts one value per period. Capacity related OPEX, however, have to be scaled to the lifetime of the investment to treat them as if they were part of the investment itself as part of the `ep_costs`.

With version 0.5.1, `omeof.solph` introduced internal handling of a global `discount_rate` and a local `interest_rate` that could be set per component, both aligned with VDI 2067 [23]. This way, the capacity related costs could be given in absolute values and were automatically adjusted for future years. However, the implementation was based on narrowing assumptions – e.g. time horizons would be multiples of full years and discount rates and interest rates would not change over time. Also, literature values for future energy prices are sometimes already discounted, requiring the user to undo it as there is an internal scaling. Thus, and to generally allow more flexibility in the pre-processing, we started removing the discounting functionality with `solph` version 0.6.0. Further, while writing this paper, we found that capacity related CAPEX were discounted in the pathway planning optimisation but the offset never was. As discussed in Sec. 2.1, also manually setting the time increments τ_m is no longer possible, effectively removing the previous (partial) support for gaps between investment periods that is implied by (5).

The plan is to implement a cost handling that allows more flexibility and is easier to maintain. To achieve this, capacity related costs will always be given as an annuity, as it is also done in PyPSA [18]. For users that are interested in upfront invest for one year, nothing would change. This is a compelling reason to choose this path for harmonising upfront and pathway investment: The functionality that is marked as stable and used more often can remain unchanged. Internally, however, (10a) would become

$$c_{\mathrm{invest,upfront}} = (t_N - t_0) \times \sum_{(i,o)\in\mathfrak{F}} C_{i,o} \times c_{\mathrm{invest},i,o}, \tag{12a}$$

where $t_N - t_0$ is the time horizon defined by the time index (1). This is just a special case of the (intended) pathway investment optimisation formulation

$$c_{\mathrm{invest,tot}} = \sum_{(i,o)\in\mathfrak{F}} \sum_{p\in\mathfrak{N}_{\mathrm{invest}}} (t_{\text{invest},p+1} - t_{\text{invest},p}) \times C_{i,o,p} \times c_{\mathrm{invest},i,o,p}. \tag{12b}$$

In contrast to the formulation of (10b), that considered total costs at the time of investment, discounted periodical costs are defined over of the lifetime. Note that $c_{\mathrm{invest},i,o}$ also need to be adjusted when the time horizon is changed when just upfront investment is optimised, as the discounted periodical costs depend on time. An advantage of this formulation is, that investment optimisation has the same form as (9), and both, discounting and interest rate, can be handled individually by the user. However, lifetime tracking has to be implemented using additional constraints, either using (11) or – if many units are considered – analogue to ramping and uptime constraints for the `Flows`.

Consequently, especially when defaulting to annuities, specific and clear keywords can and should be used to describe the investment. Currently, the parameter `ep_costs` expects either an annuity (upfront investment) or absolute costs (pathway optimisation), which can be confusing. We also suggest introducing separate keywords for the different types of costs: In the optimisation, it makes sense to treat OPEX and CAPEX as one, just considering if it is capacity-related or fixed. However, if the user provides both CAPEX and OPEX separately, it becomes possible to make a corresponding differentiation in post-processing.

3 Application

To make the definitions above more vivid, we give examples based on a PV-house energy system based on real-world measurements from a publicly available high-resolution dataset [24]. As an existing PV system could already incentivise the inhabitants to shift their demand, we take the electricity demand of a building without a PV system (building 27 from the dataset). We then combine this with a scaled curve from the south-facing PV system of $14.5\,\mathrm{kW}$ nominal power, that is distributed with the same dataset. An excerpt of the curves can be seen in Fig. 6. Here, we scaled the PV curve to $800\,\mathrm{W}$ to simulate a balcony (solar) power plant.

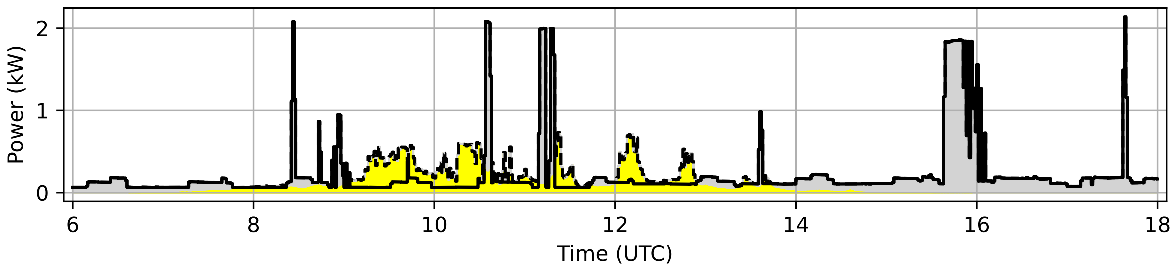

Figure 6: Electricity time series of the 3rd of November. The solid line marks the electricity demand, the dashed lines symbolises feed-in. Grey area stands for electricity coming from the grid, yellow area is local solar generated electricity.

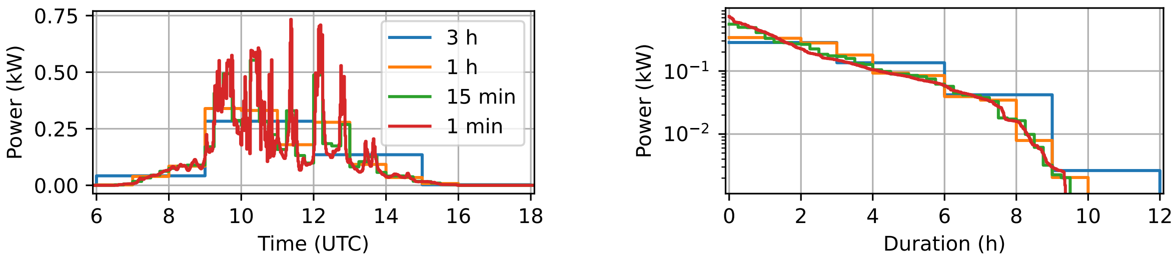

Figure 7: PV generation time series of the 3rd of November resampled with different resolutions (left) and corresponding duration curves (right). The duration curve is displayed using a logarithmic scale to highlight the time when it approaches zero.

The original data has a resoltuion of 1 min. The effect of resampling the generation curve can be seen in Fig. 7. Looking at the duration curve, it can be seen that coarser time resolutions systematically underestimate peak loads. As the total energy under the curve is preserved by averaging, however, they overestimate small loads. In particular, the duration curve extends to higher number of hours, indicating production when there was actually none. The error introduced by reduced time resolution for PV systems is well studied and can reach more then $10\,\%$ for simulated battery usage when an hourly resolution is applied [25]. In [26] a minimum resolution of $5\,\mathrm{min}$ is proposed for sizing of the battery inverter power of a storage system. As discussed in Sec. 2.1, the time resolution does not directly imply a number of required time steps: As there is only production between 06:00 and 16:00, e.g. 11 time steps are sufficient for the hourly resolution (10 for the day, one for the night).

To illustrate the concept of time indices in energy system models introduced in the previous section, we present an example implemented using the `oemof.solph` framework. The complete example is available in the documentation of `oemof.solph` under advanced tutorials[1].

The modelled energy system represents a single household and is parametrised with the data introduced above. We decided for a singe one-person household as it is expected to be sensitive to the aggregation methods we apply. The electricity sector is centred on an electric bus that can be supplied by the public grid and—if selected as an investment option—by a photovoltaic (PV) system. PV generation may be used to directly meet household electricity demand and EV charging demand, to charge a battery storage system, or to be exported to the grid in case of surplus generation. Conversely, residual electricity deficits are covered through imports from the public grid.

[1] https://oemof-solph.readthedocs.io/en/stable/index.html

The heat sector is represented by a heat bus supplying the household heat demand. Heat is provided by an electric heat pump connected to the electric bus; alternatively, a gas boiler may supply heat using natural gas drawn from the gas grid via a gas bus. Heat demand data is calculated based on the temperature time series that is also given in the provided data set. It is less volatile than the electricity time series, also reflecting the inertia typically inherent to the heating sector. Figure 8 depicts the system structure and energy flows. Components shown with solid outlines are assumed to be exogenously given and available in all model runs, whereas dashed outlines indicate technologies that are modelled as optional investments and may be endogenously selected by the optimisation.

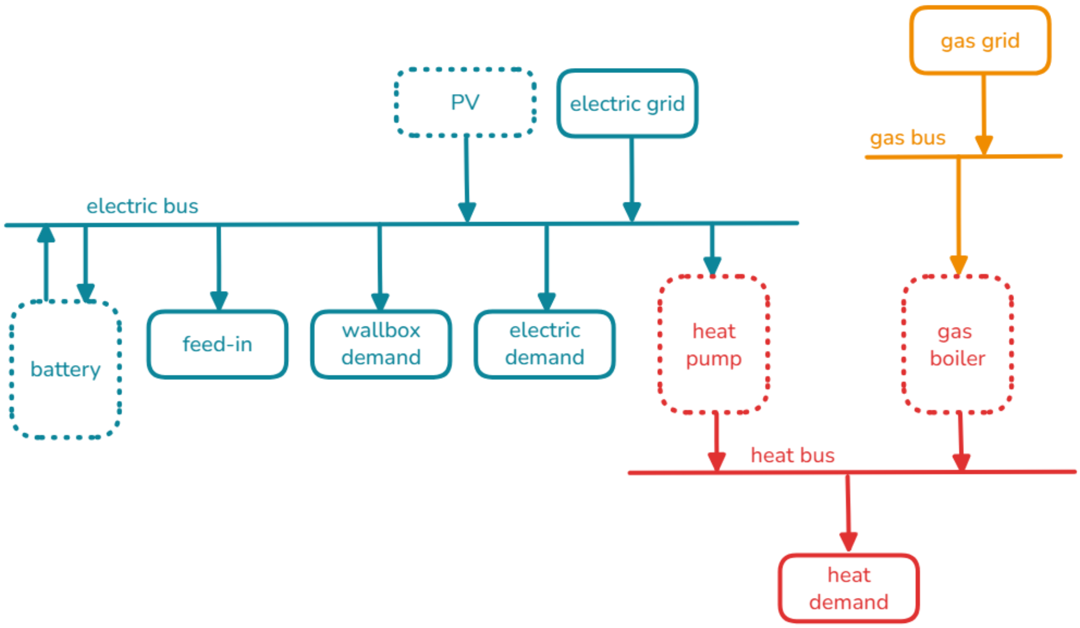

Figure 8: Example household energy system implemented in `oemof.solph`, illustrating the system structure and available technology options for different time representation approaches. Components shown with solid outlines are assumed to be exogenously given, while dashed outlines indicate technologies that are modeled as optional investment decisions.

This example is used to demonstrate the previously discussed options for handling time representation in energy system models. First, we illustrate an equidistant discretisation of the time index, followed by a non-equidistant discretisation. Next, time series aggregation is applied using the Python package TSAM [27]. Finally, we present a pathway investment optimisation model with a slightly modified set of components to highlight the different time scales addressed in such models.

The costs for all components, as well as for energy carriers, are taken from [28]. The efficiency of the heat pump is calculated using the ambient temperature.

3.1 Equidistant and Non-Equidistant Discretisation

Equidistant discretisation of time can be considered the simplest method. In principle, arbitrarily small time intervals are possible, as described in Sec. 2.1, where small intervals are necessary to achieve sufficient accuracy during periods of highly fluctuating feed-in. However, uniform intervals also lead to refinement of the time index also during periods of low activity. In the present example, the photovoltaic system generates no electricity at night, so there is no volatility to be covered by the additional time steps. Nevertheless, calculations are performed using the same time interval during these hours. When optimising an entire year, in our example, the computational time increases by a factor of 100 if the interval is refined from 10-minute to 1-minute values. A full overview also containing non-equidistant discretisation can be seen in Fig. 9 and Fig. 10.

For the non-equidistant discretisation, we anticipate that photovoltaic feed-in would be the relevant variable. Thus, the nocturnal period is consolidated into two intervals, irrespective of the time step sizes

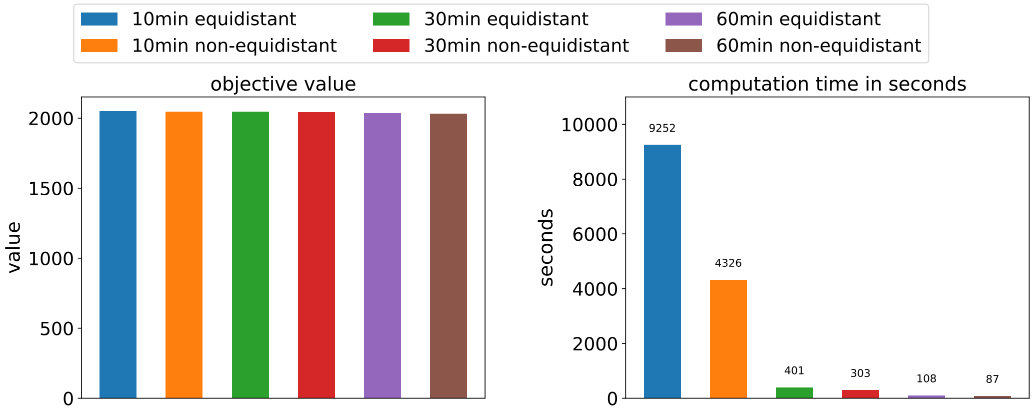

Figure 9: The objective value and computation time for a model with different time increment lengths for an equidistant and non-equidistant time index. For the non-equidistant time index, the night-time time increments remain fixed at four hours, from 21:00 to 01:00 and from 01:00 to 05:00.

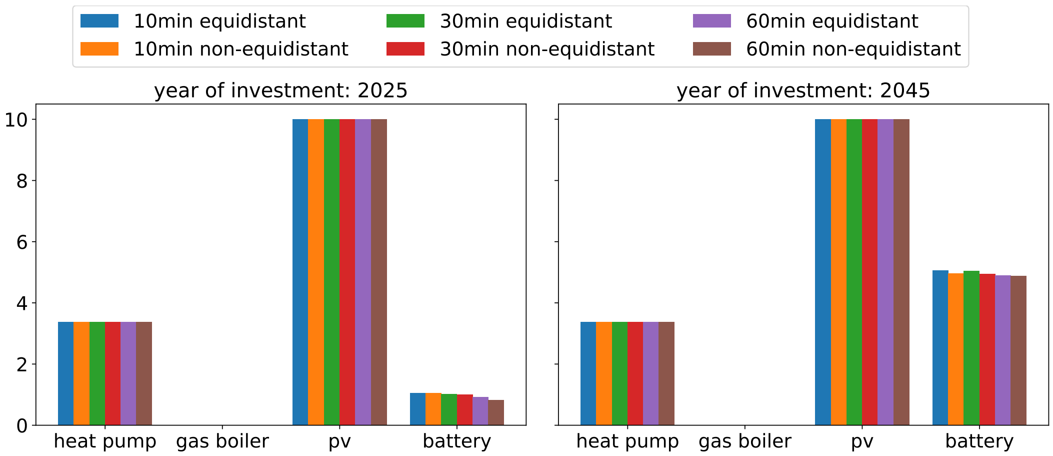

Figure 10: Installed capacities are calculated using different time increment lengths for an equidistant and a non-equidistant time index. For the non-equidistant time index, the night-time time increments remain fixed at four hours, from 21:00 to 01:00 and from 01:00 to 05:00.

employed at other times. The nocturnal time steps extend from 21:00 to 01:00 and from 01:00 to 05:00, each comprising four hours. Although this is a very basic form of segmentation, it already creates a time index with intervals of varying length.

Figure 9 shows that the computation time is reduced by more than 50%, particularly for smaller time steps when very large time steps are chosen during night-time periods. During this time, the photovoltaic (PV) system does not generate electricity. The results are practically identical compared with a uniform time index. Figure 10 illustrates this, showing the installed capacities of the PV system, gas boiler, heat pump and battery. However, it can be seen that the battery capacity decreases slightly with long time increments. While the difference is very small, it indicates that storage capacities may be underestimated when using large time steps or coarsely resolved data. The PV system is so beneficial that it has been expanded to reach the imposed capacity limit. The gas boiler is too expensive and the heat pump is sized to fully cover the heat demand. Consequently, only the battery reacts sensitively to the different model variants.

3.2 Time Series Aggregation

TSA requires the prior clustering of the high-resolution input time series into a reduced set of representative periods. For this pre-processing step, we use the Python package TSAM [27]. In TSAM, the most influential modelling choice is the definition of the representative periods, in particular, the number of typical periods and the length of a typical period. These parameters directly control the trade-off between computational tractability and the ability to reproduce intra-annual variability, especially seasonal dynamics relevant for storage operation. A second key decision concerns the clustering algorithm and the representation of each cluster. TSAM commonly applies hierarchical, k-medoids or k-means clustering; medoid-based representations preserve physically plausible period shapes compared to mean profiles, which can smooth peaks and thereby bias sizing and dispatch. Finally, the treatment of extreme events is critical for robustness. TSAM allows the explicit inclusion of extreme periods (e.g., peak demand or minimum temperature) as additional representative periods, mitigating the risk that rare but system-defining events are underrepresented by purely statistical clustering.

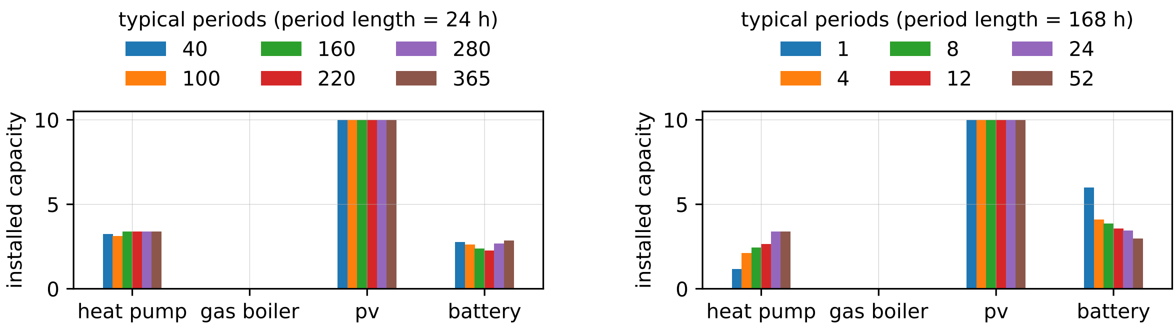

Figure 11: Installed capacities for different numbers of typical periods using daily (24 h) and weekly (168 h) typical-period formulations. PV, heat pump, and gas boiler capacities are given in kW, while battery capacity is given in kWh.

In the present example, we start with a fixed hourly resolution. Figure 11 summarises how the optimal installed capacities change with the temporal aggregation settings. The left panel reports results for daily typical periods (24 h), and the right panel for weekly typical periods (168 h). In both cases, increasing the number of typical periods generally reduces aggregation-induced bias and leads to capacities that stabilise as the representation of intra-annual variability improves. All results were generated using TSAM with k-means clustering; no extreme periods were enforced. Accordingly, deviations at low numbers of typical periods can be interpreted as effects of smoothing and the potential under-representation of production and demand peaks.

3.3 Pathway Planning

The pathway optimisation model examines the development of the energy system over the time horizon from 2025 to 2045. Investment decisions can be made every five years, and existing assets are decommissioned when they reach the end of their lifetime. The model begins in 2025 with a minimal configuration consisting of a gas boiler and a connection to the electricity grid. It can decide which components from the example energy system – such as PV panels, battery storage, or a heat pump – to install and at which point in time. The gas boiler reaches the end of its lifetime in 2035, requiring the model to select a new heating source, by investing in a new boiler or switching to a heat pump.

For the time series representation, each 5 year investment period is represented by one year:

$$\begin{aligned}\mathfrak{T}_{\text{MP,original}} = [&[\text{2025/01/01 00:00}, \dots, \text{2026/01/01 00:00}],\\ &[\text{2030/01/01 00:00}, \dots, \text{2031/01/01 00:00}],\\ &[\text{2035/01/01 00:00}, \dots, \text{2036/01/01 00:00}],\\ &[\text{2040/01/01 00:00}, \dots, \text{2041/01/01 00:00}],\\ &[\text{2045/01/01 00:00}, \dots, \text{2046/01/01 00:00}]]\end{aligned}$$

Each of these years is modelled using aggregated time series generated with TSAM, as described in Section 3.2. These aggregated series consist of 100 typical periods, each covering 24 hours. For simplicity, we assume that energy demand and PV potential remain constant over the entire planning horizon. This includes shifting days of the week and omitting leap years, so that the same time series can be applied from 2025 to 2045.

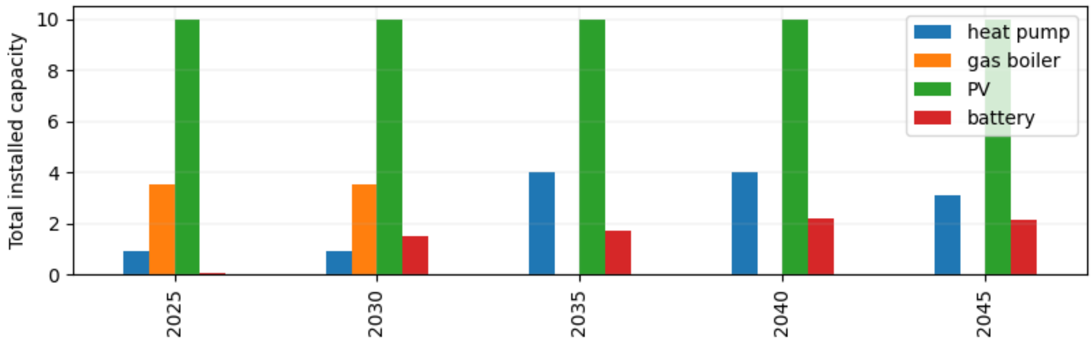

Figure 12: Total installed capacity of each component for each investment period. PV, heat pump, and gas boiler capacities are given in kW, while battery capacity is given in kWh.

To use this with the continuous time axis of `solph 0.6`, we create a modified time index. In that index, 2030 becomes 2026, 3035 becomes 2027, and so on. The last time step of every period but the last one is neglected. Since investment decisions occur only every five years, annuity calculations must reflect this interval. Consequently, both, the discount rate

$$i_{\text{adjusted}} = (1 + i)^5 - 1 \tag{13a}$$

and component lifetimes

$$t_{\text{life,adjusted}} = \frac{t_{\text{life}}}{5}, \tag{13b}$$

have to be adjusted. These adjustments were done automatically in the background until before `solph 0.6`. In the currently implemented example (January 2026), the discount rate is not adjusted, however, because `solph` does not allow discount rates that change with time. This functionality will probably never be implemented, as it will be made obsolete when changing to the cost definition of(12), that directly works with time-dependent annuities.

As shown in the results in Fig. 12, initially a hybrid solution is optimal, that adds small heat pump to the existing gas boiler. As soon as the gas boiler reaches the end of its lifetime it is replaced completely by a heat pump system. The battery capacity increases as investment costs decrease, while the PV-system is already economically viable to install at the beginning and is immediately replaced at the end of its lifetime.

3.4 Comparison of the methods

For a better comparison of the methods, the resulting capacities are shown in Fig. 13. These results cannot be directly compared to the ones of the pathway planning, in particular because that model initially has a gas boiler. However, the tendencies are the same across all examples: When it comes to investments, the heat pump is preferred over the gas boiler, and the chosen PV capacity is always the maximum allowed. As already in Sec. 3.1, and Sec. 3.2, in our example the battery investment is most sensitive to the aggregation method. For the year 2045, however, the results align. This might be due to the aggregation itself or because of the different parametrisation of the costs: As discussed before, models are also very sensitive on specifics, e.g. of discounting, and as of now, there is no complete consistency between the implementations in `solph`. While the size of the battery is significantly larger, the objective value as a measure for the total costs, does not deviate to the same extend. This is shown in Fig. 14. In total, the example illustrates the possible trade-off between model accuracy and computational time, but also that with a good choice of aggregation methods numerical results might be almost uninfluenced.

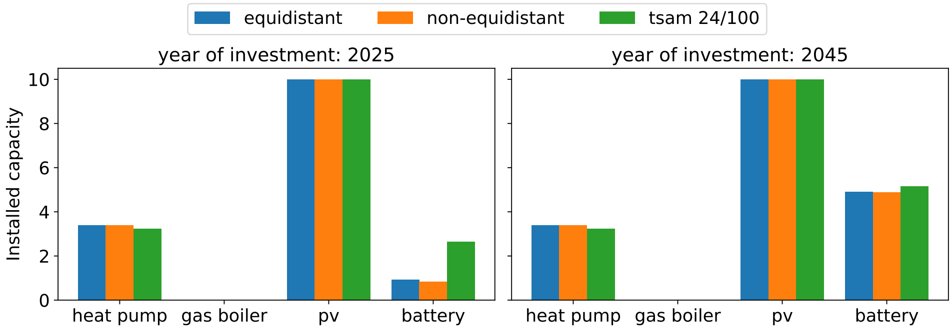

Figure 13: Total installed capacity of each component for two different years of investment. Heat pump, gas boiler and pv capacities are given in kW, while battery capacity is given in kWh.

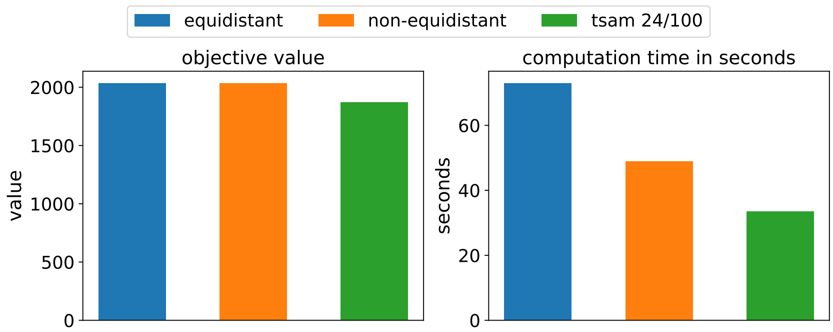

Figure 14: The objective value and the computation time for the upfront investment case of 2025.

4 Conclusion

Time discretisation is a fundamental modelling choice in energy system optimisation and can influence results as strongly as key techno-economic parameters. This paper has clarified the terminology and conceptual distinctions required to discuss time indices consistently, with particular emphasis on (i) the interpretation of timestamps as points in time versus intervals, (ii) the role of time increments in ensuring resolution-independent parametrisation, and (iii) the separation of operational and investment-related time scales. Using `oemof.solph` as a concrete reference, we have shown how these concepts translate into a flexible implementation that supports equidistant and non-equidistant time steps, time-series aggregation, and pathway planning.

The application examples highlight that no single time representation is universally optimal. Equidistant discretisation is straightforward but can become computationally prohibitive at high resolutions, while non-equidistant time steps can substantially reduce model size by refining the grid only where dynamics are relevant. Time-series aggregation enables further reductions but introduces the well-known risk of losing chronological information, which is particularly critical for storage and other path-dependent components. The two-layer storage formulation implemented in `oemof.solph` addresses this limitation by separating intra-period storage deltas from inter-period boundary states, thereby improving the representation of seasonal behaviour while retaining computational benefits. Finally, pathway planning introduces an additional temporal dimension that is essential for analysing transition dynamics and investment timing, but it also increases the need for transparent and harmonised definitions of periods, costs, and discounting assumptions.

Overall, the presented framework and terminology provide a basis for clearer communication between modellers and for more robust model design decisions. Future development in `oemof.solph` should prioritise harmonising the interfaces for time-series aggregation and pathway planning, reducing ambiguity in naming, and establishing a consistent cost representation—preferably based on annuities—to improve usability and maintainability. These steps would strengthen the capability of modelling frameworks to support both operational and transformational research questions without sacrificing transparency or reproducibility.

Declarations

Acknowledgements

Uwe Krien gratefully acknowledges funding by the Federal Ministry of Education and Research (Bundesministerium für Wirtschaft und Energie, BMWE) within the project *open_plan 2.0* (grant number 03EI1087C). We would also like to thank the oemof community for insightful discussions, particularly regarding time series aggregation and cost definitions. Special thanks go to Johannes Kochems for his feedback on the manuscript.

License

AI-assisted technologies

During the preparation of this manuscript, the authors used AI-assisted tools (ChatGPT, DeepL, and Overleaf's AI features) to support language editing and improve clarity of presentation. The authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] L. Laveneziana, M. Prussi, and D. Chiaramonti, "Critical review of energy planning models for the sustainable development at company level," *Energy Strategy Reviews*, vol. 49, p. 101136, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211467X2300086X

[2] J. Li, B. Liu, X. Wang, and H. Ma, "Event-based optimization for smoothing ramp events in combined wind-storage energy systems," *International Journal of Electrical Power & Energy Systems*, vol. 158, p. 109959, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061524001807

[3] V. M. Kiss, Z. Hetesi, and T. Kiss, "The effect of time resolution on energy system simulation in case of intermittent energies," *Renewable and Sustainable Energy Reviews*, vol. 191, p. 114099, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032123009577

[4] M. A. C. Hoffmann, "Temporal aggregation methods for energy system modeling; [überarbeitete Auflage]," Dissertation, RWTH Aachen University, Jülich, 2023. [Online]. Available: https://publications.rwth-aachen.de/record/961572

[5] M. Hoffmann, B. U. Schyska, J. Bartels, T. Pelser, J. Behrens, M. Wetzel, H. C. Gils, C.-F. Tang, M. Tillmanns, J. Stock, A. Xhonneux, L. Kotzur, A. Praktiknjo, T. Vogt, P. Jochem, J. Linßen, J. M. Weinand, and D. Stolten, "A review of mixed-integer linear formulations for framework-based energy system models," *Advances in Applied Energy*, vol. 16, p. 100190, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666792424000283

[6] M. Mier and V. Azarova, "Investment cost specifications revisited," *Energy Policy*, vol. 188, p. 114058, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421524000788

[7] U. Krien, P. Schönfeldt, J. Launer, S. Hilpert, C. Kaldemeyer, and G. Pleßmann, "oemof.solph—a model generator for linear and mixed-integer linear optimisation of energy systems," *Software Impacts*, vol. 6, p. 100028, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665963820300191

[8] U. Krien, C. Kaldemeyer, S. Günther, P. Schönfeldt, H. Simon, J. Launer, J. Röder, C. Möller, J. Kochems, H. Huyskens, A. Developer, B. Schachler, F. Pl, S. Sayadi, P.-F. Duc, J. Endres, F. Büllesbach, D. Fuhrländer, A. Developer, F. Witte, P. Kassing, E. Zolotarevskaia, S. Berendes, B. Lancien, A. Developer, A. Developer, A. Developer, L. Schürmann, A. Developer, J.-O. Delfs, A. Developer, A. Developer, T. Smalla, A. Developer, J. Wolf, A. Developer, E. Gaudchau, A. Developer, T. Rohrer, M.-C. Gering, E. Schischke, M. Fritz, and J. Freißmann, "oemof.solph," Nov. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.17633236

[9] P. de Guibert, B. Shirizadeh, and P. Quirion, "Variable time-step: A method for improving computational tractability for energy system models with long-term storage," *Energy*, vol. 213, p. 119024, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544220321319

[10] M. Lovrić, M. Milanović, and M. Stamenković, "Algoritmic methods for segmentation of time series: An overview," *Journal of Contemporary Economic and Business Issues*, vol. 1, no. 1, pp. 31–53, 2014. [Online]. Available: https://hdl.handle.net/10419/147468

[11] S. Fazlollahi, S. L. Bungener, P. Mandel, G. Becker, and F. Maréchal, "Multi-objectives, multi-period optimization of district energy systems: I. selection of typical operating periods," *Computers & Chemical Engineering*, vol. 65, pp. 54–66, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135414000751

[12] L. Kotzura, P. Markewitz, M. Robinius, and D. Stolten, "Time series aggregation for energy system design: Modeling seasonal storage," *Applied Energy*, vol. 213, pp. 123–135, 2018. [Online]. Available: https://doi.org/10.1016/j.apenergy.2018.01.023

[13] M. Hoffmann, L. Kotzur, D. Stolten, and M. Robinius, "A review on time series aggregation methods for energy system models," *Energies*, vol. 13, no. 3, p. 641, 2020. [Online]. Available: https://doi.org/10.3390/en13030641

[14] R. Renaldi and D. Friedrich, "Multiple time grids in operational optimisation of energy systems with short- and long-term thermal energy storage," *Energy*, vol. 133, pp. 784–795, 2017. [Online]. Available: https://doi.org/10.1016/j.energy.2017.05.120

[15] P. Gabrielli, M. Gazzani, E. Martelli, and M. Mazzotti, "Optimal design of multi-energy systems with seasonal storage," *Applied Energy*, vol. 219, pp. 408–424, 2018. [Online]. Available: https://doi.org/j.apenergy.2017.05.152

[16] T. Brown, J. Hörsch, and D. Schlachtberger, "PyPSA: Python for power system analysis," *Journal of Open Research Software*, Jan 2018. [Online]. Available: https://doi.org/10.5334/jors.188

[17] C. Gerbaulet and C. Lorenz, "dynELMOD: A dynamic investment and dispatch model for the future european electricity market," Deutsches Institut für Wirtschaftsforschung, Tech. Rep., 2017. [Online]. Available: https://dx.doi.org/10.18723/diw_ddc:2017-88

[18] T. Brown, J. Hörsch, and D. Schlachtberger, "Pypsa: Python for power system analysis," Dec. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.18019964

[19] Z. K. Pecenak, M. Stadler, and K. Fahy, "Efficient multi-year economic energy planning in microgrids," *Applied Energy*, vol. 255, p. 113771, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261919314588

[20] G. Mavromatidis and I. Petkov, "Mango: A novel optimization model for the long-term, multi-stage planning of decentralized multi-energy systems," *Applied Energy*, vol. 288, p. 116585, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030626192100129X

[21] I. van Beuzekom, B.-M. Hodge, and H. Slootweg, "Framework for optimization of long-term, multi-period investment planning of integrated urban energy systems," *Applied Energy*, vol. 292, p. 116880, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261921003664

[22] M. Wetzel, E. S. A. Ruiz, F. Witte, J. Schmugge, S. Sasanpour, M. Yeligeti, F. Miorelli, J. Buschmann, K.-K. Cao, N. Wulff, H. Gardian, A. Rubbert, B. Fuchs, Y. Scholz, and H. C. Gils, “Remix: A gams-based framework for optimizing energy system models,” *Journal of Open Source Software*, vol. 9, no. 99, p. 6330, 2024. [Online]. Available: https://doi.org/10.21105/joss.06330

[23] VDI 2067, “Blatt 1 - Wirtschaftlichkeit gebäudetechnischer Anlagen - Grundlagen und Kostenberechnung,” 2012.

[24] M. Schlemminger, T. Ohrdes, E. Schneider, and M. Knoop, “Dataset on electrical single-family house and heat pump load profiles in Germany,” *Scientific Data*, vol. 9, no. 1, p. 56, Feb 2022. [Online]. Available: https://doi.org/10.1038/s41597-022-01156-1

[25] P. Stenzel, J. Linssen, J. Fleer, and F. Busch, “Impact of temporal resolution of supply and demand profiles on the design of photovoltaic battery systems for increased self-consumption,” in *2016 IEEE International Energy Conference (ENERGYCON)*, 2016, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ENERGYCON.2016.7514010

[26] T. Beck, H. Kondziella, G. Huard, and T. Bruckner, “Assessing the influence of the temporal resolution of electrical load and pv generation profiles on self-consumption and sizing of pv-battery systems,” *Applied Energy*, vol. 173, pp. 331–342, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261916305104

[27] L. Kotzur *et al.*, “Tsam: Time series aggregation module,” https://github.com/FZJ-IEK3-VSA/tsam, 2024.

[28] L. Rosin, P. P. Krisam, S. D. Glombik, and E. Schischke, “Economic evaluation of residential energy systems using milp design optimization combined with a real option approach,” in *International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems 2024*, 2024. [Online]. Available: https://doi.org/10.52202/077185-0033