



solph ['sɒlv]
open energy modelling framework

oemof.solph v0.5

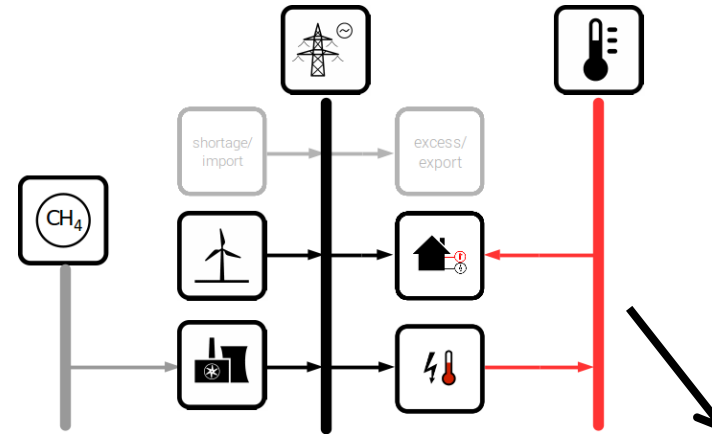
Patrik Schönfeldt

Introduction

- Model generator for mixed-integer linear optimisation
- Energy Balance
- Operation and upfront invest
- Very generic, rather easy to extend
- Time-step flexible (sub-second to years, mixed step-sizes possible)
- No spatial information



solph ['sɒlv]
open energy modelling framework



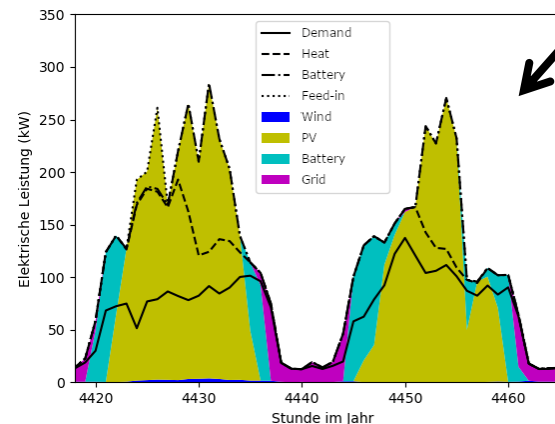
min

objective:

```
+50 flow(electricityBus_0heat_pump)
```

```
+20 flow(heat_pump_thermalBus_0)
```

...



Impact

- Developed by community supported by oemof association
- Enabled > 65 publications (examples listed in and citations to [doi:10.1016/j.simpa.2020.100028](https://doi.org/10.1016/j.simpa.2020.100028))
- 40+ contributors from 10+ institutions
- Top contributors (> 100 Commits)
 - Uwe Krien (RLI, Uni Bremen, Fraunhofer IFAM)
 - Cord Kaldemeyer (EU Flensburg, FH Flensburg)
 - Stephan Günther (Uni Magdeburg)
 - Patrik Schönfeldt (DLR)
 - Johannes Kochems (TU Berlin, DLR)
 - Simon Hilpert (EU Flensburg, ZNE Flensburg)
 - Jann Launer (RLI, TU Delft)
 - Johannes Röder (Uni Bremen)
 - Caroline Möller (RLI)

Solph v0.5.0

Patrik Schönfeldt

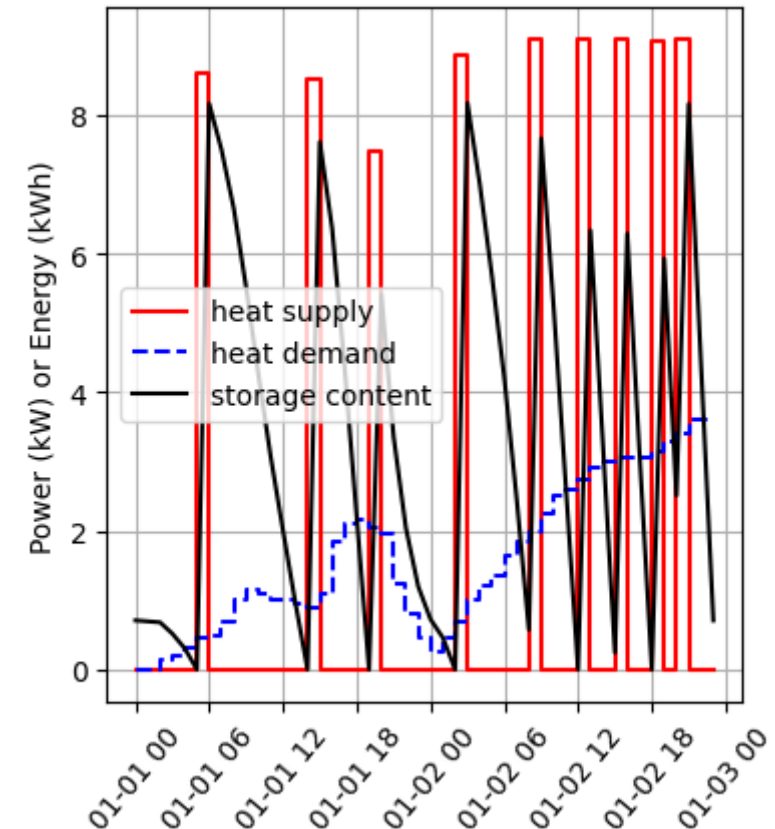
List of changes

- Clean definition of time indexes
- API designed to be more user-friendly
 - Clear structure into buses, components, and flows → `import solph.components.Sink`
 - Public and private API distinguished ('_' signifies private)
 - Experimental code is now sitting in sub-modules called `experimental` (replaces “custom”).
 - The flow arguments `summed_min` and `summed_max` now have the more descriptive names `full_load_time_min` and `full_load_time_max`.
 - Keyword arguments are now explicit.
- Examples are added to the documentation
- Add `inactivity_costs` as an option for Flows for times where a Flow is not operated.
- Possibility to combine NonConvex and Investment optimisation in the same Flow

Time stamp convention

```
EnergySystem(
    timeindex=None,
    timeincrement=None,
    infer_last_interval=True,
    ...
)
```

- Previous (< v0.5):
 - Variable time steps
→ give increment
 - From timeindex,
last interval always inferred
- New (>= v0.5):
 - Variable time steps
also using timeindex
 - Last time step will be dropped
if infer_last_interval == False

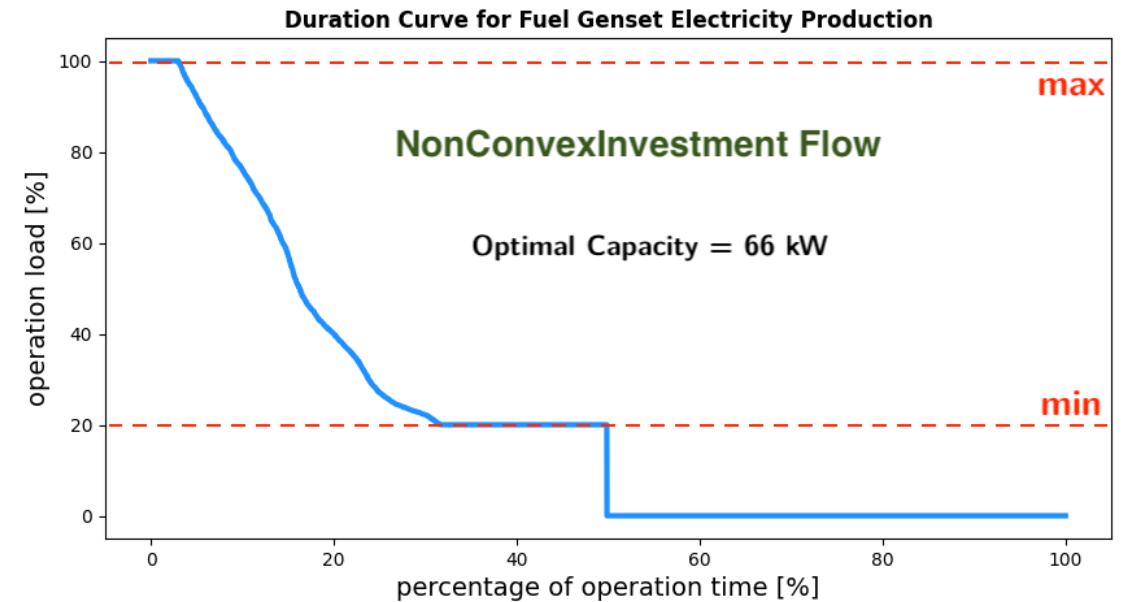
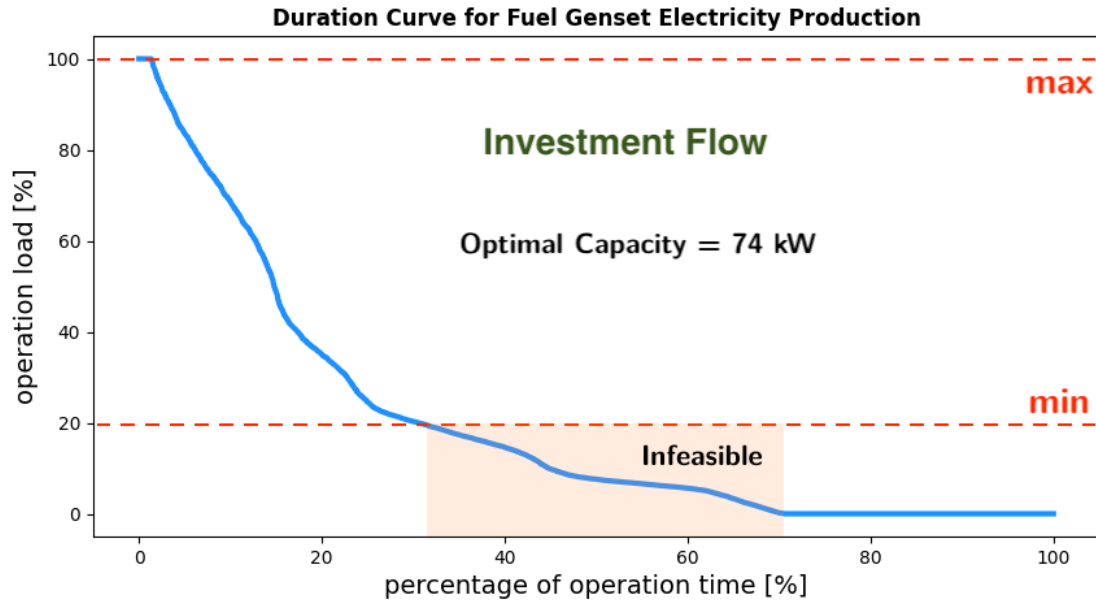


Time stamp examples

```
>>> len(create_time_index(2014))
8761
>>> len(create_time_index(2012)) # leap year
8785
>>> len(create_time_index(2014, interval=0.5))
17521
>>> len(create_time_index(2014, interval=0.5,
number=10))
11
>>> len(create_time_index(2014, number=10))
11
>>> str(create_time_index(2014, interval=0.5,
number=10)[-1])
'2014-01-01 05:00:00'
>>> str(create_time_index(2014, interval=2,
number=10)[-1])
'2014-01-01 20:00:00'
```

timestamp	Energy	Power
00:00	5	2
00:15	$5 + 0.25 * 2$ = 5.5	4
01:15	$5.5 + 1 * 4$ = 9.5	0.5
02:00	10	6
03:00	16	0
04:00	16	4
05:00	20	NaN

NonConvex and Investment



Courtesy of Saeed Sayadi
 (<https://github.com/oemof/oemof-solph/pull/826>)



solph ['sɒlv]
open energy modelling framework

Solph v0.5.1 (upcoming)

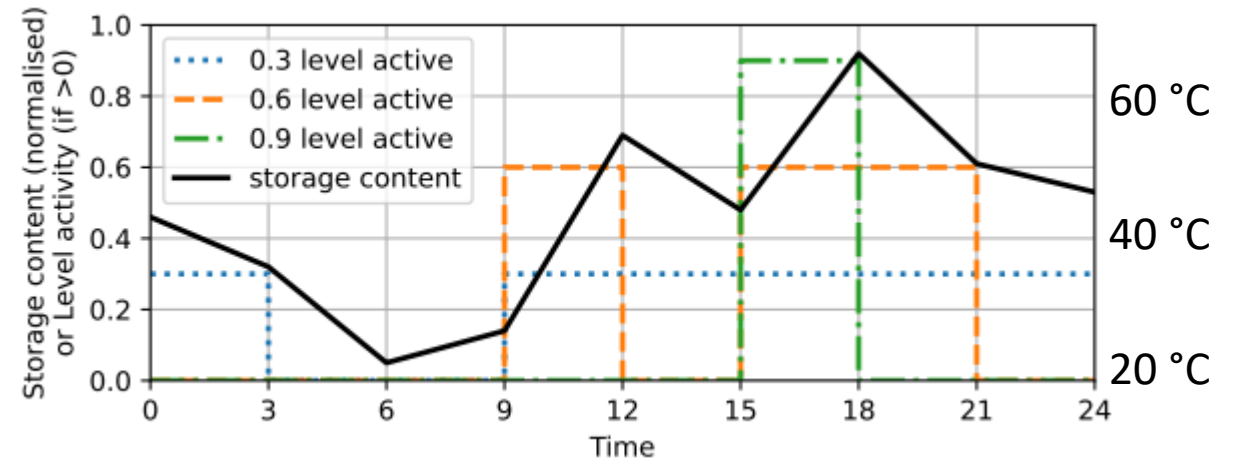
Patrik Schönfeldt

Future changes and their status

- Rename Transformer to Converter (decided)
- Discontinue kwarg “investment” (proposed)
- storage_level_constraint (proposed)
- Custom “preferred time unit” (proposed)
- Multi-period optimisation (merged)

storage_level_constraint

- Example problem:
 - Fully mixed heat storage
 - Temperature dependent on storage content
 - Can it supply 50 °C ?



$$y_n(t) = 0 \text{ if } E(t) < E_n, \text{ and}$$

$$y_n(t) = 1 \text{ if } E(t) \geq E_n.$$



$$y_n(t) \leq \frac{E(t)}{E_n},$$

$$\hat{y}_n(t) \geq \frac{E(t) - E_n}{E_{\max}},$$

$$\bar{y}_n(t) = 1 - \hat{y}_n(t),$$

$$1 = y_n(t) + \bar{y}_n(t)$$

$$P_{\text{out},n}(t) \leq y_n(t) \cdot P_{\text{out,max},n},$$

$$P_{\text{in},n}(t) \leq \bar{y}_n(t) \cdot P_{\text{in,max},n},$$

