LÆND

Documentation v.0.3.2

Heidi Hottenroth, Pforzheim University

March, 2024

The development of this version was carried out within the research project InPEQt - Integrated cost- and life-cycle-based planning of decentralized energy systems for energy and resource efficient neighborhood development. The project was funded by the German Federal Environmental Foundation.

INEC INSTITUTE FOR INDUSTRIAL ECOLOGY

HS PF

Contents

1. About LAEND

LAEND stands for "Life Cycle Assessment based ENergy Decision", which is based on the open source toolbox oemof (open energy model framework) and the life cycle assessment software openLCA and enables a coupled energy system analysis with environmental sustainability assessment and optimization. The method used is Life Cycle Assessment (LCA). The tool is written in the Python programming language and is based on oemof version 0.4.1[1]. In addition to the tools mentioned, it also uses other Python libraries. Parts of this version of LAEND were developed by Dorothee Birnkammer as part of her master thesis.

LAEND is primarily an extension of OEMOF. It incorporates environmental impacts over the entire life cycle of energy technologies into the optimization. In multi-objective optimization, the model minimizes the total impact resulting from the sum of weighted and normalized monetary and environmental impacts. Alternatively, LAEND optimizes for a single objective, such as global warming potential. The environmental impact data in LAEND is imported via a link to openLCA, a product system modeling tool for life cycle assessment. The necessary LCA data for the main renewable energy technologies and energy sources are already provided in LAEND as importable xlsx files. The provided data have been modeled using the ecoinvent LCA database[2]. The EU Environmental Footprint (EF) method 2.0[3] integrated in this database is used as the impact assessment method. This method provides 16 impact indicators and corresponding normalization and weighting factors to form a weighted sum of the individual indicators. This aggregated value is used instead of costs in multi-criteria optimization.

LAEND is based on linear programming and determines an energy system with minimal impact. To account for the transition to renewable energy systems, LAEND uses a myopic optimization approach to map a multi-year period. Existing plants can be included using a brownfield approach[4]. The myopic optimization consists of three steps: (1) mapping the existing energy system, (2) optimizing for a representative year representing one period of support years, and (3) transferring the results to the next representative year representing the next period of support years. For example, a period is five years, of which the first year is always optimized. These steps are repeated until the entire modeling period has been optimized. To optimize the representative year, the graph structure of the energy system is created from a set of investment options. Economic and environmental factors are assigned to all components via investment effects and variable effects. The energy system model is then optimized with perfect foresight for the first year of the period of support years. The solution consists of a set of selected technologies with corresponding installed capacities. This configuration is stored along with the hourly dispatch results. The size of the assets determined is carried over to the next year of optimization. For the second optimization year (e.g. the fifth year of the optimization horizon), the program creates a graph based on previous investments, provided that the technical life of the plant has not been exceeded. New investment opportunities are also added to the model. In the second optimization year, the optimization problem can decide whether to use the existing assets or to invest in new assets. Again, the results are available as a set of technologies with associated

[1] https://oemof-solph.readthedocs.io/en/v0.4.1/index.html
[2] https://ecoinvent.org/
[3] Fazio, S., Castellani, V., Sala, S., Schau, E., Secchi, M., Zampori, L. (2018): Supporting information to the characterisation factors of recommended EF Life Cycle Impact Assessment methods; New models and differences with ILCD EUR 28888 EN; JRC109369; Ispra.
[4] Opposite of greenfield approach where existing plants are not considered

capacities and their utilization. The iterative process ends after the last year of the modeling period has been solved and the results have been saved.

For multi-criteria optimization, the weighted sum of total costs and environmental impacts, aggregated as the environmental footprint, is minimized in a 50:50 ratio (can be customized). Alternatively, only the aggregated environmental impacts, costs, or each of the 16 environmental indicators can be minimized.

Objective outcomes include investment and usage phase information. The investment data includes the facilities implemented and their capacity. The use phase data includes the sum of energy (carriers) consumed or delivered in each year. The optimization results in terms of newly built facilities and flows are multiplied by all impact factors to compile the results in terms of total cost or full LCA. If representative years are used, each value is multiplied by the number of years in the period represented by the optimized year (default is 5 years). This gives the total impact of an optimization to an objective over the entire model period.

As an option political emission reduction targets are handled with emission constraints, an adjustable decarbonization foresight horizon, and model-specific climate neutrality.

Utilizing weather data of typical meteorological years (TMY) to create fixed timeseries simplifies data research and preparation, as the program can derive most fixed timeseries from a single TMY file. Additionally, this ensures cohesive timeseries, as all weather-dependent load curves relate to the same data.

Helpful resources

- Book *Optimization of Energy Supply Systems* by Janet Nagel https://doi.org/10.1007/978-3-319-96355-6
- Oemof documentation at https://oemof-solph.readthedocs.io/en/v0.4.1/index.html
- Oemof github at https://github.com/oemof
- A former version of LAEND with perfect foresight: Tietze, I., Lazar, L., Hottenroth, H., Lewerenz, S. (2020): LAEND: A Model for Multi-Objective Investment Optimisation of Residential Quarters Considering Costs and Environmental Impacts. Energies 13: 614. doi:10.3390/en13030614.

2. Installing LAEND v0.3.2

- Install Python 3.7 or 3.8 (this might be done by installing Anaconda, see https://en.wikipedia.org/wiki/Anaconda_(Python_distribution))
- Create your virtual environment (e.g. https://docs.anaconda.com/anaconda/navigator/tutorials/manage-environments/)
- Install Spyder in your LAEND environment (using Anaconda button)
- Install oemof 0.4.1 (https://oemof-solph.readthedocs.io/en/v0.4.1/readme.html#installation; https://www.youtube.com/watch?v=eFvoM36_szM) (includes also solver installation)
- Install a solver (LAEND assumes you use the CBC solver, though this can be changed in the config file)
- Fetch LAEND from GitHub (https://github.com/inecmod/laend)
- Install the python packages as listed in requirements.txt (https://docs.anaconda.com/free/navigator/tutorials/manage-packages/)

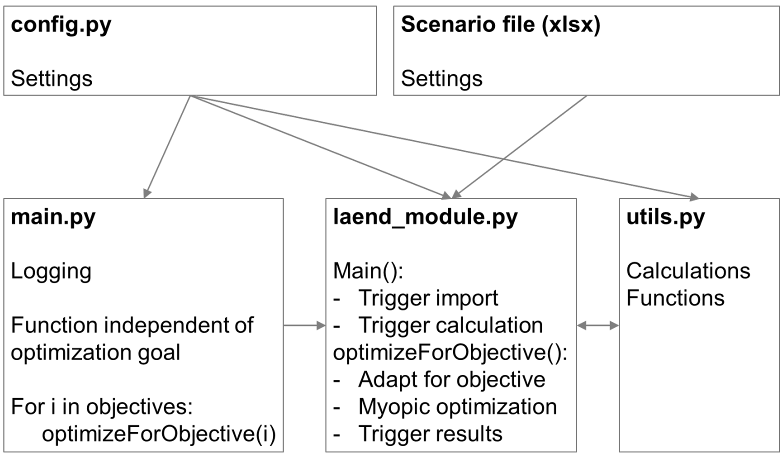
With the download of LAEND, a limited number of usable energy technologies are available. To include other technologies, new LCA data must be created. This requires the freely available openLCA software with a compatible ecoinvent license (for a fee) (more on creating LCA data in section 4.2).

3. File Structure of LAEND

LAEND is separated into four main python files located in the main folder. The scenario of the energy system is configured in an xlsx file.

- main.py
  Main file to run if you want to run LAEND
- laend_module.py
  Two most important functions get triggered here:
  - Main(): This function runs all preparations that are independent of the optimization objective
  - optimizeForObjective(): This function gets an optimization objective and then runs the myopic optimization
- utils.py: Document that has all calculations/functions
- config.py: This file contains settings concerning time, objective, location, profile generation etc.
- Scenario file (xlsx): configuration of the energy system and technology specific settings

The files are connected as follows:



In three subfolders input and output data are saved. Folder ~\in contains the xlsx file with normalization and weighting factors for aggregation of environmental impacts. Input data retrieved from pvgis (see Typical meteorological year (TMY), page 16 and Updating Fixed Renewable Energy Profiles, page 18) are saved in subfolder ~\in\pvgis. Input data for environmental impacts can be found in folder ~\LCA. Results will be saved in folder ~\runs.

4. Configuring the Energy System Model

To configure the energy system model to a specific location the location specific settings are made in a xlsx file and in the config.py file. Both are located in the main folder of LAEND.

4.1. Scenario file

Scenario.xlsx[5] is the main input table for LAEND. This chapter explains the input fields in detail. Names in the column 'label' must not occur twice or end with four integers.

Basic yes/no inputs:

- 0 = False
- 1 = True

In general, all input data should consistently be put in for the following units:

- Energy: kWh
- Power: kW
- Financial data: €
- Emissions: kg CO_2-Eq
- Area: m²

To convert LCA data to these units in some cases conversion factors are necessary. For example, if LCA data for commodity sources like wood chips is given in impact value per kg a conversion factor to convert LCA data to impact value per kWh is applied. E.g. 39 MJ/m³ for natural gas listed in ecoinvent. Conversion factor = 1/39*3.6.

Buses

Buses describe the nodes of the energy system. If the bus is not created, the energy system can neither obtain energy from this bus nor feed into it. It's recommended to keep buses active, even if you don't use the bus. Too many active buses do not impact the system, but a missing bus becomes an issue.

Field	Description	Type of input
label	Name of the bus	string
active	Should the bus be included in the optimization?	Integer (0 or 1)
excess	Should an excess sink be created? If the potential of the other energy sources is insufficient, it supplies the corresponding bus. Set this to active for buses that feed directly into the demand items!	Integer (0 or 1)
shortage	Should a shortage source be created? Creation of a shortage can prevent an unsolvable model.	Integer (0 or 1)
unit	Used in the results xlsx tables	String
shortage costs	If a shortage is created high (variable) costs prevent shortage from being taken instead of the intended sources.	Float
excess_costs	Variable costs e.g. for feed-in tariff	Float
excess_env	Name of LCA dataset for variable environmental impacts e.g. as credit (impact indicators with negative sign)	String
Excess_env_conversion	Conversion factor that is multiplied with excess_env to convert LCA data to impact per kWh.	Float

[5] Default name. If the name is changed the filename has to be changed in the config.py, too.

Demand

The items in demand describe energy sinks. The optimization problem seeks to satisfy the requirements of these energy sinks within given constraints.

Field	**Description**	**Type of input**	**Limitation**
label	Name of the sink / demand item	string	Must be unique
active	Should this row be included in the optimization?	Integer	Only 0 or 1
year_start	Relative start year of period in which the demand shall be used. 0 means in the first year, 5 means in the fifth year. To account for decreasing or increasing demand over time	Integer	
year_end	year of period until the demand shall be used. To account for decreasing or increasing demand over time	Integer	
from	Label of the bus that this sink obtains its energy from	string	Must be included in buses/label
unit	Used in the results Excel tables	String	
nominal value	Multiplication factor The fixed timeseries that is associated with the demand item is multiplied with this factor. If, for example, the fixed timeseries is for one electric mobility load point, a nominal value of 50 would scale the load curve to 50 identical load pints.	Float	>= 0
fixed	Is the flow to this sink fixed (e.g. load curve)? Currently only fixed flows for demand are integrated! Inputting 0 results in errors.	Integer	Only 0 or 1
emission_baseline [kgCO2eq/kWh]	Specific emission factor for the energy type from the year 1990 (as current goals are based on emissions from 1990)	Float	>= 0
climate neutral by	Year in which climate neutrality should be reached	Integer	> 0
comment	Not used in LAEND, for reference only		

Commodity_Sources

Commodity sources describe energy sources that do not require an investment, but can be purchased based on the unit, e.g. electric energy from the grid. They are set up as sources in oemof/LAEND.

Field	Description	Type of input	Limitation
label	Name of the source / commodity source	string	Must be unique
active	Should this row be included in the optimization?	Integer	Only 0 or 1
to	Label of the bus that this source feeds into	string	Must be included in buses/label
unit	Used in the results tables	String	
year_of_availability	Year (number) in which the source becomes available. 0 means from the first year, 5 means from the fifth year on	Integer	
end_of_availability	How long is the source available counted from the year of availability?	Integer	
max_availability	Available amount in kWh has to be divided by nominal value	Float	
nominal_value	Default value needs no change	Float	
variable costs	Variable cost per sourced unit	Float	>= 0
var_env1	Name of LCA dataset for variable environmental impacts	string	
var_env1_conversion	Conversion factor to convert LCA data to impact per kWh.	Float	>= 0
comment	Not used in LAEND, for reference only		
fuel_based	Is this commodity source fuel based?	Integer	Only 0 or 1

Renewables

These items describe renewable energy sources, for example wind, that have a fixed flow based on atmospheric conditions. These renewables are set up as oemof sources with a fixed flow in LAEND.

Field	Description	Type of input	Limitation
label	Name of the renewable source	string	Must be unique
active	Should this row be included in the optimization?	Integer	Only 0 or 1
to	Label of the bus that this source feeds into	string	Must be included in buses/label
unit	Used in the results Excel tables	String	
area	Will be included in the area constraint if > 0; unit per m^2	Float	>= 0
initial_existance	Does this technology already exist?	Integer	Only 0 or 1
initially_installed_capacity	Initially installed capacity in kW	Float	>= 0
year_of_availability	Year (number) in which the source becomes available. 0 means from the first year, 5 means from the fifth year on	Integer	
end_of_availability	How long is the source available counted from the year of availability?	Integer	
variable costs	Variable cost per sourced unit	Float	>= 0
var_env1	Name of LCA dataset for variable environmental impacts	string	
var_env1_conversion	Conversion factor to convert to one kWh	Float	>= 0

Field	Description	Type of input	Limitation
om	Operation and maintenance cost for one kW and one year	Float	>= 0
invest	Investment costs for one kW	Float	>= 0
lifetime	Lifetime of technology	Integer	> 0
max_capacity_invest	Maximum capacity for investments in kWh	Float	>= 0
inv1	Name of LCA dataset for investment	String	
inv1_conversion	Conversion factor to get LCA for 1 kW capacity	Float	> 0
fixed	Is flow from this source fixed? Should always be 1 for renewables	Integer	Only 0 or 1

In order to include energy retrofits in the model, they can be modeled approximately like a fixed flow renewable energy source. The fixed flow represents the difference in heat demand between the current and retrofit condition. It must be manually added to the Time Series worksheet of the scenario file (8784 hours). This source feeds the heat bus. Since the time series is not normalized, the absolute investment cost is given to achieve the heat demand reduction assumed in the profile. In this case, the maximum capacity must be set to 1. The LCA record can refer to the sum of all materials required for the renovation or to a specific insulation material. This can be stored in kg, in which case the conversion factor must be related to the sum of the required mass.

Storages

Storage items include all technologies that can store energy.

Field	Description	Type of input	Limitation
label	Name of the storage (heat or electricity storages can be modelled)	string	Must be unique
active	Should this row be included in the optimization?	Integer	Only 0 or 1
bus	Label of the bus that this storage takes energy from and feeds back into	string	Must be included in buses/label
unit	Used in the results Excel tables	String	
initial_existance	Does this technology already exist?	Integer	Only 0 or 1
initially_installed_capacity	Initially installed capacity in kW	Float	>= 0
year_of_availability	Year (number) in which the source becomes available. 0 means from the first year, 5 means from the fifth year on	Integer	
end_of_availability	How long is the source available counted from the year of availability?	Integer	
balanced	Should storage level be the same at the end of the optimization period as at the beginning? E.g. if 1 (=True) then storage level at the end of the year must be at level on January 1, when it is first bought	Integer	Only 0 or 1
capacity loss	How much (fraction) of the capacity is lost within one hour?	Float	0 <= x <= 1
efficiency inflow	How much of the energy that the bus feeds into the storage actually ends up in storage?	Float	0 <= x <= 1
efficiency outflow	How much of the energy that leaves the storage actually ends up back at the bus?	Float	0 <= x <= 1

Field	Description	Type of input	Limitation
initial capacity	What is the capacity when the storage is purchased?	Float	0 <= x <= 1, must be larger than capacity min
capacity min	Minimum storage capacity, e.g. LFP batteries should not be deeply discharged below 10% of storage capacity	Float	0 <= x <= 1
capacity max	Maximum storage capacity	Float	0 <= x <= 1, must be larger than capacity min
invest_relation_ input_capacity	How quickly can energy be passed to storage in relation to the storage size? E.g. what is the charging speed (in kW) in relation to the storage size (kWh)	Float	
invest_relation_output_capacity	How quickly can energy leave storage?	Float	
variable input costs	Variable costs associated with moving energy to storage	Float	>= 0
variable output costs	Variable costs associated with moving energy out of storage	Float	>= 0
max_capacity_invest	Maximum capacity for investment in kWh	Float	>= 0
inv1	Name of LCA dataset for investment	String	
inv1_conversion	Conversion factor to get LCA for 1 kWh capacity	Float	> 0
om	Operation and maintenance cost for one kWh capacity and one year	Float	>= 0
invest	Investment costs for one kWh	Float	>= 0
lifetime	Lifetime of technology	Integer	> 0
comment	Free field, not used in optimization	String	

Cycle times are not included in LAEND. The system assumes full functionality until the end of the lifetime is reached.

Transformers_in

Transformers_in include all items that transform energy from one bus to another. The investment mode is placed on the input flow. In LAEND, a transformer_in can have one energy input and one or two energy outputs.

Important to note: since all investment data applies to the energy input they should be provided in €/kW_{input}

Field	Description	Type of input	Limitation
label	Name of the transformer	string	Must be unique
active	Should this row be included in the optimization?	Integer	Only 0 or 1
unit	Used in the results Excel tables	String	
initial_existance	Does this technology already exist?	Integer	Only 0 or 1
initially_installed_capacity	Initially installed capacity in kW	Float	>= 0
year_of_availability	Year (number) in which the source becomes available. 0 means from the first year, 5 means from the fifth year on	Integer	
end_of_availability	How long is the source available counted from the year of availability?	Integer	
from1	Label of the primary energy input bus	string	Must be included in buses/label
var_from1_costs	Variable costs of from1 → transformer flow	Float	>= 0
to1	Label of the primary energy output bus	string	Must be included in buses/label
var_to1_costs	Variable costs of transformer → to1 flow	Float	>= 0
conversion_factor1	Conversion factor of energy input to energy that feeds into to1	Float	>= 0
var_env1	Name of LCA product system for variable environmental impacts	string	
var_env1_conversion	Conversion factor to convert to one kWh	Float	>= 0
to2	Label of the primary energy output bus *Can be left blank if not applicable*	string	Must be included in buses/label
var_to2_costs	Variable costs of transformer	Float	>= 0
conversion_factor2	Conversion factor of energy input to energy that feeds into to2	Float	>= 0
var_env2	Name of LCA product system for variable environmental impacts	string	
var_env2_conversion	Conversion factor to convert to one kWh	Float	>= 0
om	Operation and maintenance cost for one kW capacity and one year	Float	>= 0
invest	Investment costs for one kW (input capacity!)	Float	>= 0
lifetime	Lifetime of technology	Integer	> 0
max_capacity_invest	Maximum capacity for investment in kWin	Float	>= 0
inv1	Name of LCA product system for investment	String	
inv1_conversion	Conversion factor to get LCA for 1 kW of capacity (input capacity!)	Float	> 0
fixed_cop	Does this technology have a fixed (different for all hours in the year) conversion_factor1?	Integer	Only 0 or 1
fuel_based	Is this transformer fuel based? 0 sets inv_env and var_env to zero for model specific climate neutrality calculation	Integer	Only 0 or 1
comment	Free field, not used in optimization	String	

Transformers_out

Transformers_out include all items that transform energy from one bus to another. The investment mode is placed on the output flow. In LAEND, a transformer_out can have one or two energy inputs and one energy output.

Important to note: since all investment data applies to the energy output they should be provided in €/kW_{output}

Field	Description	Type of input	Limitation
label	Name of the transformer	string	Must be unique
active	Should this row be included in the optimization?	Integer	Only 0 or 1
unit	Used in the results Excel tables	String	
initial_existance	Does this technology already exist?	Integer	Only 0 or 1
initially_installed_capacity	Initially installed capacity in kW	Float	>= 0
year_of_availability	Year (number) in which the source becomes available. 0 means from the first year, 5 means from the fifth year on	Integer	
end_of_availability	How long is the source available counted from the year of availability?	Integer	
from1	Label of the primary energy input bus	string	Must be included in buses/label
var_from1_costs	Variable costs of from1 → transformer flow	Float	>= 0
from2	Label of the socondary energy input bus	string	Must be included in buses/label
to1	Label of the primary energy output bus	string	Must be included in buses/label
conversion_factor1	Conversion factor of energy input to energy that feeds into to1	Float	>= 0
fixed_cop	Does this technology have a fixed (different for all hours in the year) conversion_factor1?	Integer	Only 0 or 1
var_to1_costs	Variable costs of transformer → to1 flow Variable costs for transformer → to2 flow not yet implemented!	Float	>= 0
var_env1	Name of LCA product system for variable environmental impacts	string	
var_env1_conversion	Conversion factor to convert to one kWh	Float	>= 0
om	Operation and maintenance cost for one kW capacity and one year	Float	>= 0
invest	Investment costs for one kW (output capacity!)	Float	>= 0
lifetime	Lifetime of technology	Integer	> 0
inv1	Name of LCA product system for investment	String	
inv1_conversion	Conversion factor to get LCA dataset for 1 kW of capacity (output capacity!)	Float	> 0
max_capacity_invest	Maximum capacity for investment in kWin	Float	>= 0
fuel_based	Is this transformer fuel based? 0 sets inv_env and var_env to zero for model specific climate neutrality calculation	Integer	Only 0 or 1
comment	Free field, not used in optimization	String	

Timeseries

Timeseries is the sheet that allows for the import of fixed flows e. g. load curves into the model. Column A lists the hourly timestamps in the following format: yyyy-mm-dd HH:MM:SS e. g. 2016-01-01 03:00:00.

All other columns can be filled with fixed timeseries. To create a fixed profile for a renewable energy source or demand item, create the column title in the format label.fix. To create a fixed cop, e. g. for an air-water heat pump, name the column label.cop. Every row of energy source of demand which is marked as fixed needs a profile in the timeseries sheet. Here is an example:

timestamp	**load_el.fix**	**wind.fix**	**heat_pump_a_w.cop**
2016-01-01 00:00:00	76.7409	0.08515	3.7548
2016-01-01 01:00:00	76.9936	0.08386	3.76269
2016-01-01 02:00:00	69.6969	0.08228	3.77176

Make sure that manually added timeseries need 8784 h to account for leap years. Load curves for heat, electricity and electric mobility electricity demand can be generated with nPro tool[6] (mind the missing leap day).

For automatically generated timeseries, see section "Updating fixed demand load curves" and "Updating Fixed Renewable Energy Profiles".

4.2. LCA data configuration

For goods, renewables and transformers, the multi-criteria optimization model requires data on the environmental impacts of investment and operation in addition to the costs.

Unlike the costs, which are only inserted into the scenario file as a numerical value, a set of impact indicators must be available for the environmental impacts in order to be taken into account by LAEND. These files must contain the impact indicator results for the impact assessment method used in LAEND (currently EU Environmental Footprint v2.0). They are imported via xlsx files. For common technologies and goods, these files are already stored in the ~\LCA folder. An overview of the files can be found in the appendix. If new technologies or goods are to be included in LAEND, corresponding product systems must first be created in openLCA. These are accessed via the interface to openLCA to create the xlsx files. As with costs, a distinction is made between investments and variable environmental impacts. For this reason, the product systems in openLCA must also be available in this form. This differs from the usual approach in LCA, where a data set for the generation of a unit of electricity or heat also contains the infrastructure and, if applicable, a supplied energy source. In the LCA data supplied in the ~\LCA folder, which relates to the operation of systems and is based on ecoinvent datasets, the infrastructure and energy source share was therefore deleted in order to obtain only the pure operating emissions (e.g. heat generation in the pellet furnace does not contain any pellets nor furnace which are considered elsewhere).

The name used in the scenario file for the LCA to be used must be available as an xlsx file in the LCA folder, otherwise the interface to openLCA must be activated (open your database in openLCA → Tools → Developer tools → IPC Server), whereby a product system with the name must be available in the open database. The missing xlsx file is then generated from this and stored in the LCA folder, where it

[6] www.npro.de

is available for further runs. To create a new product system in openLCA, please refer to the software manual[7].

4.3. Configurations in config.py

Config.py is the main configuration file for the optimization. In the first part, model specific settings are done, in the second part location specific settings are done.

Temporal settings

- Define the **start year** and the **end year**. Leave the calc_start and calc_end unchanged.
- Choose *None* for **number of timesteps**. For testing purposes, the number of timesteps can be changed to calculate less than a full year. Use a number smaller than 8760. Emission constraints do not scale to this limited view! Some other functionalities may also not work as expected.
- **Aux_years:** Set to *True* to use representative years and select the steps (in years). Use False to optimize each year individually.
- **Aux_year_steps:** Steps between representative years. Choosing *aux_year_steps* with a period of 5 years and starting in the year 2024 will mean, that the first period runs from 01.01.2024 to 31.12.2028. The year 2024 will serve as the representative year.
- **max_cap_once**: Set to *True* means that if the maximum capacity is set this capacity is limited to the whole lifetime of the investment. Set to False means that a new maximum capacity is available in every optimization year to simulate a rising capacity over time.

Set Optimization Objectives

- **Objective:** Activate an objective by removing the #. Deactivate, by placing a # in the row before the name. Possible objectives include the following:
    - System costs.
    - Environmental impact in the form of a single EF 2.0 indicator,
      'climate change - climate change total',
      'resources - dissipated water',
      'resources - fossils',
      'resources - land use',
      'resources - minerals and metals',
      'ecosystem quality - freshwater and terrestrial acidification',
      'ecosystem quality - freshwater ecotoxicity',
      'ecosystem quality - freshwater eutrophication',
      'ecosystem quality - marine eutrophication',
      'ecosystem quality - terrestrial eutrophication',
      'human health - carcinogenic effects',
      'human health - ionising radiation',
      'human health - non-carcinogenic effects',
      'human health - ozone layer depletion',
      'human health - photochemical ozone creation',
      'human health - respiratory effects, inorganics'
    - Total environmental impacts aggregated as EU Environmental Footprint (here named JRCII), which applies normalization and weighting factors to arrive at a single factor.

---

[7] https://greendelta.github.io/openLCA2-manual/introduction/index.html

- Combination of system costs and Environmental Footprint (EnvCosts). Costs are normalized. Costs and aggregated environmental impacts are each weighted individually (default: 50 :50).
- Combination of normalized system costs and single environmental impacts at an equilibrium weighting (Equilibrium)

Set multiprocessing

Multiprocessing of several objectives can be enabled or disabled. True if several objectives should run in parallel to speed up the calculation time, False for testing and to use debug mode, since debug messages are not displayed during multiprocessing.

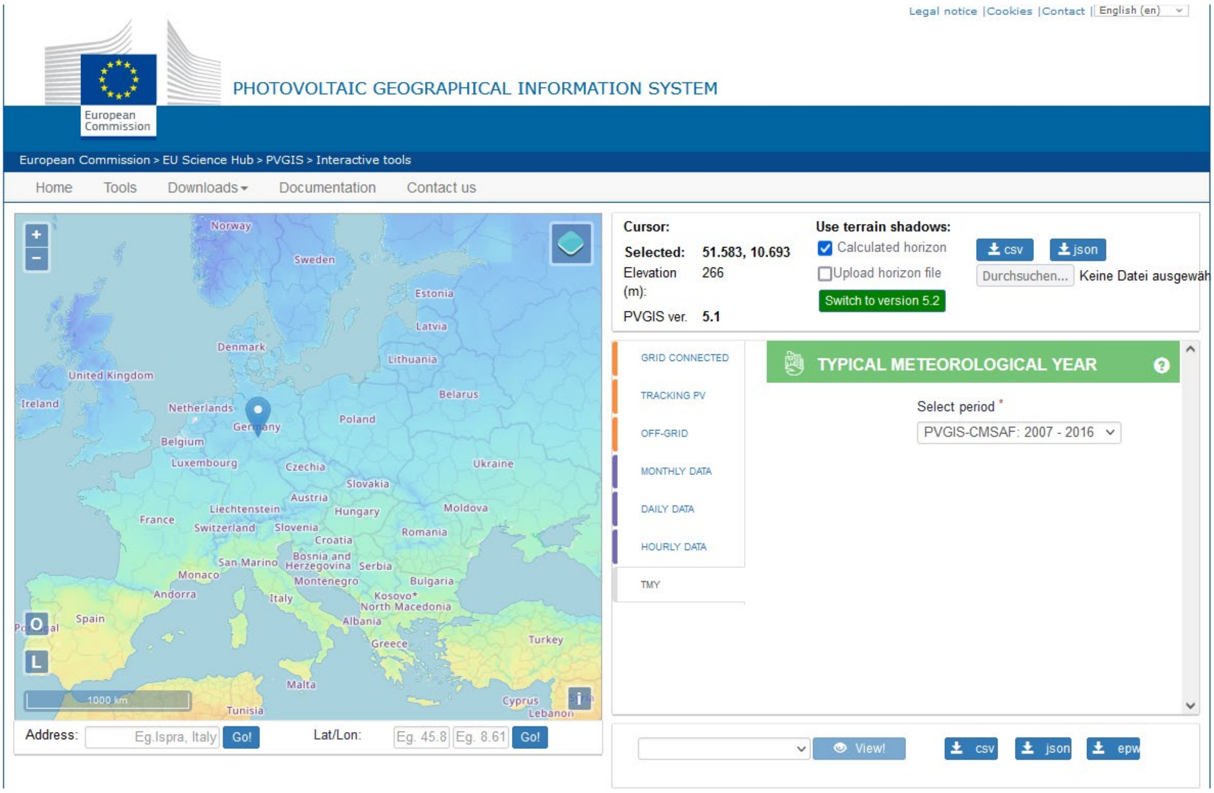
Filename configuration

Change the name of your scenario file if necessary.

Typical meteorological year (TMY)

The TMY has to be adapted to the location of the modelled energy system. Several fixed timeseries can be created with a TMY. The TMY acts as a filter for PV data and as the source to generate fixed curves for all other types of timeseries. If you want to update the TMY to a new location, go to PVGIS (https://re.jrc.ec.europa.eu/pvg_tools/en/#api_5.1). Select period 2007-2016 for the selected location and download csv (Make sure that PVGIS ver. 5.1 is selected). Store the file tmy_XX.XXX_X.XXX_2007_2016.csv in folder \in\pvgis.

PVGIS:



- **filename_tmy**: Change the file location and name of the TMY here.

Location settings

Adapt timezone, longitude, latitude and location name.

Updating fixed demand load curves

In general, the following settings are *True/False*. If set to *True*, then the program generates the corresponding csv file with the parameters set here. This file is then modified and saved as a timeseries in ~\runs\[time]\files\[time]_scenario.xlsx. Please note, that the original scenario.xlsx is never changed by the program. Instead, it changes the copy of the file in the corresponding run folder.

Electricity

If update_electricity_demand is set to True a new electricity demand profile with the parameters set here will be generated based on BDEW standard load profiles for different sectors.

# G0	Trade in general	Weighted average of profiles G1-G6
# G1	Commercial on weekdays 8 a.m. - 6 p.m.	e.g. offices, doctors' surgeries, workshops, administrative facilities
# G2	Businesses with heavy to predominant consumption in the evening hours	e.g. sports clubs, fitness studios, evening restaurants
# G3	Continuous trade	e.g. cold stores, pumps, sewage treatment plants
# G4	Shop/hairdresser	
# G5	Bakery with bakehouse	
# G6	Weekend operation	e.g. cinemas
# G7	Mobile phone transmitter station	Consistent band load profile
# L0	Farms in general	Weighted average of profiles L1 and L2
# L1	Farms with dairy farming/part-time animal husbandry	
# L2	Other farms	
# H0/H0_dyn	Household/household dynamized	

The program generates the corresponding csv file with the parameters set here with the filename set for filename_el_demand which is stored in the folder ~\in.

The corresponding *varname_el_demand* must be exactly the same as the corresponding *label* in the sheet *demand* in scenario file.

Heat

Heat demand profiles will be generated internally if update_heat_demand is set to *True*. The total heat demand load curve is derived from natural gas demand curves in relation to the atmospheric temperature. This method was originally developed by Hellwig (2003) and formalized in a guideline by the German Bundesverband der Energie- und Wasserwirtschaft e.V. (BDEW) (2011). The python library bdew, part of the demandlib that integrates with oemof, computes an hourly total heat demand profile. The following settings go into the function that employs the bdew python library to calculate the load curve.

- Annual demand for private buildings separated in single family houses (efh) and multifamily houses (mfh) and for trade, commerce, service (ghd) have to be set.

- **building_class:** 1 – 11; class of building according to bdew classification for share of new and old buildings based on housing unit according to https://www.eko-netz.de/files/eko-netz/download/3.5_standardlastprofile_bgw_information_lastprofile.pdf

	Share of old buildings		Mean proportion of	
classe	from	to	Old buildings	New buildings
1	85,5%	90,5%	88,0%	12,0%
2	80,5%	85,5%	83,0%	17,0%
3	75,5%	80,5%	78,0%	22,0%
4	70,5%	75,5%	73,0%	27,0%
5	65,5%	70,5%	68,0%	32,0%
6	60,5%	65,5%	63,0%	37,0%
7	55,5%	60,5%	58,0%	42,0%
8	50,5%	55,5%	53,0%	47,0%
9	45,5%	50,5%	48,0%	52,0%
10	40,5%	45,5%	43,0%	57,0%
11			75,0%	25,0%

Old buildings until 1979

- **building_wind_class:** 0=not windy, 1=windy
- The program generates the corresponding csv file with the parameters set here with the filename set for **filename_th_demand** which is stored in the folder ~\in.
- **Separate_heat_water:** separates space heating and hot water into two load curves. The development included subdividing the total heat load curve into space heating (SH) and hot water (HW) demand to account for different circuit flow temperatures. This becomes relevant when low-temperature heat sources like heat pumps are integrated into the energy system. The average hourly heat demand in July and August is assumed to be equal to the constant HW demand. If this average exceeds the total required thermal energy for one given hour, the HW demand equals the total heat demand during this period. The SH curve is derived by subtracting the HW demand from the total heat load curve. With enabled separation two buses for high and low temperature heat and individual transformers feeding these buses are needed. Setting this to False requires one load curve, one demand profile and only one bus.
- The corresponding *varname_th_low* and *varname_th_high* must be exactly the same as the corresponding *labels* in the sheet *demand* in scenario file.

Updating Fixed Renewable Energy Profiles

In general, the following renewable energy settings always contain the following:

- **Update_technology**: True/False, determines if the profiles in ~\runs\[time]\files\[time]_[scenario file].xlsx get updated.
- **Varname_technology**: This string must be exactly the same as the label in sheet renewables in scenario file for the update to work.

PV

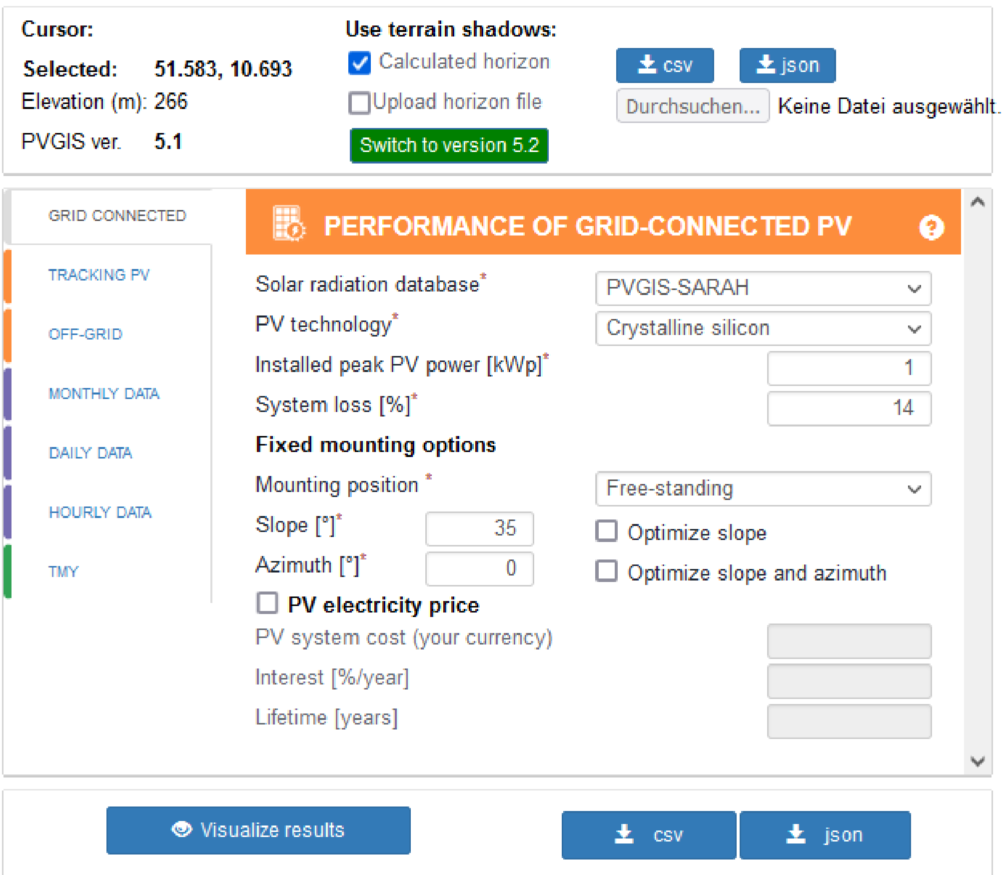
Generation of PV profiles is differentiated according to the orientation of the plant.

- **update_pv_**...: Setting them to `True` will import the new pvgis file(s) and update in\location_name\pvgis_tmy_varname.csv.
- **filename_pv_..._fix**: path to the necessary file; please watch whether it's an csv or xlsx file before changing this!

Generation of location specific PV profiles

To update PV data, proceed as with the TMY at https://re.jrc.ec.europa.eu/pvg_tools/en/#PVP, but download the grid connected data. Make sure that the data contains all years that the TMY uses. Save the file and make sure that the file in config.py links to it.

Choose the specific location, switch to version 5.1 and choose years 2007-2016, database (PVGIS-SARAH), PV technology, system loss, mounting position, slope, azimuth or optimal direction, respectively. Download the csv file for every configuration of PV plant that will be available in the model.



Optimal/south orientation

Flat roof and open ground plants are assumed to be south oriented. Currently, three variables can be equipped with the same load curve.

Wind

The windpowerlib tool is used to generate a profile for a wind turbine. The following parameters must be available for this, which can be taken from https://www.wind-turbine-models.com, for example:

- **windPlantCapacity**: capacity in [kW]
- **wind_hub_hight**: hub hight in [m]
- **p_in_power_curve**: a list with power in W for wind speed of 0, 3, 5, 10, 15, 25 m/s
- **wind_z0_roughness**: surface roughness; length in m; data according to https://www.researchgate.net/figure/Surface-roughness-length-m-for-each-land-use-category-in-WAsP-and-WRF-and-the_tbl1_340415172:
  bare rocks, sparely vegetated areas = 0.01 m;

arable land, pastures, nat. grasslands = 0.1 m;
forest = 0.9 m

Solar collectors

The python library oemof.thermal allows for the generation of a solar collector heat supply load based on irradiance and temperature data of the TMY as well as technological parameters such as efficiency, slope, and inflow temperature.

- **temp_collector_inlet**: Collectors inlet temperature in [°C]
- **delta_temp_n**: Temperature difference between collector inlet and mean temperature (of inlet and outlet temperature)
- collector_tilt
- collector_azimuth
- **a_1**: Thermal loss parameter k1 [W/(m²K)]
- **a_2:** Thermal loss parameter k2 [W/(m²K²)]
- **eta_0**: Optical efficiency of the collector (float between 0 and 1)

Air/water heat pump COP

For air to water heat pumps, the efficiency depends on the temperature of the heating circuit and the heat source, here ambient air. In LAEND, this flexible COP can be modeled with the python library oemof.thermal for two temperature levels.

- **hp_temp_high**: maximum outlet temperature in [°C]
- **hp_temp_low**: maximum outlet temperature in [°C]
- **a_w_hp_quality_grade**: 0.4 is default setting for air/water heat pumps
- **hp_temp_threshold_icing**: temperature below which icing occurs at heat exchanger [°C]
- **hp_factor_icing:** sets the relative COP drop by icing; 1 = no efficiency drop [0<f<1]

<u>Area constraint</u>

If the area for renewable technologies is limited this can be considered by a constraint. If the area constraint is set to True the model sets a constraint to all technologies in sheet "renewables" in scenario file, where row "area" > 0.

- **area_constraint**: *True* or *False*
- **area:** in [m²] is the overall area for restricted technologies

<u>LCA Update</u>

By default, LAEND will use the excel files in folder \LCA as input data. If a technology in scenario file does not have an associated file, the program will automatically try to connect to openLCA to get the correct file. If you want to force the program to update all LCA files from the database, use these settings:

- **Update_LCA_data:** False (default); use True to force an update (open your database in openLCA → Tools → Developer tools → IPC Server; only possible with an ecoinvent licence for openLCA; this version is usable without)
- **LCA_impact_method:** This is the LCA impact method that is pulled from openLCA. Changing the method entails changes in several places in the code.

<u>Normalization & Weighting</u>

These settings are necessary to build the weighted sum for the multi-objective optimization.

- **Filename_weight_and_normalisation:** Excel filename for the LCA normalization and weighting file. Ensure that headings for impact categories here match the LCA excel files (see above)
- **Normalization_cost_gdp:** The costs get normalized by GDP 2010 with current prices for 2023 to be in line with environmental normalization, change it here if investment and variable costs do not refer to the year of current prices assumed (see Appendix for data sources and approach).
- **Normalization_per_person:** Instead of using a global value to normalize, the values can also be normalized per person. If set true, the normalization values are divided by the global population (variable “normalization_person_population”).
- **Weight_cost_to_env:** The objective “EnvCosts” allows for an individual weighting of aggregated environmental impacts and costs. Set the ratio of costs to aggregated environmental impacts here.
- **Weight_cost_to_env_equilibrium**: 1/17 if equal weighting between every single goal for multi-objective "Equilibrium"

Emission Constraints

Settings for the functionality of the emission constraint, emission factor and model-specific climate neutrality.

- **Emission_constraint:** True to activate, False to run without a constraint
- **Ec_horizon:** Window of foresight for the emission constraint in years (must be integer>1)
- **Ec_impact_category:** The impact category that the emission factor is pulled from. Ensure this is an exact match with the openLCA export data.
- **Ec_buffer:** If the optimization for one year with an emission constraint failed, then the same year runs again with an emission constraint that is increased by this buffer. After 3 tries the program aborts.
- **Ef_fuel_based_only:** Is the emission factor only applied to fuel based technologies? Whether or not a technology is fuel based can be set in scenario file

Defining Climate Neutrality

- **Def_cn_calculate_climate_neutrality**: True/False to activate/deactivate calculation for climate neutrality. If set to false, it will only calculate based on the values given in scenario file
- **Def_cn_include_investment:** True/False. Should the investment impacts be included in the calculation for climate neutrality? If set to false, then all manufacturing impacts are set to zero for the climate neutrality optimization
- **Def_cn_year_climate_neutrality**: Integer. If scenario file only lists emission targets until a specific year, for example 2030, then this value determines the end of the linear decrease. However, if the model specific climate neutrality (e.g. 1 t CO_2/year) > last given emission constraint (0.5 kg CO_2/kWh for an annual demand of 1000 kWh), then it will linearly increase the emission limit to reach the higher climate neutrality. This does not influence the final outcome, as the higher value (of climate neutrality or the political goal) is selected as the emission constraint.
- **Def_cn_fuel_based_only:** True/False. Use this setting to employ only fuel-based technology when calculating climate change. If set to true, all technologies set to non-fuel-based in scenario.xlsx will have variable environmental impacts of zero.

Financial Settings

- **InvestWacc:** This weighted average cost of capital (wacc) is used to calculate the annuity of investments and employs the function oemof.economics.annuity; for a discount rate of e.g. 4% insert 0.04
- **Invest_min_threshhold (in kW/kWh):** Currently, the program can use extremely small capacities. This threshold does not limit the systems' ability to employ new small capacities. However, if a capacity below this threshold is selected in year 1, then it does not get passed on to the next calculation year as an employed capacity.

Technical Settings

Changing these technical settings should be handled with caution. Most importantly, this section contains information about the employed solver. No testing has been done with activated solver options.

- **Logging level:** See the API-doc of `define_logging()` [8] for all details.
- **showTable:** True or False; results for invest capacities per year are displayed in a table per optimization goal and year on the console if True
- **InvestTimeSteps**: Do not change!
- **System_impacts_index:** Do NOT change this list unless you're updating the LCIA method!

5. Logging

Logging runs through the oemof library *logging*. If you want to change the logging level e. g. only show info tags and remove all debug messages, this has to be adjusted in oemof. Locate the site packages > oemof > tools > logger.py. In line 95, you can set the level. Please note that this applies to the screen level and to the file level.

Set to DEBUG to get all messages, set to INFO to get information messages.

```
if logpath is None:
    logpath = extend_basic_path('log_files')

file = os.path.join(logpath, logfile)

log = getLogger('')

# Remove existing handlers to avoid interference.
log.handlers = []
log.setLevel(DEBUG)

if file_format is None:
    file_format = (
        "%(asctime)s - %(levelname)s - %(module)s - %(message)s")
file_formatter = Formatter(file_format, file_datefmt)
```
Changing the settings in laend_module.py will not affect the results.

[8] https://oemof-tools.readthedocs.io/en/latest/reference/oemof_tools.html#oemof.tools.logger.define_logging

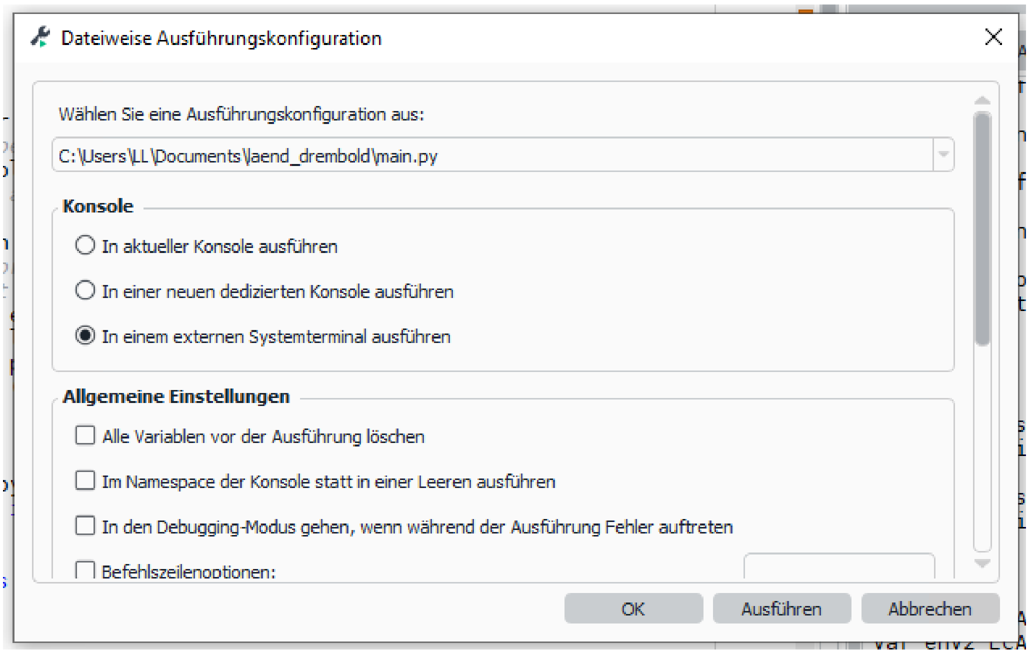
6. Running LAEND

The program can be run in two different ways:

- External System Terminal: Running it in an external system terminal does NOT allow a user to stop the program while it is running, but log files are visible while the program runs.
- In Spyder Control Panel: Running it in the Spyder Kernel using multiprocessing may result in logs not being written to the console, which means that the user cannot read the progress as the program calculates results. The program will still run and will write the logs to log files, but the progress is not immediately visible.

6.1. External System Terminal

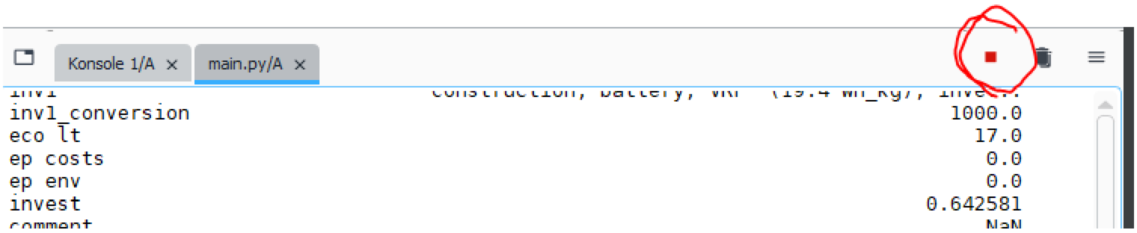
1. Double click on main.py so that it is selected
2. Go to Run > File configuration (Ausführen > Dateiweise Konfiguration)
3. Choose to run the console in an external system terminal. Make sure the file mentioned in the first item is indeed main.py



4. Click on run/Ausführen
5. A console window will open to run the program
6. Should the program not show any progress for a few minutes, try hitting enter.

6.2. In Spyder Console

1. Open main.py in Spyder
2. Click “run file”
3. If you may want to interrupt the execution click the red square



6.3. First run

Before the first run for a new location the update variables for fixed profiles should set to *True*. The profiles will be saved in the copied scenario file in the results folder.

6.4. Subsequent runs

For subsequent runs copy the timeseries generated in the first run and saved t the copied scenario file in the results folder to your original scenario file in the main folder. If necessary make changes to the scenario file, set the variable for updating fixed time series to *False* and run the optimization again.

7. Results

The results for the invested capacities are displayed in a table on the console per optimization objective and year if **showTable** = True. This allows you to get an initial overview and do a first plausibility check.

	label	period end	technology	type	unit	value	year	objective
0	PV_facade_2020	2020	PV_facade	invest	kW	0.000000	2020	EnvCosts
1	PV_flat_roof_2020	2020	PV_flat_roof	invest	kW	378.636060	2020	EnvCosts
2	ambient_heat_ground_2020	2020	ambient_heat_ground	invest	kW	179.036670	2020	EnvCosts
3	batt_LFP_2020	2020	batt_LFP	invest	kWh	0.000000	2020	EnvCosts
4	batt_VRF_2020	2020	batt_VRF	invest	kWh	0.000000	2020	EnvCosts
5	heating_unit_2020	2020	heating_unit	invest	kW	50.334039	2020	EnvCosts
6	heat_pump_a_w_high_2020	2020	heat_pump_a_w_high	invest	kW	0.000000	2020	EnvCosts
7	heat_pump_b_w_high_2020	2020	heat_pump_b_w_high	invest	kW	240.314110	2020	EnvCosts
8	heat_storage_high_2020	2020	heat_storage_high	invest	kWh	540.275170	2020	EnvCosts
9	solar_thermal_FPC_high_2020	2020	solar_thermal_FPC_high	invest	kW	0.000000	2020	EnvCosts

The program saves all results in the same folder with all the other program files. In the folder *runs*, a new folder with the date and time of the optimization run is created. Here, the following can be found:

- **Laend_config.py:** copy of the config file for later reference
- **Files:**
  - *[time]_[scenario file].xlsx*: copy of file at start of run for later reference
  - Results for *objective_[time]*.xlsx: Result for this single objective
    - Impacts: Table of the environmental and financial impacts. In the case of representative (aux) years, the results from the one year will be multiplied with the number of years until the next optimization year/end of the modeling period. Highlight the entire sheet and create a pivot table for easier analysis.
    - Tech: Shows the capacity and energy flows for each technology.
  - Flows for *objective_[time]*.csv: Dispatch results for each optimized hour.
  - Results for total: If all optimizations were successful, then the files *Results for objective.xlsx* are aggregated into a single table for easier analysis (*Results total [time].xlsx*).
- **Logs:** These log files help to find errors. While running LAEND in the (Spyder) console they also help to see if the program is still running while multiprocessing.
- **Oemof_dumps:** These are all of the optimization runs that the program optimized and saved. These tend to get quite large, so deleting unnecessary runs every now and then is recommended. These dumps can be restored to see individual oemof results.

8. Appendix

8.1. Cost normalization

The normalization of the environmental impacts is based on the sum of the global environmental impacts per indicator in 2010. Since then, the world population and industrialization have continued to increase, so that further increases in environmental impacts can be assumed up to the present day. In order to aggregate costs and environmental impacts in a weighted sum, the costs must also be normalized, whereby the normalization value must be calculated under comparable assumptions as for the environmental impacts. For costs, global "costs" from the year 2010 are therefore required at current prices in euros (or at the time of the cost assumptions in the model).

Data basis:

World GDP 2010 in US dollars at current prices: 66.54 billion dollars

Source: IMF, World Economic Outlook (October 2023), https://www.imf.org/external/datamapper/NGDPD@WEO/WEOWORLD?year=2010, accessed 23.02.2024

Conversion to EURO with the exchange rate for 2023: 1.08 Domestic currency per EURO

Source: IMF, https://data.imf.org/?sk=cb5462fc-9197-43d1-af26-18d6e8e4e784&hide_uv=1, accessed 23.02.2024

Normalization factor: 1/61.61 billion EURO

8.2. LCA data

CATEGORY	Process	original	original but per capacity unit	w/o infrastructure	geographical scope adjusted	other content adjustments	Freely available data set
Commodity sources							
	biomethane, low pressure_RER_MJ - RER				+		
	market for cleft timber, measured as dry mass, commodity_DE_kg				+	+	
	Marketforelectricity_2022_Naegler_Sc6						
	Marketforelectricity_2023_Naegler_Sc6						
	Marketforelectricity_2024_Naegler_Sc6						
	Marketforelectricity_2025_Naegler_Sc6						
	Marketforelectricity_2026_Naegler_Sc6						
	Marketforelectricity_2027_Naegler_Sc6						
	Marketforelectricity_2028_Naegler_Sc6						
	Marketforelectricity_2029_Naegler_Sc6						
	Marketforelectricity_2030_Naegler_Sc6						
	Marketforelectricity_2031_Naegler_Sc6						
	Marketforelectricity_2032_Naegler_Sc6						
	Marketforelectricity_2033_Naegler_Sc6						
	Marketforelectricity_2034_Naegler_Sc6						
	Marketforelectricity_2035_Naegler_Sc6						
	Marketforelectricity_2036_Naegler_Sc6						

CATEGORY	Process	original	original but per capacity unit	w/o infrastructure	geographical scope adjusted	other content adjustments	Freely available data set
	Marketforelectricity_2037_Naegler_Sc6						
	Marketforelectricity_2038_Naegler_Sc6						
	Marketforelectricity_2039_Naegler_Sc6						
	Marketforelectricity_2040_Naegler_Sc6						
	Marketforelectricity_2041_Naegler_Sc6						
	Marketforelectricity_2042_Naegler_Sc6						
	Marketforelectricity_2043_Naegler_Sc6						
	Marketforelectricity_2044_Naegler_Sc6						
	Marketforelectricity_2045_Naegler_Sc6						
	Marketforelectricity_2046_Naegler_Sc6						
	Marketforelectricity_2047_Naegler_Sc6						
	Marketforelectricity_2048_Naegler_Sc6						
	Marketforelectricity_2049_Naegler_Sc6						
	Marketforelectricity_2050_Naegler_Sc6						
	market for biomethane, high pressure_m3 - RER				+		
	market for natural gas, low pressure, commodity_RER_m3				+		
	market for light fuel oil_kg	+					
	market for wood chips, measured as dry mass_commodity_kg - DE				+	+	
	market for wood pellet, measured as dry mass, commodity_kg	+					
OPERATION, ELECTRICITY							
CHP	electricity, CHP, biomethane, low pressure burned in micro gas turbine 345 kW, 100kWe, variable/kWh - CH			+			
	electricity, CHP, biomethane, low pressure burned in polymer electrolyte membrane fuel cell 2kWe, future, variable/kWh - CH			+			
	electricity, CHP, biomethane, low pressure burned in solid oxide fuel cell 266 kW, 125kWe, future, variable/kWh - CH			+			
	electricity, CHP, biomethane, low pressure burned in solid oxide fuel cell, with micro gas turbine, 327 kW, 180kWe, future, variable/kWh - CH			+			
	electricity, CHP, natural gas, burned in micro gas turbine, 345 kW, 100kWe, variable/kWh - CH			+			
	electricity, CHP, wood chips, 2000 kW, 1200 kWth, 530 kWel, variable – CH					+	
	electricity, CHP, wood chips, 2000 kW, state-of-the-art 2014, variable/kWh - CH			+			
	electricity, CHP, wood chips, wet, 6667 kW, state-of-the-art 2014, variable/kWh – DE			+			
OPERATION, HEAT							
CHP	heat, central or small-scale, CHP, biomethane, low pressure burned in micro gas turbine, 345 kW, 100kWe, variable_MJ – CH			+			
	heat, central or small-scale, CHP, biomethane, low pressure burned in polymer electrolyte membrane fuel cell 2kWe, future, variable/MJ – CH			+			
	heat, central or small-scale, CHP, biomethane, low pressure burned in solid oxide fuel cell, 266 kW, 125kWe, future, variable/MJ – CH			+			
	heat, central or small-scale, CHP, biomethane, low pressure burned in solid oxide fuel cell, with micro gas turbine, 327 kW, 180kWe, future, variable/MJ – CH			+			
	heat, central or small-scale, CHP, natural gas, low pressure, burned in micro gas turbine, 345 kW, 100kWe - variable/MJ – CH					+	
	heat, district or industrial, CHP, wood chips, 2000 kW, 1200 kWth, 530 kWel, variable – CH			+			
	heat, district or industrial, CHP, wood chips, 2000 kW, state-of-the-art 2014, variable/MJ – CH			+			
	heat, district or industrial, CHP, wood chips, wet, 6667 kW, state-of-the-art 2014, variable/MJ – DE			+			

CATEGORY	Process	original	original but per capacity unit	w/o infrastructure	geographical scope adjusted	other content adjustments	Freely available data set
Furnace/boiler	heat, central or small scale, light fuel oil, at boiler 100kW condensing, non-modulating, variable_MJ – DE			+			
	heat, central or small scale, light fuel oil, at boiler 10kW, non-modulating, variable_MJ – DE			+			
	heat, central or small-scale, biomethane, at boiler condensing modulating <100kW, variable_MJ – DE			+			
	heat, central or small-scale, mixed logs, at furnace 30kW, state-of-the-art 2014, variable/MJ – DE			+			
	heat, central or small-scale, natural gas, at boiler condensing modulating max 100kW, variable_MJ – DE			+			
	heat, central or small-scale, wood pellet, at furnace 25kW, state-of-the-art 2014, variable/MJ – DE			+			
	heat, central or small-scale, wood pellet, at furnace 300kW, state-of-the-art 2014, variable/MJ – DE			+			
	heat, central or small-scale, wood pellet, at furnace 9kW, state-of-the-art 2014, variable/MJ – DE			+			
	heat, district or industrial, hardwood chips from forest, at furnace 300kW, state-of-the-art 2014, variable/MJ – DE			+			
	heat, district or industrial, softwood chips from forest, at furnace 300kW, state-of-the-art 2014, variable/MJ – DE			+			
	heat, district or industrial, wood chips from industry, at furnace 300kW, state-of-the-art 2014, variable/MJ – DE			+			
	heat, district or industrial, wood chips, at furnace 300kWth, state-of-the-art 2014, variable_MJ			+			
Heat pump	heat, central or small-scale, air-water heat pump 10kW, variable/MJ – Europe without Switzerland			+			
	heat, central or small-scale, brine-water heat pump 10kW, variable_MJ – Europe without Switzerland			+			
INVESTMENT							
CHP	construction, CHP, wood chips, 6667 kW, state-of-the-art 2014, investment/kW – DE		+				
	construction, CHP, micro gas turbine, 345 kW, 160 kWth, 100kWe, investment_kW - RER				+		
	construction, CHP, wood chips, 2000 kW, investment_kW – RER				+		
	construction, CHP, solid oxide fuel cell 266 kW, 125kWe, future, investment_kW – RER				+		
	construction, CHP, fuel cell, polymer electrolyte membrane, 6.25kW, 2kWe, 3.44 kWth, future, investment_kW - RER				+		
	construction, CHP, fuel cell, solid oxide, with micro gas turbine, 327kW, 180kWe, 92kWth, future, investment_kW - RER				+		
HEAT	production, auxiliary heating unit, electric, 5kW, investment_kW – RER				+		
	production, brine-water heat pump 10kW, investment_kW – RER				+		
	production, furnace, logs, 30kW, investment_kW – RER				+		
	production, furnace, pellets, 25kW, investment_kW – RER				+		
	production, furnace, wood chips, with silo, 1000kW, investment_kW – RER				+		
	production, furnace, wood chips, with silo, 300kW, investment_kW – RER				+		
	production, furnace, wood pellet, 300kW, investment_kW – RER				+		
	production, heat storage, 2000l_unit - RER				+		
	production, solar collector system, Cu flat plate collector, multiple dwelling, for combined system, investment_m2 – RER				+		
	production, solar collector system, Cu flat plate collector, one-family house, for combined system, investment_m2 – RER				+		

CATEGORY	Process	original	original but per capacity unit	w/o infrastructure	geographical scope adjusted	other content adjustments	Freely available data set
	construction, solar collector system, flat plate collector, ground-mounted_m²						+
	construction, solar collector system, tube collector, ground-mounted_m²						+
	construction, borehole heat exchanger, investment_m – DE				+	+	
	production, air-water heat pump 10kW, investment_kW – Europe without Switzerland		+				
	production, heat exchanger, wastewater_m2						+
	production, furnace 9kW, state-of-the-art 2014, wood pellet, investment/kW – Europe without Switzerland		+				
Photovoltaics	production, PV, facade installation, 3kWp, cryst_2021_kW - DE						+
	PV, flat-roof installation, 156kWp, cryst_kW - DE						+
	PV, flat-roof installation, 3kWp, cryst_kW – DE						+
	production, PV, flat-roof installation, 156kWp, cryst_2021_kW - DE						+
	PV, slanted-roof installation, 3kWp, CIS_kW – DE						+
	PV, slanted-roof installation, 3kWp, CdTe_kW – DE						+
	production, PV, slanted-roof installation, 3kWp, cryst_2021_kW - DE						+
	construction, photovoltaic plant, 1.3MWp open ground, CdTe_kW						+
	construction, photovoltaic plant, 1.3MWp open ground, cryst-Si_kW						+
	construction, photovoltaic plant, 570kWp open ground, CdTe_kW						+
	construction, photovoltaic plant, 570kWp open ground, cryst-Si_kW						+
BATTERY	production, stationary battery, LFP-LTO (37.9 Wh per kg), investment_kWh – GLO						+
	production, stationary battery, VRF (19.4 Wh effective per kg), investment_kWh – GLO						+
WIND	construction, wind turbine, 3.2 MW, DDSG, DDPMSG, DFIG, onshore_kW						+
	construction, wind turbine, 6kW, onshore_kW						+
	construction, wind turbine, onshore_kW						+
	construction, wind turbine, 3.2MW, DDPMSG, onshore_kW						+
	construction, wind turbine, 3.3MW, DFIG, onshore_kW						+
	construction, wind turbine, 3MW, DDSG, onshore_kW						+
	wind turbine construction, 3.2 MW (DDPMSG), onshore, w_o maintenance						+
	maintenance, wind turbine, 3.2 MW, (DDPMSG), onshore						+
	market for maintenance, wind turbine, 3.2 MW, DDSG, DDPMSG, DFIG, onshore_unit						+
	credit_Marketforelectricity_2033_Naegler_Sc6						+
	storage production, 10'000 l, investment_unit						+