

Modelling an Energy System- Tutorial

This tutorial demonstrates ZEN garden's functionalities by applying them to simple energy systems. To execute the datasets presented in this tutorial, you first need to install ZEN garden by following the instructions stated in the ["How to ZEN garden" manual](#). Once the installation is completed, the [dataset examples folder](#) must be copied into the *data* folder of your ZEN garden python project (to run a dataset it must be located in the *data* folder). If you have not yet copied the *config.py* file from the [testcases folder](#) into your data directory, this must be done to select the individual dataset which should be executed.

The following dataset examples are sorted by increasing complexity of the functionalities, starting with the most basic features. In the following datasets, the newly introduced features are applied to the previous dataset such that the change caused by the added adaption can be seen in the results as clearly as possible.

Flow Chart Conventions

The flow charts describe the energy system setup for each dataset by visualising all technologies defined in the dataset. Transparent technology symbols were not built by the optimizer. The numbers above the arrows represent the flow rates of the carriers (e.g., heat, electricity, etc.). For datasets with multiple time steps per year, the corresponding time step of the flow rates stated in the flow chart is included in the title (e.g., TS 50).

Content

- 1) [1_base_case](#): most basic features to define an energy system
- 2) [2_add_photovoltaics](#): add an additional conversion technology to the energy system
- 3) [3_multi_year_optimization](#): change basic time parameters such as the number of optimized years
- 4) [4_variable_demand](#): define time-/space-dependent parameters
- 5) [5_reduced_import_availability](#): additional example of defining time-/space-dependent parameters, resulting in the need for a transport technology
- 6) [6_PWA_conversion_technology](#): approximate non-linear conversion efficiency of conversion technology by piece-wise affine (PWA) approximation
- 7) [7_multiple_time_steps_per_year](#): optimize multiple time steps per year
- 8) [8_reduced_import_availability_yearly](#): additional example of defining time-/space-dependent parameters, resulting in the need for a storage technology
- 9) [9_time_series_aggregation](#): using time series aggregation (TSA) to reduce computational complexity of optimizing larger multi-time step problems
- 10) [10_full_year](#): optimization of an hourly resolved full year
- 11) [11_yearly_variation](#): specifying yearly parameter changes more easily
- 12) [12_myopic_foresight](#): optimize using myopic foresight instead of perfect foresight
- 13) [13_brown_field](#): include existing capacities from before the start of the optimization
- 14) [14_multi_scenario](#): optimize several scenarios with differences in parameter values more efficiently
- 15) [15_multiple_output_carriers_conversion_techs](#): define conversion technologies with multiple output carriers
- 16) [16_multiple_input_carriers_conversion_techs](#): define conversion technologies with multiple input carriers

- 17) [17_yearly_parameter_missing_values](#): define yearly parameter and use interpolation feature to get values at time indices without specified value
- 18) [18_interpolation_off](#): switch off the interpolation feature in order to use the default values at time indices without specified value

1_base_case

In this example, we will investigate a simple energy system to supply the heat demand in Switzerland (CH) and Germany (DE). Only natural gas (NG) boilers are available, which consume natural gas, imported at each node (country). Furthermore, the optimizer can build NG storages and pipelines (Figure 1).

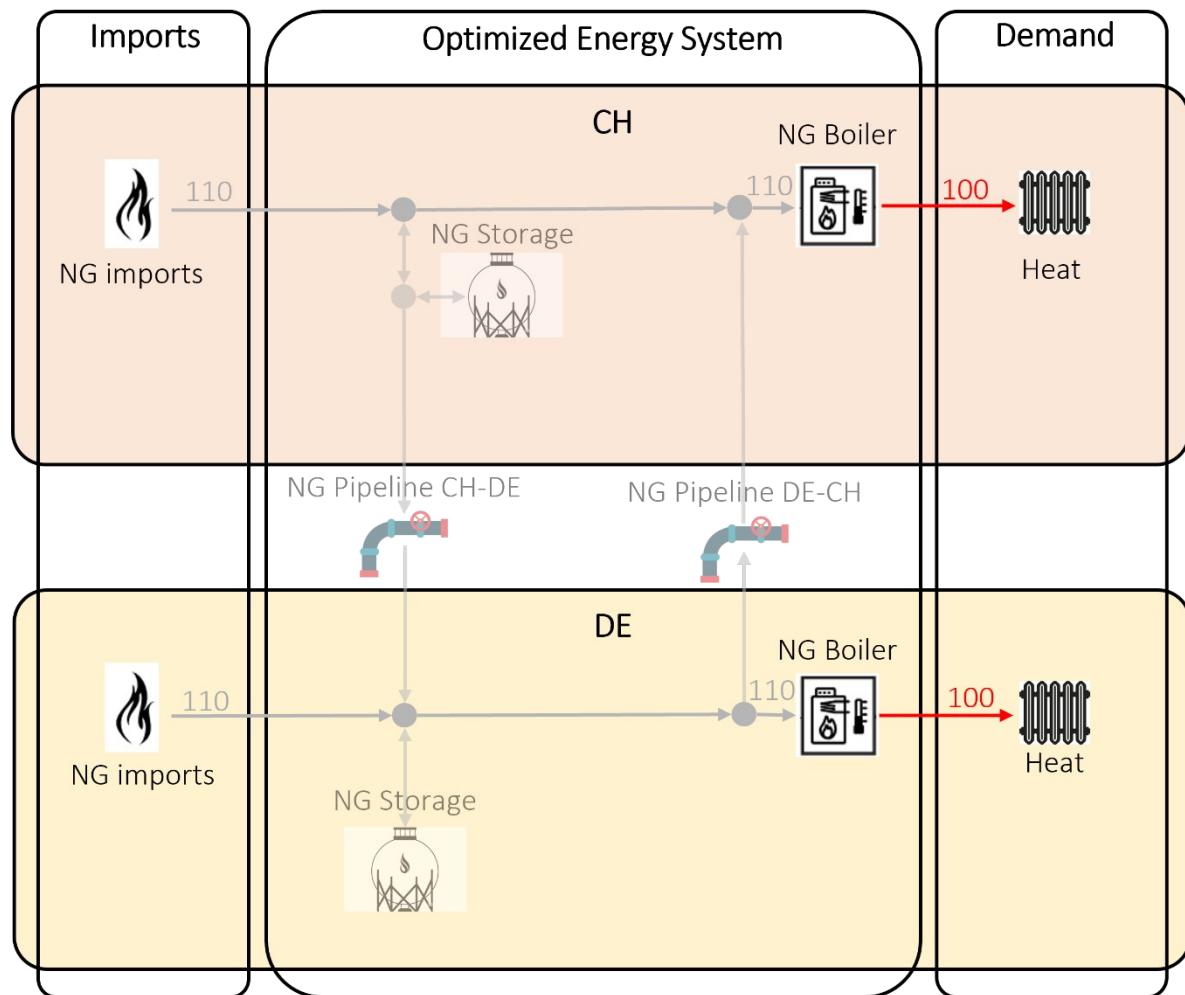


Figure 1: Energy system "base_case" generating heat by burning natural gas in boilers.

Input data structure

To define an energy system in ZEN garden, the available technologies and the corresponding energy carriers must be specified. Since all technologies can be allocated to either conversion, transport or storage technologies, these three technology class folders must be provided. Additionally, a directory to define the carriers corresponding to the technologies and a directory for system-wide specifications must be part of each dataset (Figure 2).

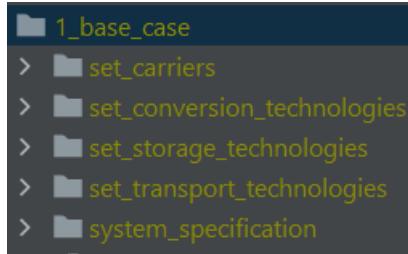


Figure 2: Dataset folder structure.

As a next step, the technologies of the introduced energy system (Figure 1) must be defined by creating the three folders named after the technologies in the corresponding technology classes (Figure 3 left). The individual technology folders must contain the so-called *attributes.csv* file which specifies the technology's **default parameter values** (can be copied from other technologies of the same type) (Figure 3 middle/right). A default value is always defined as **index, value, unit**. If the parameter is unitless, the unit is an empty string. In addition to the *attributes.csv* file, transport technologies must always be given the *distance.csv* file, specifying the distances between the nodes (can be copied as well from other datasets) (Figure 3 left).

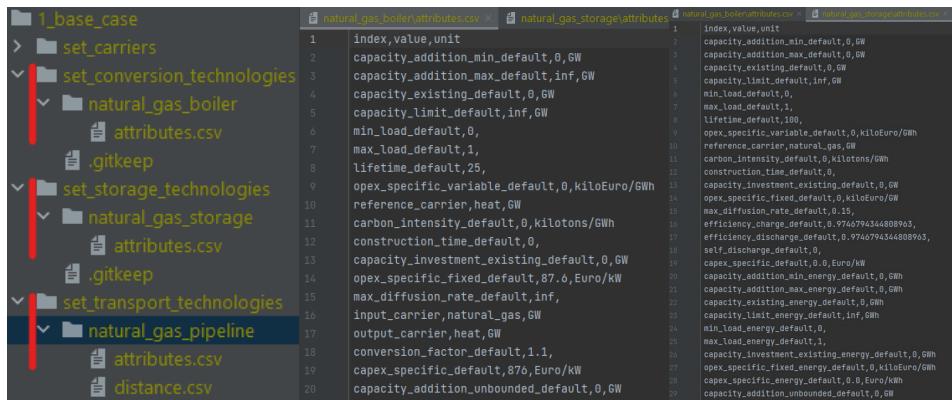


Figure 3: (f.l.t.r.) Folder structure for technology definition; attributes file needed to define parameters of natural gas boiler; attributes file needed to define parameters of natural gas storage.

As the three different technology classes convert, store or transport energy carriers, all the carriers utilised by the technologies must be defined by creating a folder named after the carrier (Figure 4 left). The carrier names must be consistent with the names used in the technologies' attribute files (Figure 4 middle). The carrier folders must contain the *attributes.csv* file which specifies the default values of the carrier parameters (Figure 4 right).

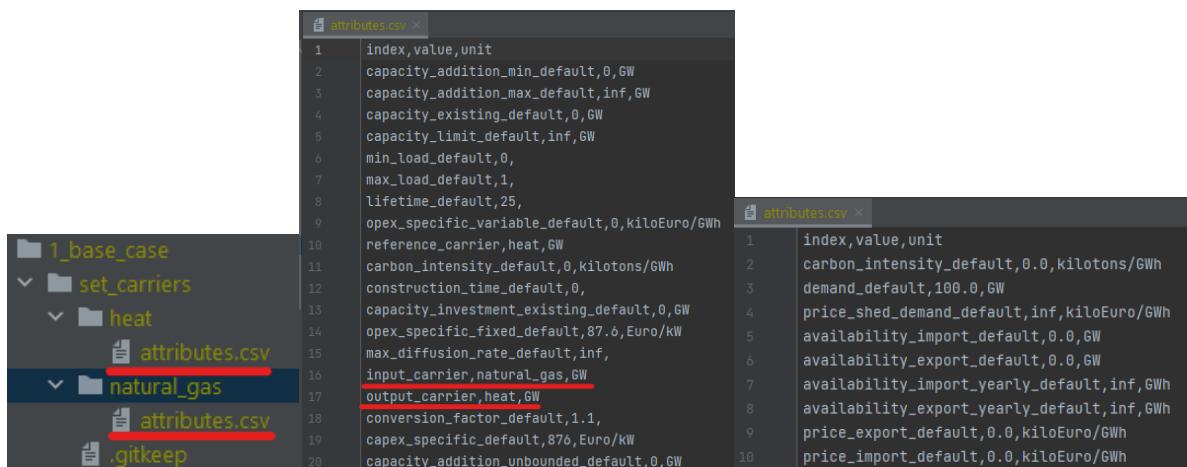


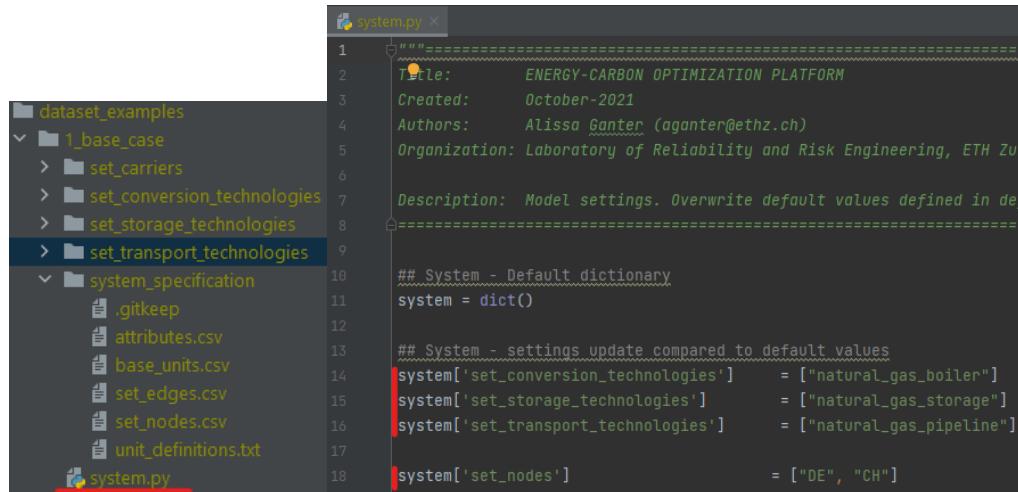
Figure 4: (f.l.t.r.) Carrier definition ; carrier names attributes file; attributes file heat.

The directory “*system_specification*” defines system-wide parameters. Its *attributes.csv* file specifies default values for the system parameters. The *base_unit.csv* file is needed to convert the units used to quantify the default values in the attribute files to a set of matching base units (the base unit file can be copied from other datasets as well).

To create your own dataset, all the directories and files the “*1_base_case*” dataset consists of must be included (of course, the technologies can be substituted by other ones).

Technology and Node Selection

The selection of nodes and technologies that are allowed to be built during the optimization is done in the *system.py* file, each dataset must have in its top-level directory (Figure 5 left). Therefore, the technologies natural gas boiler, natural gas storage and natural gas pipeline must be selected in the *system.py* file in order to represent the energy system in Figure 1 (Figure 5 right). Since the energy system consists of the countries Switzerland and Germany, they must be selected as nodes.



The screenshot shows a file explorer on the left and a code editor on the right. The file explorer shows a folder structure:

- dataset_examples**
 - 1_base_case**
 - set_carriers
 - set_conversion_technologies
 - set_storage_technologies
 - set_transport_technologies
 - system_specification**
 - .gitkeep
 - attributes.csv
 - base_units.csv
 - set_edges.csv
 - set_nodes.csv
 - unit_definitions.txt
- system.py**

The code editor displays the *system.py* file with the following content:

```

1  """=====
2   Title: ENERGY-CARBON OPTIMIZATION PLATFORM
3   Created: October-2021
4   Authors: Alissa Ganter (aganter@ethz.ch)
5   Organization: Laboratory of Reliability and Risk Engineering, ETH Zurich
6
7   Description: Model settings. Overwrite default values defined in def
8   ====="""
9
10  ## System - Default dictionary
11  system = dict()
12
13  ## System - settings update compared to default values
14  system['set_conversion_technologies'] = ["natural_gas_boiler"]
15  system['set_storage_technologies'] = ["natural_gas_storage"]
16  system['set_transport_technologies'] = ["natural_gas_pipeline"]
17
18  system['set_nodes'] = ["DE", "CH"]

```

Figure 5: (f.l.t.r.)Location in folder structure of *system.py* file; *system.py* file where nodes and technologies must be selected.

Whereas it is allowed to define more technologies than selected in the *system.py* file, it is not allowed to select technologies which are not defined in the corresponding technology class folder.

Results

In this example, the optimizer only installs NG boilers to supply the heat demand (Figure 6). No storage or transport capacities are built since the heat demand in the two countries can be met by directly importing and converting natural gas.

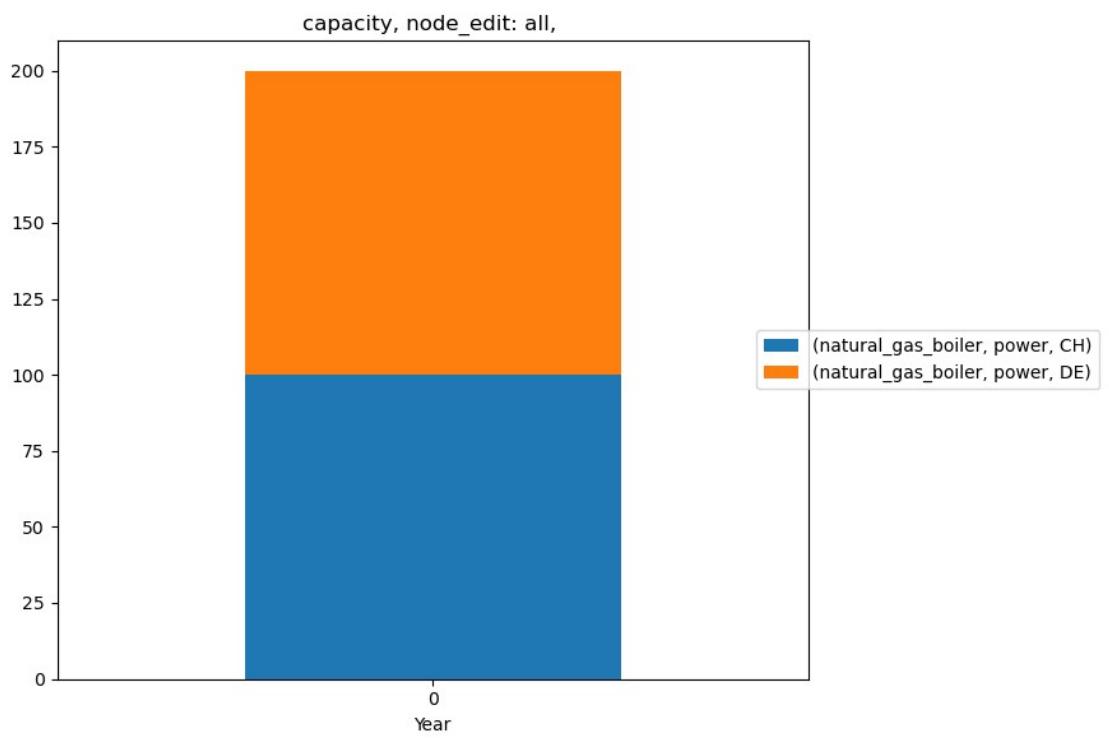


Figure 6: Installed technology capacities of natural gas boilers at the nodes Germany and Switzerland.

2_add_photovoltaics

In the second example, we extend the system by an electricity demand that can be supplied by photovoltaics (PV) (Figure 7).

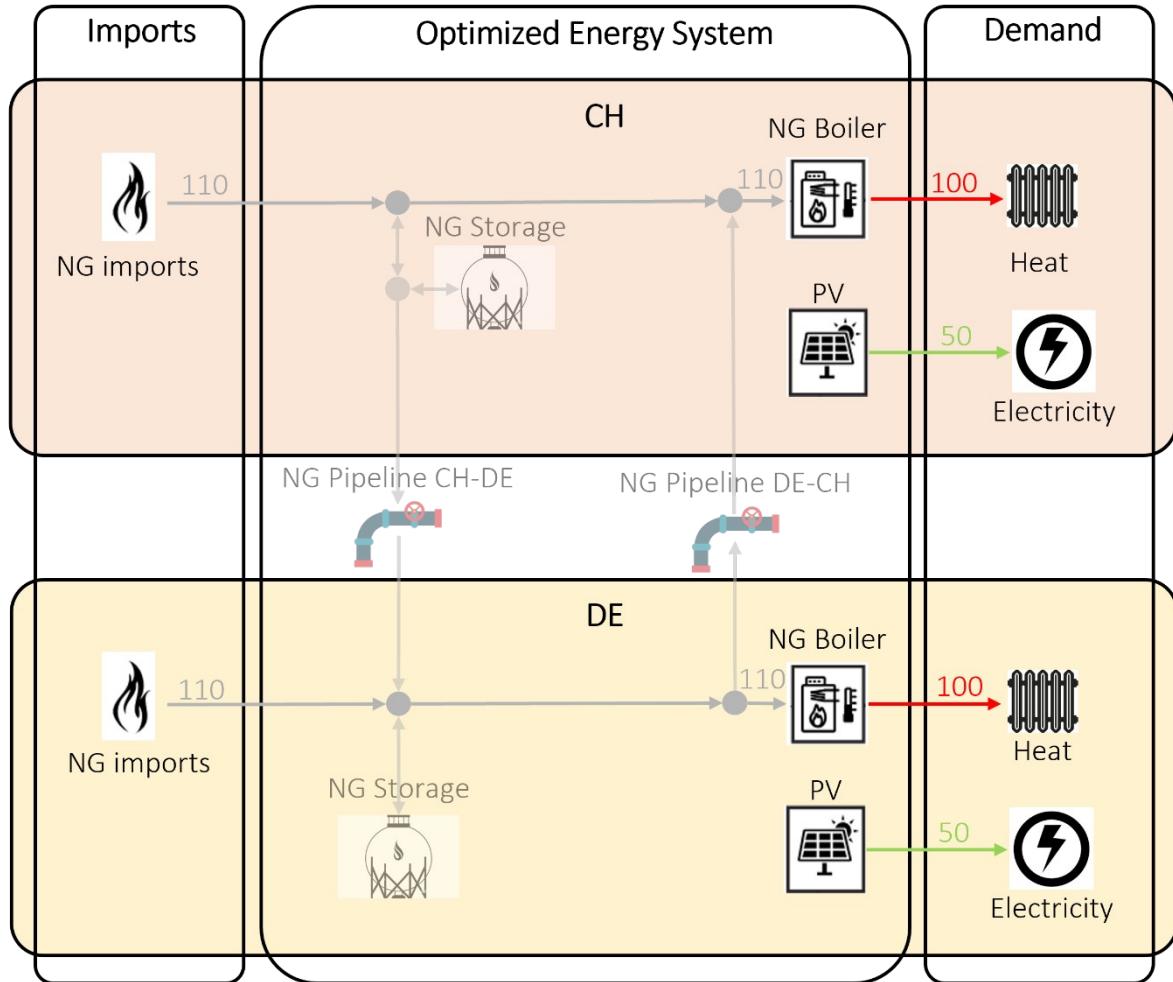


Figure 7: Energy system with newly introduced photovoltaics technology to satisfy electricity demand.

Therefore, the corresponding photovoltaics folder containing its attributes file must be created (Figure 8 left) to define the new technology. As the added technology generates electricity, it has to be defined as a carrier by creating a directory named electricity containing its *attribute.csv* file (Figure 8 middle). To introduce an electricity demand, the value of the *demand_default* parameter is set to 50 GW in electricity's *attributes.csv* file (Figure 8 right).

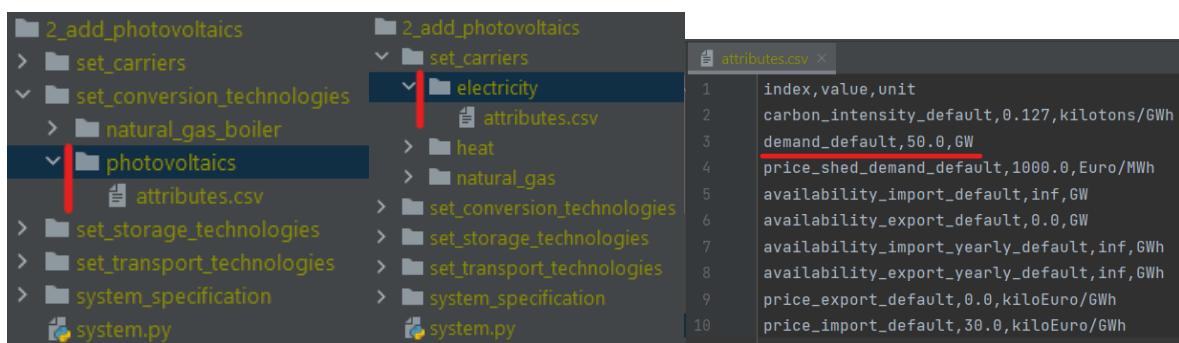


Figure 8 : (f.l.t.r.) Creation of photovoltaics folder with attributes file to define PV; introduction of electricity carrier; introduction of electricity demand.

After specifying photovoltaics as a technology and defining the carrier electricity, PV can be selected to be built by the optimizer by adding PV as a conversion technology in the *system.py* file (Figure 9).

```
14     system['set_conversion_technologies'] = ["natural_gas_boiler", "photovoltaics"]
15     system['set_storage_technologies'] = ["natural_gas_storage"]
16     system['set_transport_technologies'] = ["natural_gas_pipeline"]
17
18     system['set_nodes'] = ["DE", "CH"]
```

Figure 9: Selection of photovoltaics in *system.py* file.

Results

As a non-zero default electricity demand has been introduced in the *attributes.csv* file of electricity, PV is built at each node to meet the demand (Figure 10).

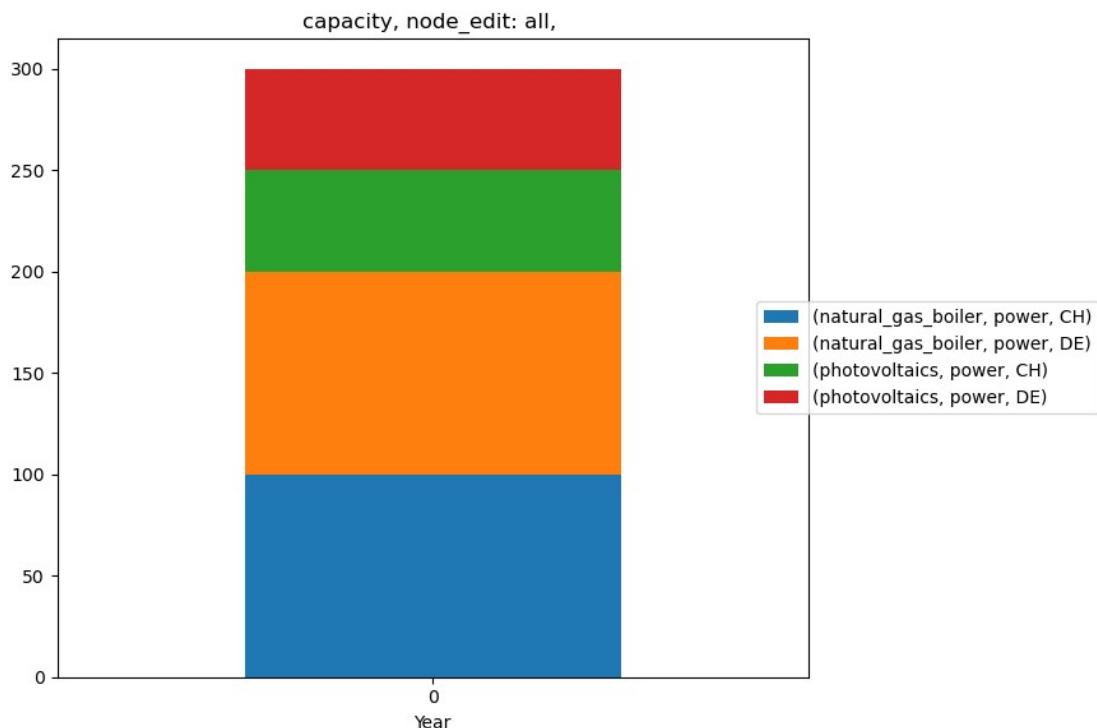


Figure 10: Installed technology capacities of PV and NG boilers at the nodes Germany and Switzerland.

3_multi_year_optimization

To modify time-related parameters of the optimization the values of the system's time variables can be adjusted (Figure 11). The most basic parameters are thereby the *reference_year*, the starting year of the optimization, the *optimized_years*, the number of years being optimized, and the *interval_between_years* which allows optimizing every n-th year (e.g., every second year for an interval of 2).

```
21 | system["reference_year"]          = 2023
22 | system["unaggregated_time_steps_per_year"] = 1
23 | system["aggregated_time_steps_per_year"]   = 1
24 | system["conduct_time_series_aggregation"] = False
25 |
26 | system["optimized_years"]           = 3
27 | system["interval_between_years"]    = 2
28 | system["use_rolling_horizon"]      = False
29 | system["years_in_rolling_horizon"] = 1
```

Figure 11: Modification of time-related parameters in the system.py file.

The way the parameters are defined in the screenshot's system.py file the years 2023, 2025 and 2027 will be optimized.

Results

The capacity plot now shows the technology capacities for the three years (Figure 12). Since our input data is constant in time, we cannot see any changes over the years (and the adjustment of the reference year and the interval between the years does not show an effect).

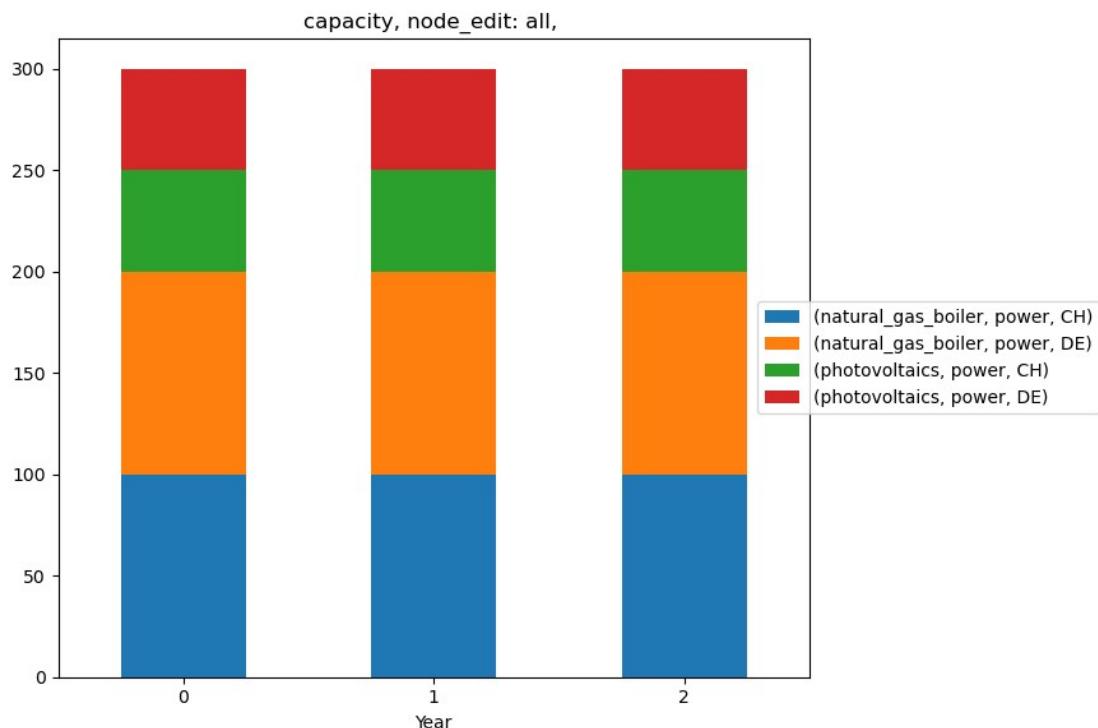


Figure 12: Installed technology capacities of PV and NG boilers at the nodes Germany and Switzerland.

4_variable_demand

In this example a first variable parameter is introduced by specifying a larger heat demand for Germany and a smaller one for Switzerland compared to the previous datasets (Figure 13).

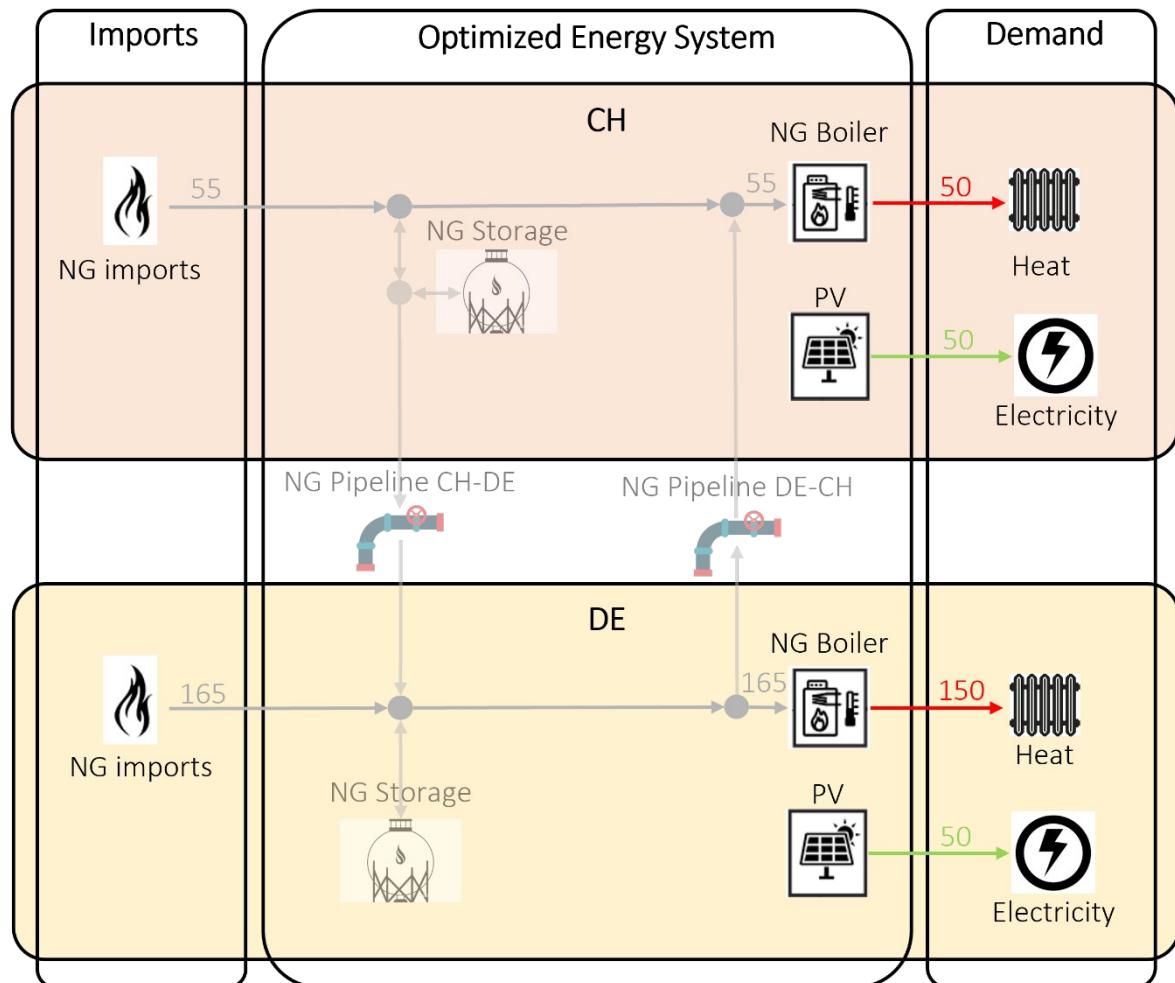


Figure 13: Energy system with different heat demands at the nodes Germany and Switzerland.

To introduce parameters which are variable in time or space, additional files are used. To do so, a `.csv` file named after the parameter must be created, e.g., `demand.csv` (Figure 14). The file is then used to define values for the parameter's index sets, e.g., introduce different heat demands for the nodes Switzerland and Germany. The exact procedure of creating such files is described in the “How to ZEN garden”.

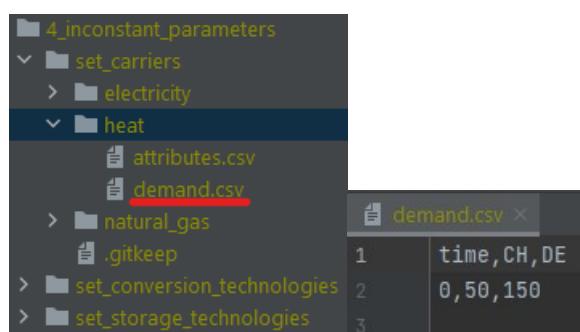


Figure 14: Definition of an inconstant heat demand by using the additional `demand.csv` file.

Results

The introduction of the individual heat demands of Germany and Switzerland effects the amount of built capacity at the corresponding nodes (Figure 15). As Germany's heat demand is now triple that of Switzerland, the capacity of German NG boilers is also triple that of Switzerland.

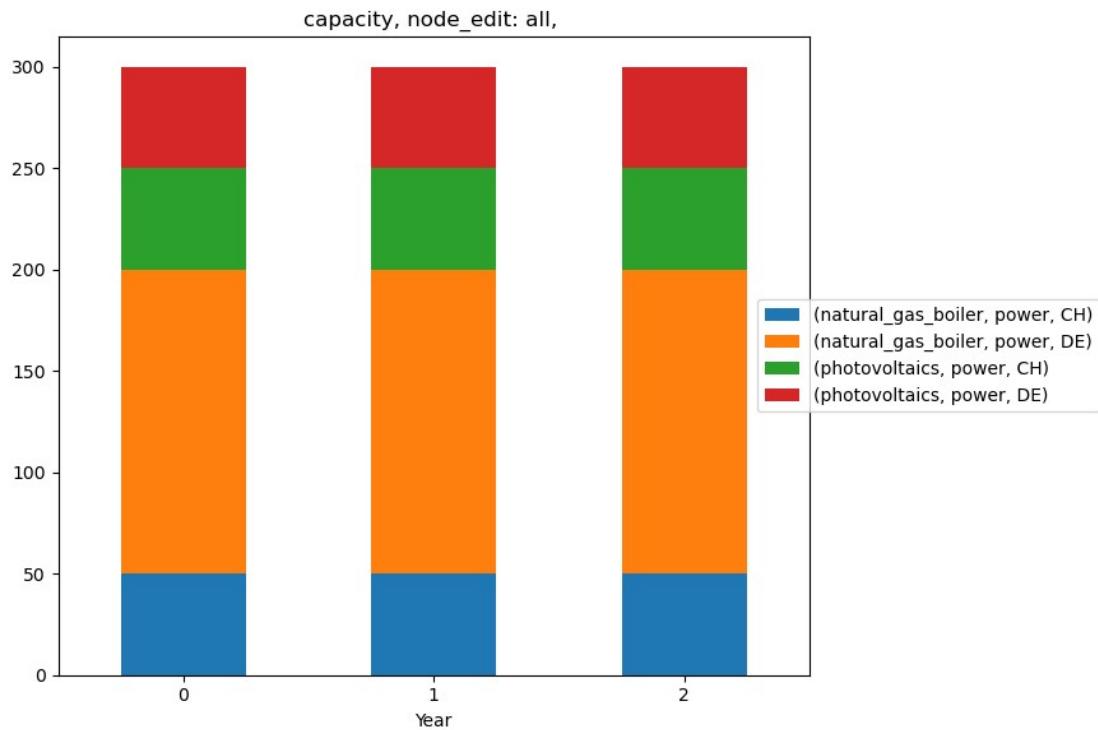


Figure 15: Installed technology capacities of PV and NG boilers at the nodes Germany and Switzerland.

5_reduced_import_availability

In this dataset the import availability of natural gas is constrained at the node Germany which results in the construction of a NG pipeline to compensate the missing natural gas (Figure 16).

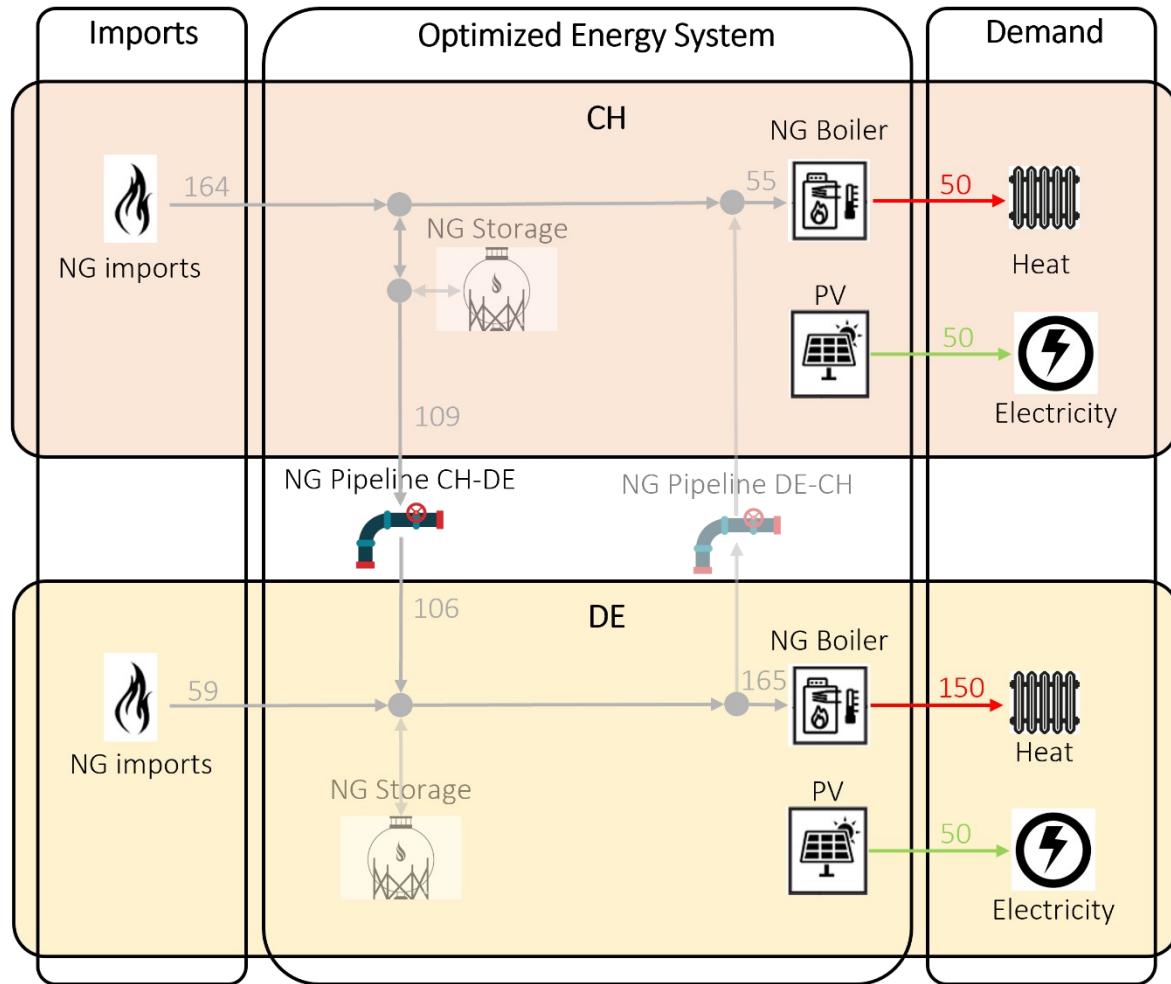


Figure 16: Energy system with reduced natural gas import availability at the node Germany which enforces the construction of an NG pipeline to transport the missing NG from Switzerland to Germany

Therefore, the import availabilities of the carrier natural gas are defined differently from the default value (Figure 17). In contrast to the default value which was set to infinity there now exists an import limitation. Whereas the one of Germany is set to 59.0, Switzerland can still import as much gas as needed since no value is assigned in the input file and thus gets the default value (inf). All the values of the other countries will not be used in the optimization but are allowed to be defined in the file anyway (this is generally the case). In this example, the import availability is constant in time.

	node, availability_carrier
1	AT, 0.83989200913242
2	BE, 0.006047488584474885
3	BG, 17.94294680921964
4	CZ, 0.21591335616438356
5	DE, 59.0
6	DK, 1.5811340182648401
7	EE, 11.808722256563529
8	EL, 16.605158085877132
9	ES, 13.656872527210941
10	

Figure 17: Additional file to define variable import availabilities of natural gas.

Results

As a result of the import limitation of natural gas in Germany, a pipeline must be built such that Germany has as much gas as needed to satisfy its heat demand (Figure 18). Figure 19 shows that the reduced import in Germany is substituted by Swiss imports (flows of carriers from outside the network) which are then transported through the pipeline to Germany.

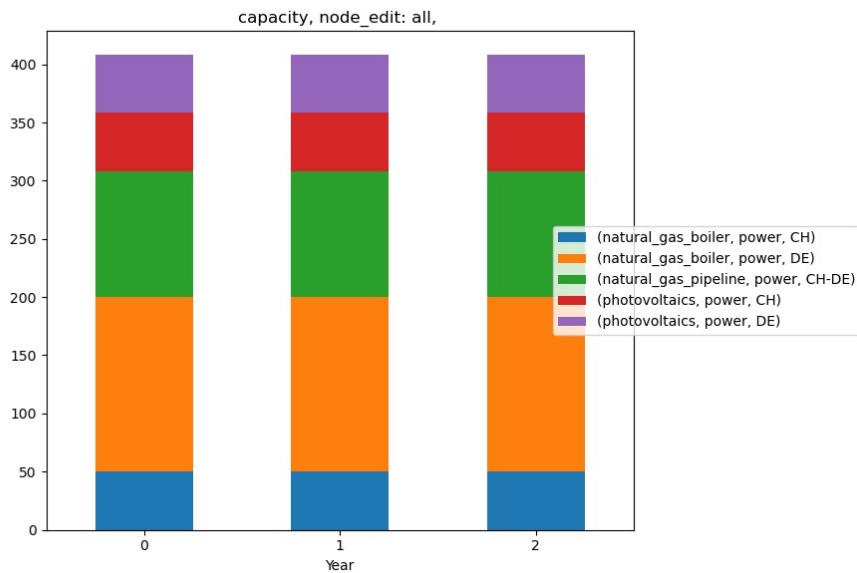


Figure 18: Installed technology capacities of PV, NG boilers and NG pipelines at the nodes Germany and Switzerland.

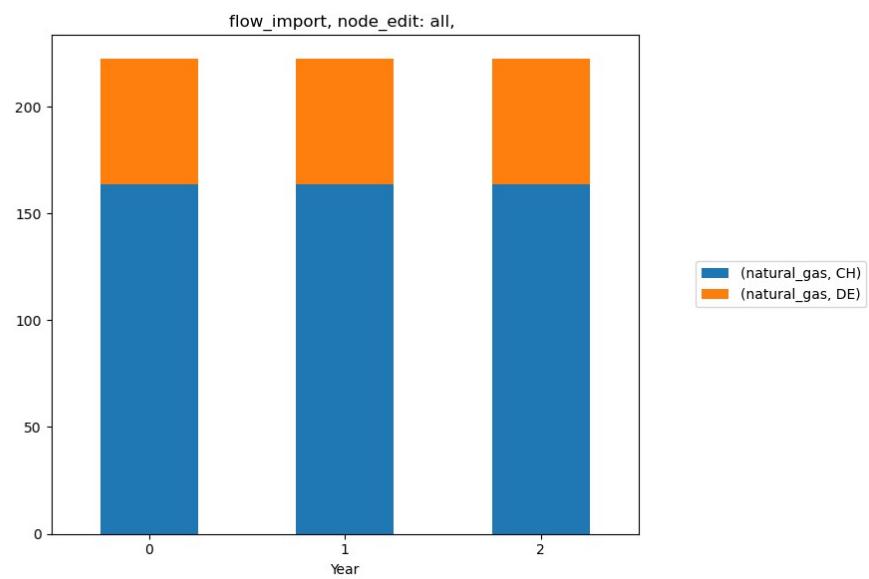


Figure 19: Import flow rates of natural gas at the nodes Germany and Switzerland.

6_PWA_conversion_technology

To account for the nonlinearities of conversion technologies, such as the conversion factor or capital expenditure, while keeping computational costs low, the piecewise affine approximation of non-linear functions can be used by defining two additional files in the corresponding folder (Figure 20 left). In this dataset, the nonlinearity of the natural gas boiler's conversion factor is defined in the *nonlinear_conversion_factor.csv* file (boiler becomes more efficient at higher loads) (Figure 20 right). To conduct the PWA, the *breakpoints_pwa_conversion_factor.csv* file specifies the number and the values of the breakpoints, i.e., in how many sections of linear approximations the non-linear function is divided (e.g., non-linear function should be approximated by four linear functions as in this dataset) (Figure 20 middle).



File	Content
nonlinear_conversion_factor.csv	1 natural_gas,heat 2 0,0 3 10,3.530819672 4 20,3.221639344 5 30,4.072459016 6 40,6.083278689 7 50,9.254098361 8 60,13.58491803 9 70,19.0757377 10 80,25.72655738 11 90,33.53737705 12 100,42.50819672 13 110,52.63901639 14 120,63.92983607 15 130,76.38065574 16 140,89.99147541
breakpoints_pwa_conversion_factor.csv	1 natural_gas 2 0 3 50 4 100 5 150 6 200 7 GW

Figure 20: (f.l.t.r.) Additional input files needed to define the PWA; file including the PWA breakpoints; file specifying the non-linear relation between the natural gas input and the heat output.

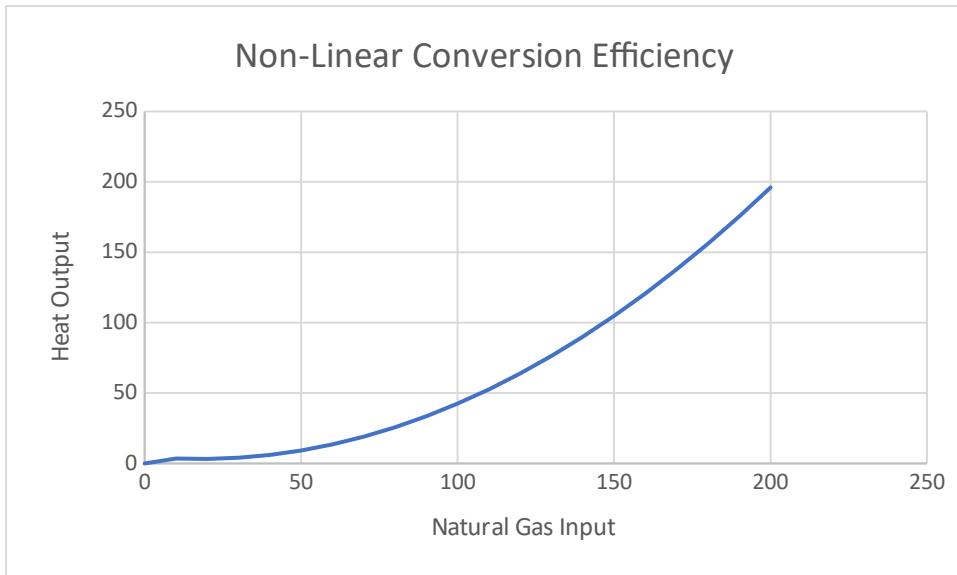


Figure 21: Non-linear conversion efficiency of the natural gas boiler.

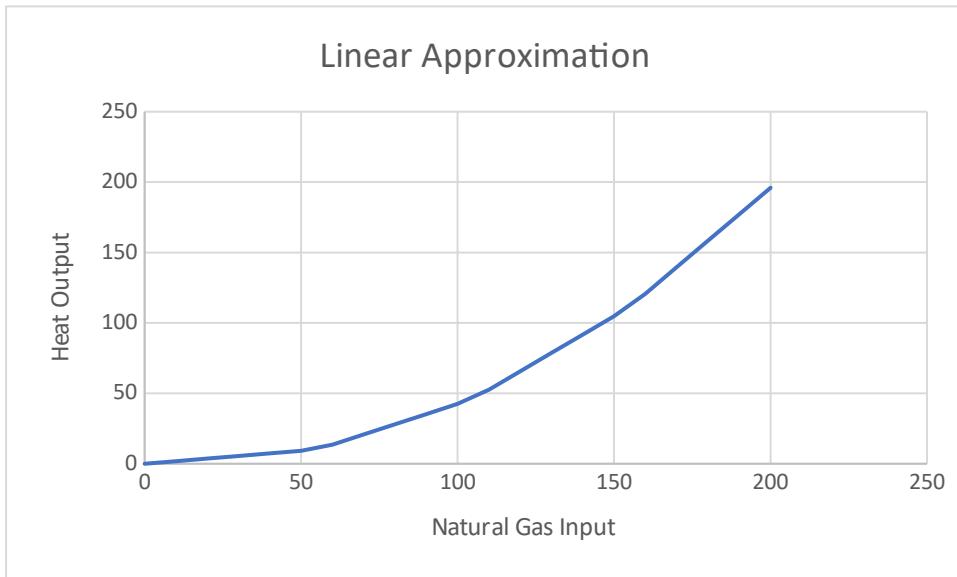


Figure 22: PWA approximation of the NG boiler's non-linear conversion efficiency by four linear functions.

Besides the conversion factor, the PWA feature can be used to model nonlinearities of technology capital expenditures.

Results

Since the boiler becomes more energy efficient for higher loads, the fraction of natural gas input and heat output does not increase linearly. Therefore, Switzerland needs proportionally more natural gas per produced unit heat than Germany does since the Swiss heat demand is lower than in Germany (Figure 23).

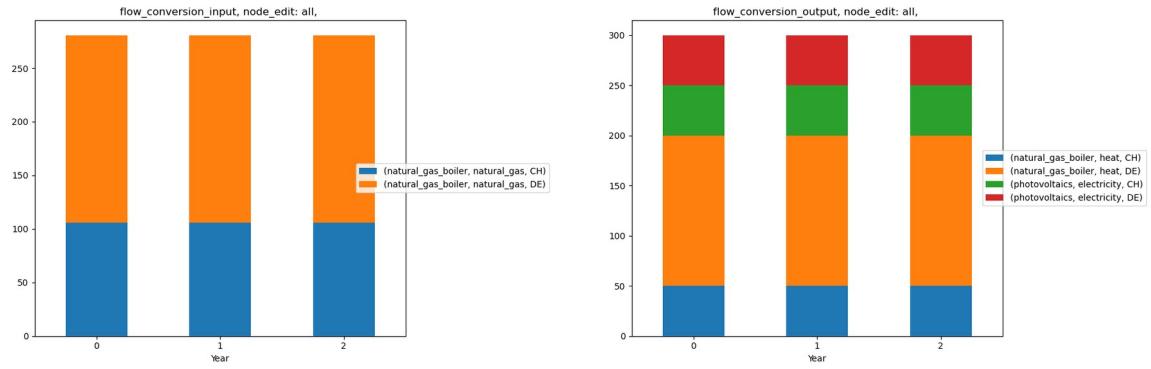


Figure 23: Natural gas input flows of the NG boilers at the nodes Germany and Switzerland; heat output flows of the natural gas boilers at the nodes Germany and Switzerland.

7_multiple_time_steps_per_year

So far, the example datasets did only account for a single time step per year. In this dataset, multiple time steps are optimized per year, of which the first one is represented in Figure 24.

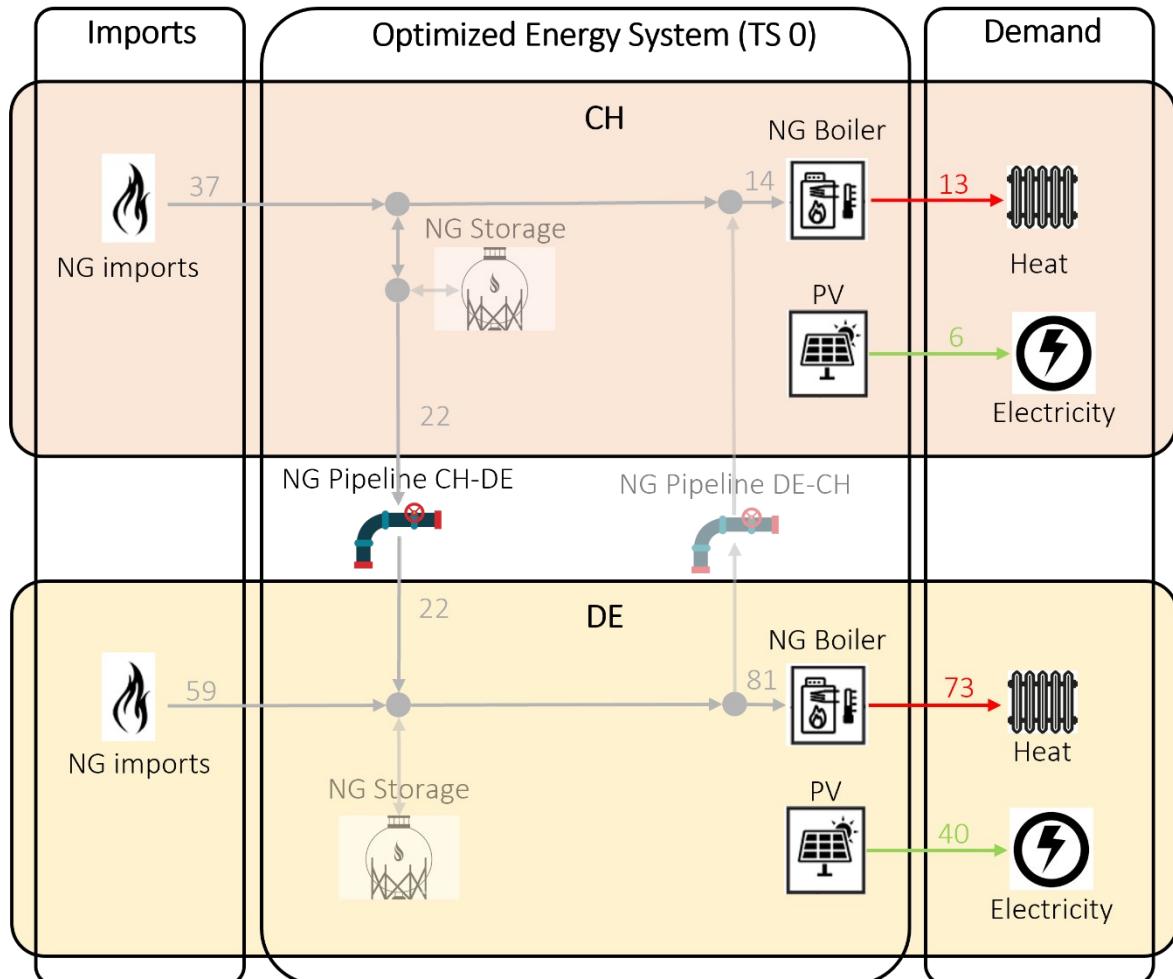


Figure 24: Energy system at the specific time step 0.

To optimize multiple time steps per year, e.g., 96 time steps to resolve the first four days of the year on an hourly basis, the system file must be modified as follows (Figure 25).

```

21 | system["reference_year"]          = 2023
22 | system["unaggregated_time_steps_per_year"] = 96

```

Figure 25: The number of time steps per year can be modified in the system.py file.

To make the dataset and the optimization's results more interesting, the electricity and heat demands are now defined on an hourly basis (operational time step) for the whole year, of which the first 96 will be used (Figure 26).

	demand.csv	
The file size (4.98 MB) exceeds the configured limit (2.56 MB). Code insight features are not available.		
1	time,AT,BE,BG,CH,CY,CZ,DE,DK,EE,EL,ES,FI,FR,HR,HU,IE,IT,LT,LU,LV,ME,MK,MT,NL,NO,PL,PT,R0,RS,SE,SI,SK,UK	
2	0,5.142944488259433,8.578937589170964,2.5364608671469817,5.77572091835615,0.3711869505759771,4.8129515458	
3	1,4.885802339784393,8.054878585068511,2.2681911019284806,5.850321116718984,0.3186869505759771,4.733925302	
4	2,4.628296241275497,7.488545868334284,1.9528333157732294,5.777156138624603,0.28918695057597715,4.62733368	
5	3,4.137246119044602,6.942666823948979,1.3708035536904614,5.631384286432207,0.27868695057597714,4.26530920	
6	4,3.5043656312129334,6.438547425751148,1.273830448874755,4.717153227037052,0.27768695057597714,3.65742392	
7	5,3.5894975135782174,6.37422141550948,1.3066676903419403,4.518906774498279,0.28268695057597715,3.59981557	
8	6,3.6929675772451955,6.563475671953827,1.3400393699787119,4.566954765529877,0.28168695057597715,3.6014807	
9	7,3.9275026115833835,6.566773202996899,1.5755568485174056,4.5573509299723,0.27968695057597714,3.6553669	
10	8,4.208880244135431,6.854905942780472,1.8776602583360453,4.495785702305753,0.26318695057597713,3.77512355	
11	9,4.572151315750297,7.171454202172119,2.047197160850959,4.766925201029963,0.25918695057597714,4.2333933863	
12	10,4.940762073568329,7.5351645487553425,2.105796423218659,4.983610418447373,0.28568695057597715,4.6913130	
13	11,5.169478449078552,7.8071024414785,2.0628143220305244,5.221580153181999,0.3924869505759771,4.867783435	
14	12,5.104488522800641,8.033247836873728,1.9297284926334735,5.066564962789716,0.2956869505759771,4.84196924	
15	13,4.977740913995955,7.759866552560632,1.796829903670524,5.0958345517562185,0.3061869505759771,4.9241624	
16	14,4.854788236422079,7.51418211569373,1.7424646419049439,5.386094391708094,0.33068695057597713,4.81416311	
17	15,4.842202156747643,7.5637133288382,1.8920175477511987,5.236957732348776,0.3736869505759771,4.8083345134	
18	16,5.046754728501253,7.735271806763177,5.3862125187682916,4.82585752139875,0.45968695057597714,4.9129764	

Figure 26: Hourly electricity demand defined at different nodes and at each hourly time step of the year.

Results

The built technologies differ in comparison with the previous datasets due to the changes in heat and electricity demand (Figure 27).

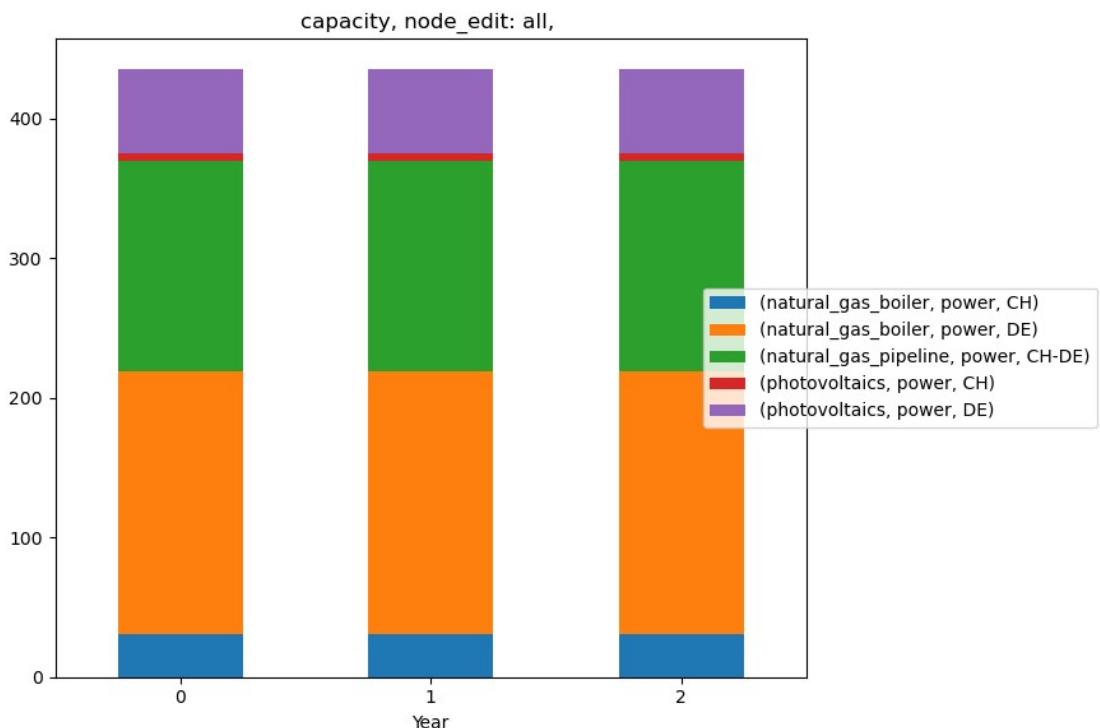


Figure 27: Installed technology capacities of PV, NG boilers and NG pipelines at the nodes Germany and Switzerland.

The output flow rates of the conversion technologies now change hourly since the heat and electricity demand are defined on an hourly basis (Figure 28 and Figure 29).

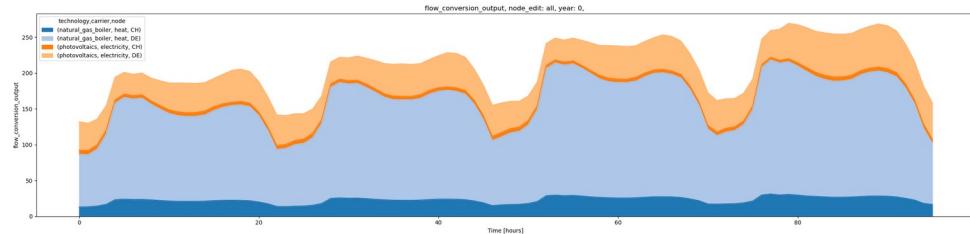


Figure 28: Output flow rates of the conversion technologies PV and NG boilers at the nodes Germany and Switzerland.

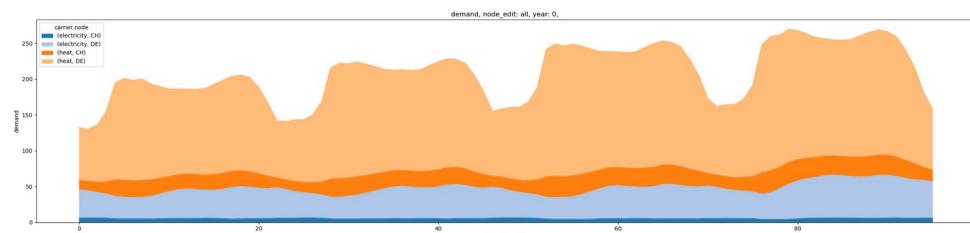


Figure 29: Heat and electricity demands at the nodes Germany and Switzerland.

8_reduced_import_availability_yearly

So far storage technologies are never built during the optimization. To enforce a situation in which natural gas storages become financially attractive, the yearly import availability of natural gas is constrained in this dataset. Additionally, heat pumps are introduced to provide an alternative way of generating heat without using natural gas (Figure 30).

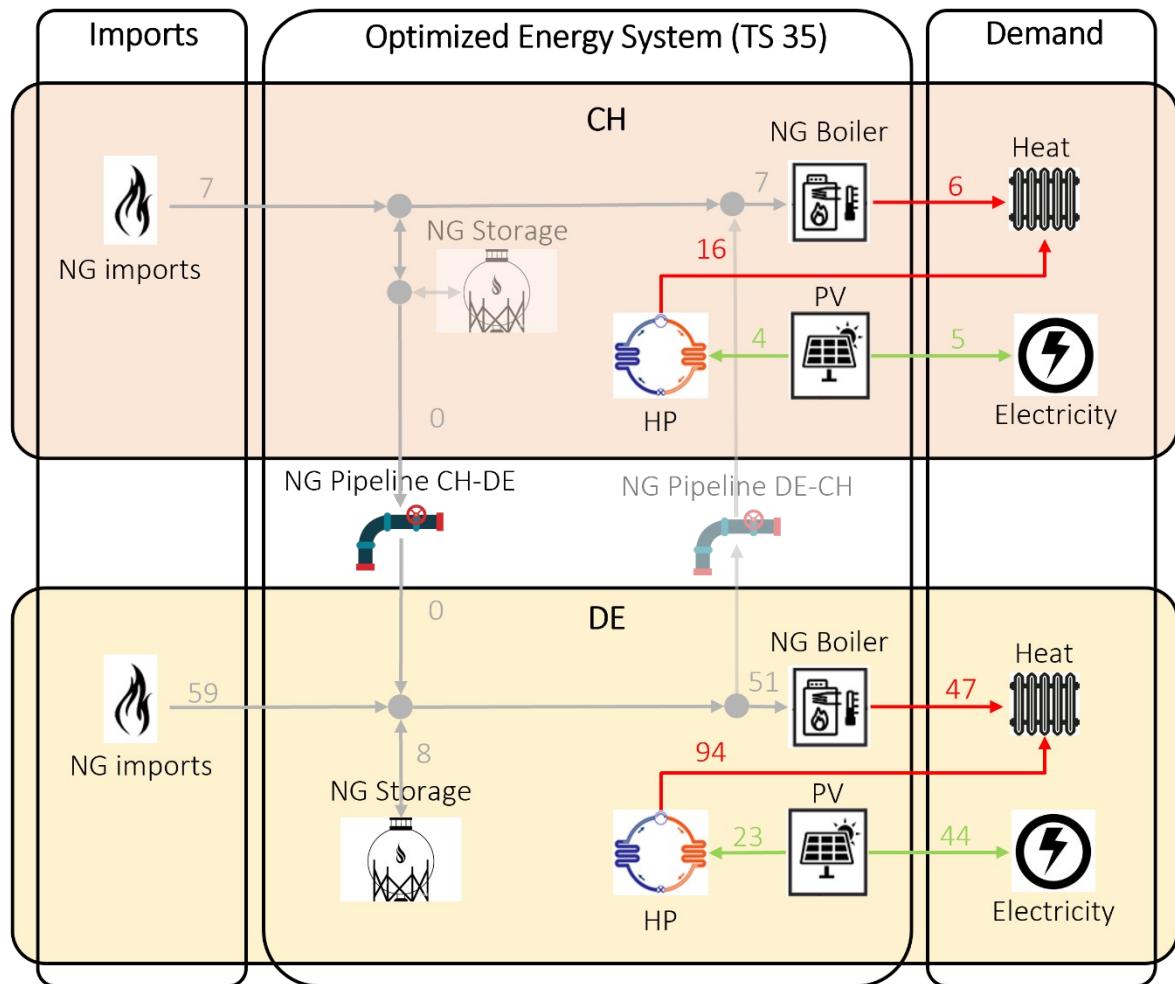


Figure 30: Energy system with constrained yearly import availability of natural gas enforcing the construction of a NG storage and newly introduced heat pumps at time step 35.

The yearly import availability of natural gas is constrained for Switzerland in the years 2024 and 2025 by creating an additional input file for the `availability_import_yearly.csv` parameter of the carrier natural gas (Figure 31 left). Since heat pumps use electricity to transfer heat from a lower temperature to a higher temperature, they must be defined in the conversion technology folder (Figure 31 right).

```

8_reduced_import_availability_yearly
  availability_import_yearly.csv
  > set_carriers
  > set_conversion_technologies
    > heat_pump
      attributes.csv

```

1	node, 2023, 2024, 2025
2	CH, inf, 80, 40

Figure 31: csv file containing yearly import availability of natural gas in Switzerland; definition of the heat pumps technology.

Results

Due to the yearly limits in importing natural gas in Switzerland, Germany can no longer compensate its needed natural gas by pipeline transport from Switzerland and thus builds NG storage capacities (Figure 33). In addition, some heat pumps are installed (Figure 33).

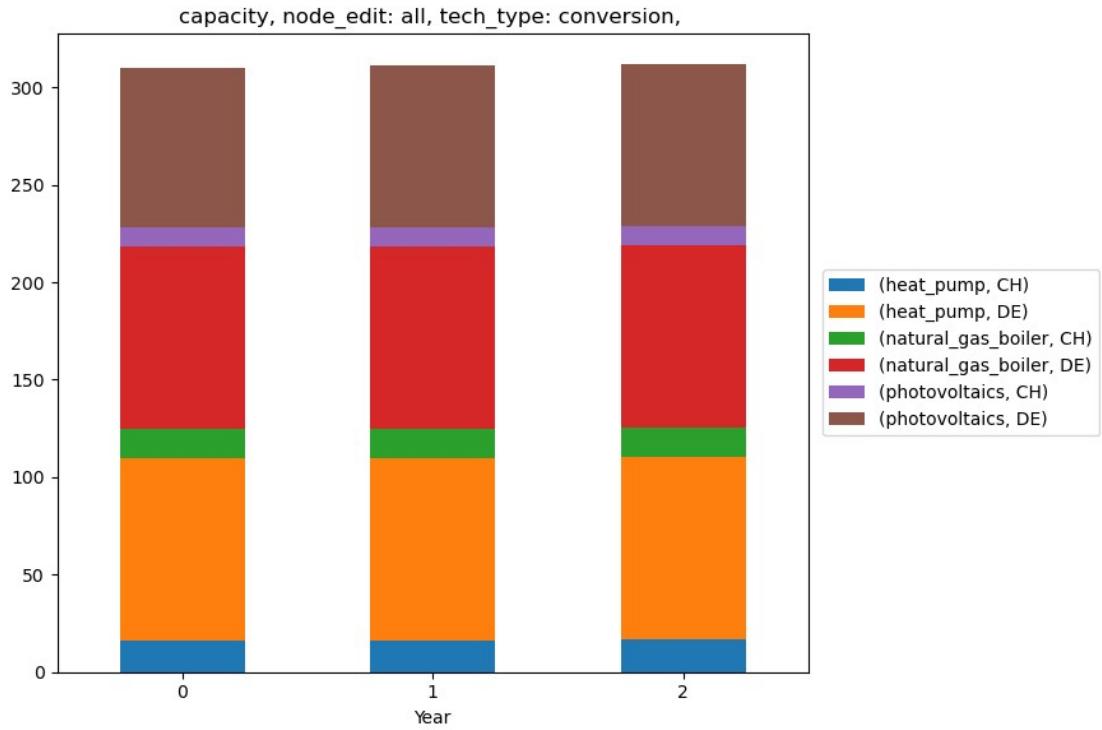


Figure 32: Installed technology capacities of PV, NG boilers and heat pumps at the nodes Germany and Switzerland

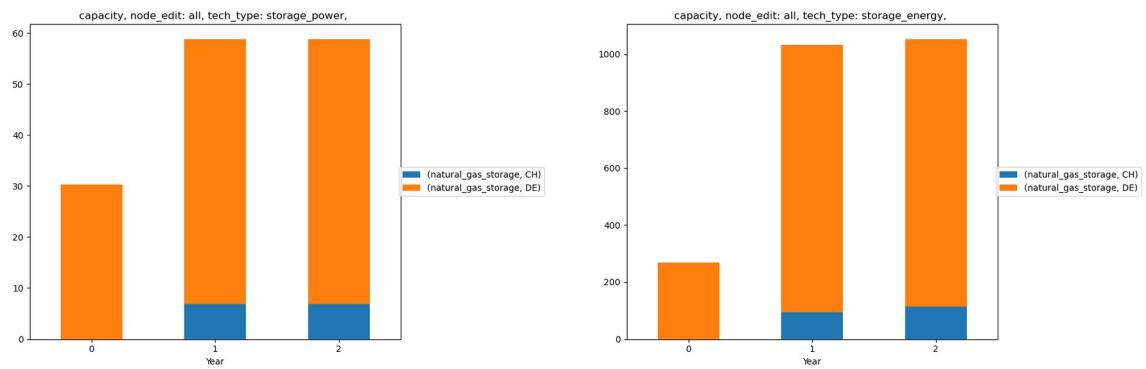


Figure 33: Installed technology power capacities of NG storages; installed technology energy capacities of NG storages.

Figure 34 shows the energy balance of natural gas in Germany at all 96 time steps.

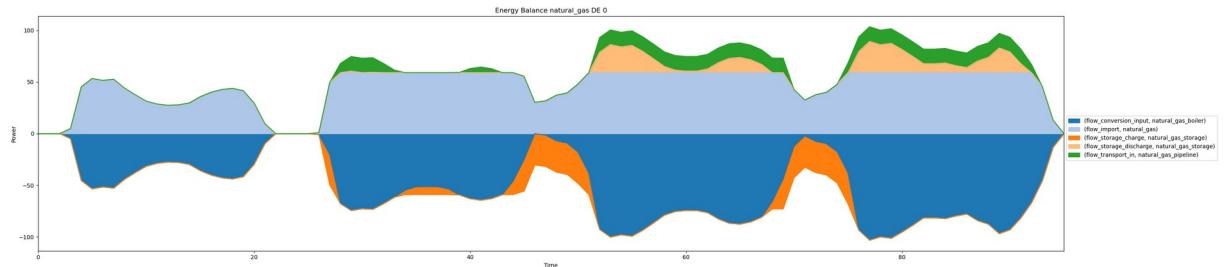


Figure 34: Energy balance of natural gas at the node Germany showing the different parts of import, consumption, storage, transport and demand.

In addition, Figure 35 shows the energy balance of heat in Germany.

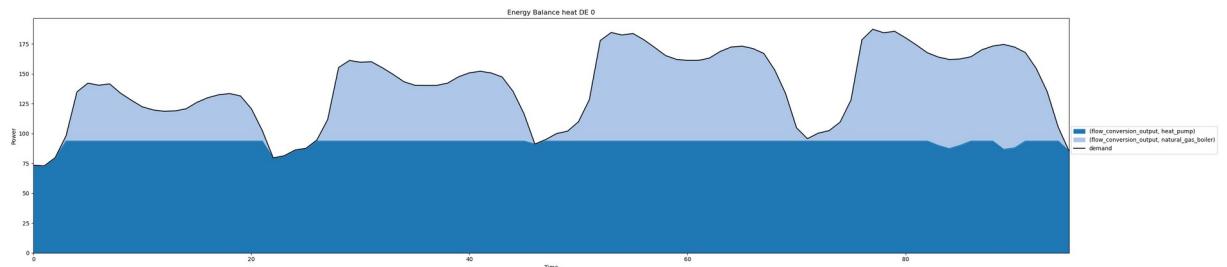


Figure 35: Energy balance of heat at the node Germany showing the different parts of production and demand.

9_time_series_aggregation

In this example the usage of the time series aggregation (TSA) is shown by aggregating the 96 time steps of the previous dataset example to 5 time steps (Figure 36).

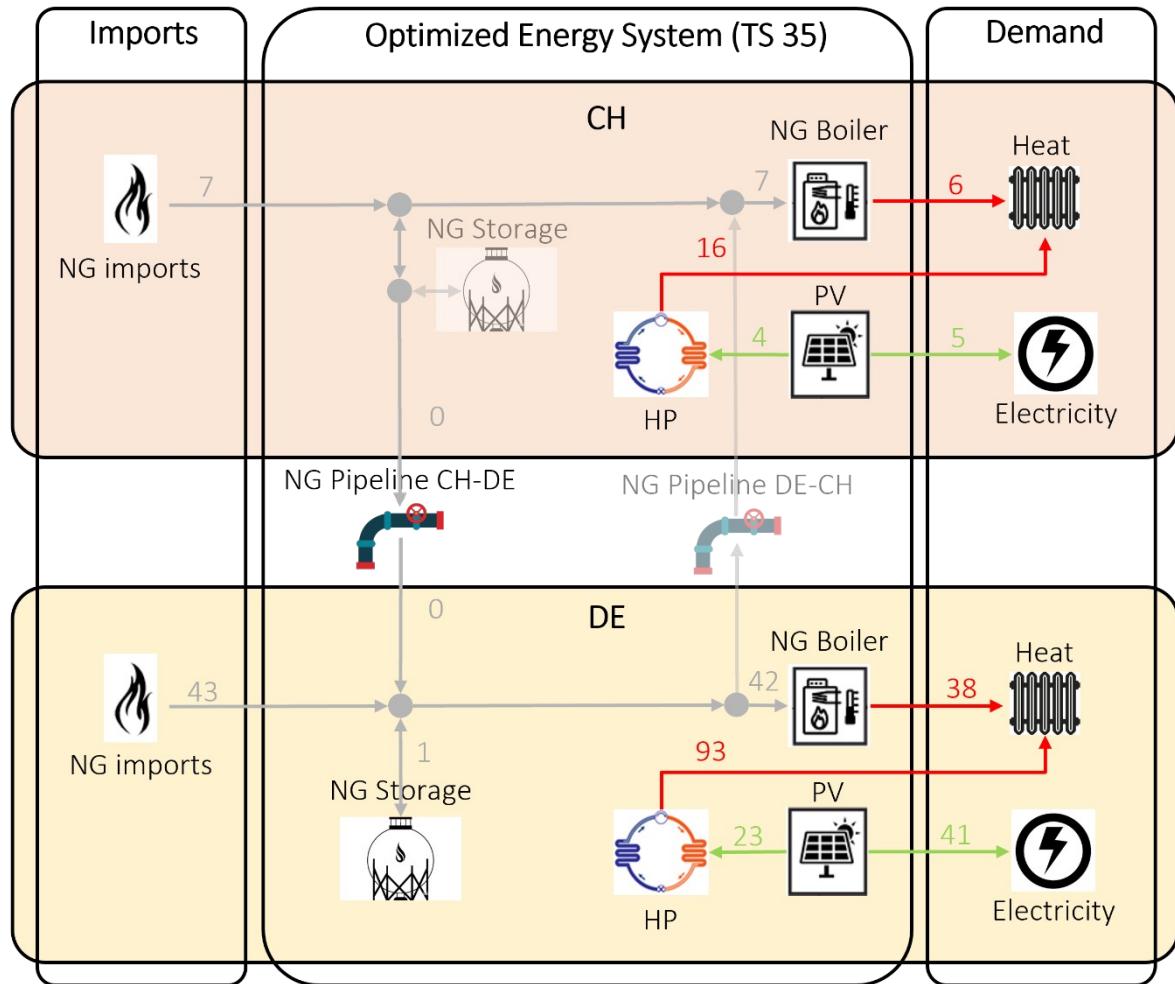


Figure 36: Energy system optimized using timely aggregated parameter data.

Since the complexity of solving an optimization problem increases significantly with the number of time steps, computational efforts can explode quite fast. Therefore, time series aggregation is used to overcome long runtimes while keeping the precision of the optimization problem's results accurate. To specify how many distinctive time steps of the underlying input data should be used to define the optimization problem, the following adjustments have to be made. Here, the same 96 values as in the previous dataset are considered, but the time series aggregation reduces the problem to five distinctive time steps before the optimization problem is solved (Figure 37).

```

21     system["reference_year"]          = 2023
22     system["unaggregated_time_steps_per_year"] = 96
23     system["aggregated_time_steps_per_year"]   = 5
24     system["conduct_time_series_aggregation"] = True
  
```

Figure 37: Activation of time series aggregation to 5 time steps in system.py file.

Results

By comparing the energy balance of natural gas for the unaggregated case (Figure 38) and the aggregated case (Figure 39) a relatively large difference caused by the time series aggregation can be seen.

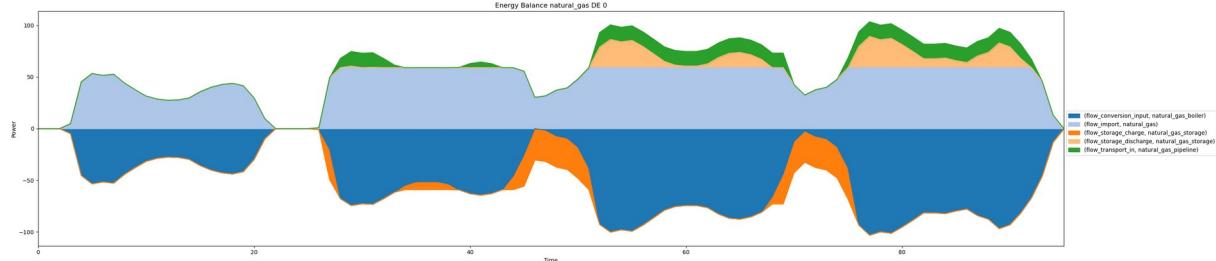


Figure 38: Unaggregated energy balance of natural gas at the node Germany.

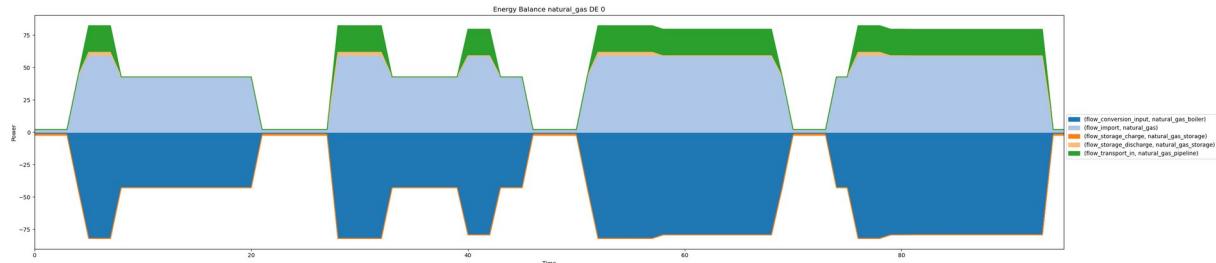


Figure 39: Aggregated energy balance of natural gas at the node Germany (5 time steps).

To get a result which is more similar to the unaggregated case, the dataset is optimized again with ten aggregated time steps per year (Figure 40).

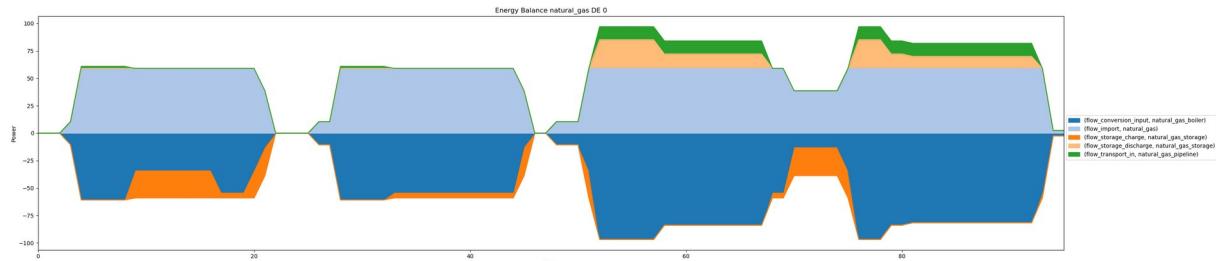


Figure 40: Aggregated energy balance of natural gas at the node Germany (10 time steps).

While the run time to solve dataset 9 does not differ significantly in comparison to the run time of dataset 10, the effect can be experienced by increasing the number of unaggregated time steps per year to, e.g., 8760 (full year) and solving the problem with and without time series aggregation.

10_full_year

In this example a full year is optimized in hourly time resolution (Figure 41).

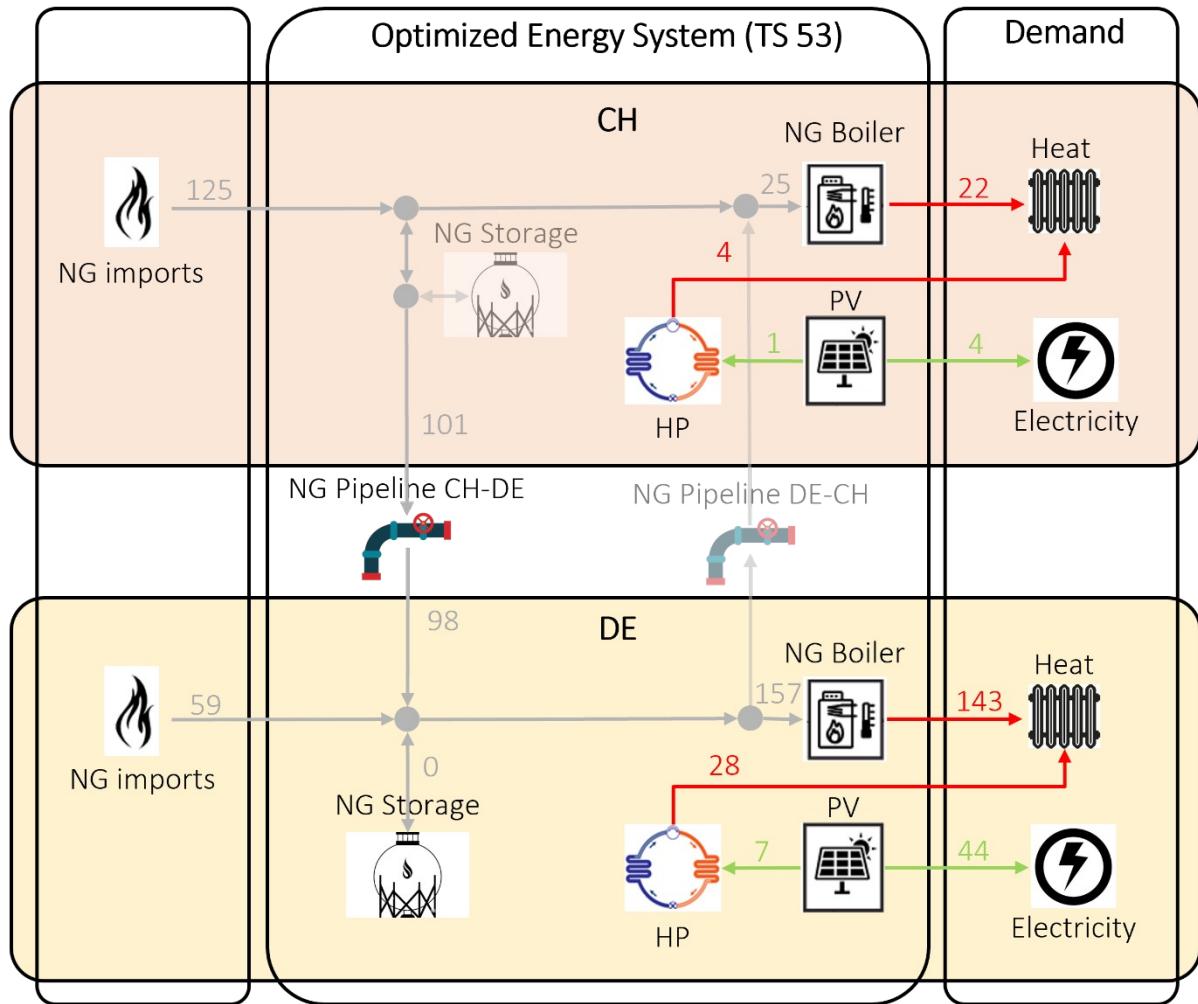


Figure 41: Energy system optimized for a full year at time step 53.

To solve the optimization problem with a complete hourly year resolution, i.e., 8760 time steps per year, time series aggregation is used again to keep computational costs low (to experience the difference in runtime, the time series aggregation can be switched off) (Figure 42).

```

21 system["reference_year"]          = 2023
22 system["unaggregated_time_steps_per_year"] = 8760
23 system["aggregated_time_steps_per_year"]   = 10
24 system["conduct_time_series_aggregation"] = True

```

Figure 42: System.py file adaptions for full year optimization with TSA.

Results

The energy balances now show the yearly changes in heat (Figure 43) and electricity (Figure 44) demand (here for the first 8760 hours, i.e., the first year).

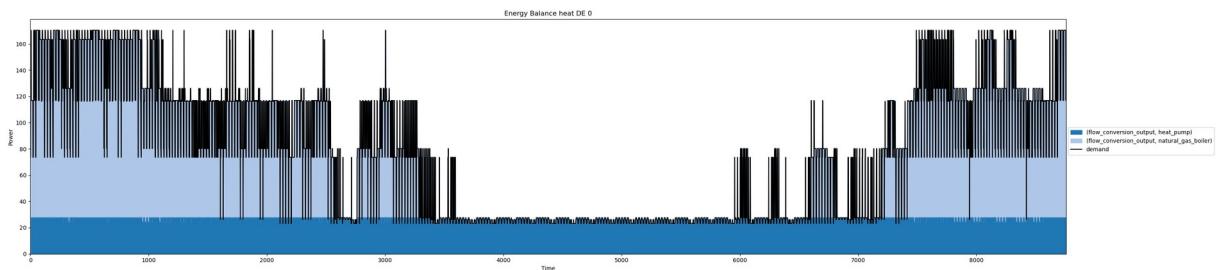


Figure 43: Energy balance of the carrier heat at the node Germany in year 0 (2023).

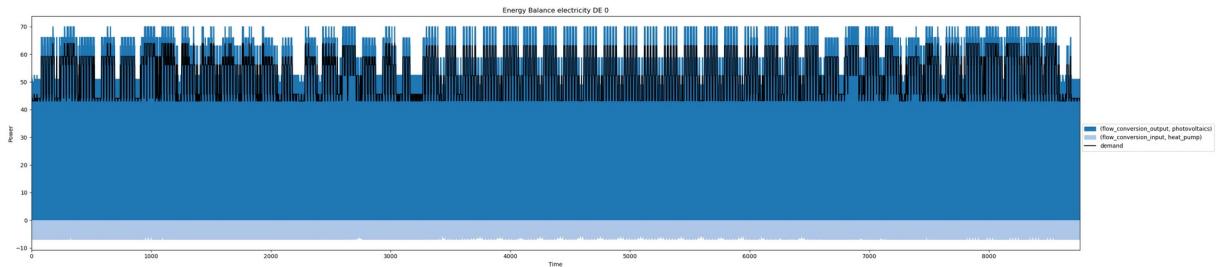


Figure 44: Energy balance of the carrier electricity at the node Germany in year 0 (2023).

11_yearly_variation

This dataset demonstrates how yearly changing parameters can be defined more easily.

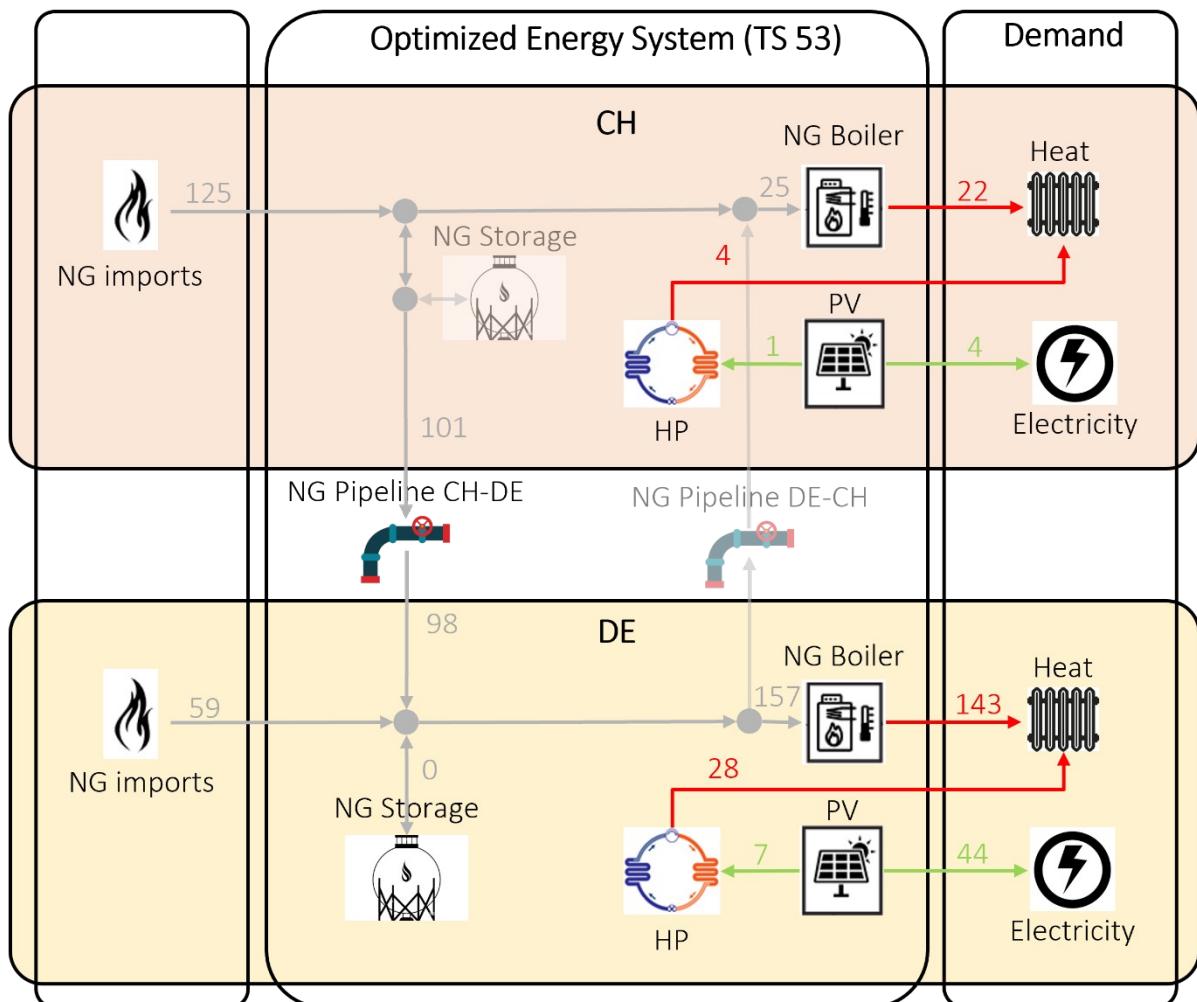


Figure 45: Resulting flow rates of energy system with yearly changing parameters.

To facilitate the consideration of changes in parameters with hourly time steps over several years (e.g., carrier prices), their yearly percentual change can be defined in additional input files. Therefore, the hourly input data must be provided for a single year only and not for every upcoming year. In this dataset, the yearly change in the import price of natural gas is defined by the file *price_import_yearly_variation.csv* (add *_yearly_variation* ending to parameter name) (Figure 46). As there are no time- or node-dependent import prices for natural gas defined in this dataset, the default value from the attributes file is scaled according to the factors specified in the yearly variation file. To get the import price of a specific year, the default value will be multiplied by the factor of the corresponding year (factors are not the percentual increase between the years).

1	year,import_price_carrier
2	2023,1.0
3	2024,1.04
4	2025,1.09
5	2026,1.14
6	2027,1.18
7	2028,1.25
8	2029,1.31
9	2030,1.36
10	2031,1.39
11	2032,1.41

Figure 46: Definition of yearly changing parameters by creating an additional input file.

Additionally, the yearly change in electricity demand is regarded as well. In contrast to the natural gas prices, the electricity demand is specified for time and nodes (again the input data for the countries not regarded in the optimization is allowed to be stated although it is not used) (Figure 47). Since there is no yearly change defined for Switzerland, its electricity demand will stay the same.

1	node,2022,2023,2024,2025,2026,2027,2028,2029,2030,2031,2032,2033,2034,2035,2036
2	AT,1,1.032576018,1.065152036,1.097728054,1.130304071,1.167081982,1.203859892,1.240637802,1.277415712,1.314193622,1.347778624,1.381363627,1.414948629,1.448533632,1.482118634
3	BE,1,1.022538391,1.045076781,1.067635172,1.090153562,1.111283157,1.132412751,1.153542346,1.174671941,1.198801535,1.233710506,1.271619477,1.309528448,1.347437419,1.385463639
4	BG,1,1.024453267,1.048906534,1.073359802,1.097813069,1.11625122,1.134689372,1.153127523,1.171565675,1.198003826,1.219174478,1.277515781,1.306686432,1.335857084
5	CY,1,1.041072193,1.082144380,1.12321658,1.164288773,1.197417435,1.230546097,1.263674759,1.299883421,1.329932083,1.380629012,1.431325941,1.482022871,1.5327198,1.58341673
6	CZ,1,1.027104119,1.054208238,1.081312358,1.108841647,1.146048876,1.185681043,1.221313320,1.258945608,1.296577891,1.326088851,1.355599891,1.385110769,1.414621728,1.444132687
7	DE,1,1.030163216,1.060326431,1.090489647,1.120652865,1.146967805,1.173282747,1.199597689,1.225912631,1.252227573,1.281276081,1.31032459,1.339373099,1.368421667,1.3974570116
8	DK,1,1.051099265,1.102198529,1.153297794,1.204397058,1.241555858,1.278714057,1.315873457,1.353032260,1.390191050,1.422784794,1.455378532,1.487972271,1.520560609,1.553159748
9	EE,1,1.009735356,1.019470712,1.029206667,1.03894123,1.061480441,1.084019459,1.106558477,1.129099749,1.151636512,1.181456676,1.21127684,1.241097004,1.270917168,1.300737332
10	ES,1,1.019195828,1.038391655,1.057587483,1.07678331,1.107290891,1.137798472,1.168380454,1.19883635,1.229321216,1.273475282,1.317573382,1.361700131,1.405826436,1.449952741
11	FI,1,1.04028409,1.080568179,1.120852269,1.161136358,1.221652607,1.282168976,1.342688284,1.40320193,1.405717902,1.477916957,1.4972116012,1.506315068,1.520914123,1.534713178
12	FR,1,1.0201430368,1.04291163,1.085715151,1.141319249,1.169118098,1.174411947,1.224715790,1.255740236,1.286764676,1.317789110,1.346813557,1.379837997
13	EL,1,1.013141915,1.02628383,1.039425745,1.052567059,1.08426335,1.11608501,1.147723085,1.17944236,1.211161035,1.257493451,1.303825868,1.350158284,1.396490701,1.442823117
14	HR,1,1.015177033,1.030354065,1.045531098,1.06070813,1.07359162,1.086370125,1.112032254,1.124863287,1.181029449,1.23719561,1.293361772,1.349527933,1.405694094
15	HU,1,1.029369714,1.058739428,1.088109141,1.117478855,1.1401793,1.162879745,1.18585019,1.20280635,1.23098108,1.269982209,1.308923337,1.347894460,1.380805594,1.425830723
16	IE,1,1.060182182,1.120364365,1.180546547,1.24072873,1.273695822,1.306662914,1.339630000,1.372597098,1.40556419,1.447193513,1.488822835,1.530452158,1.57208148,1.613710803
17	IT,1,1.00872409,1.01744818,1.02617227,1.03489636,1.062555218,1.098214077,1.117872936,1.145531795,1.171390854,1.213784142,1.254377631,1.294971119,1.335564608,1.376158096
18	LT,1,1.020619429,1.061853885,1.082477715,1.088921089,1.098920775,1.096104602,1.099511149,1.134669973,1.169828798,1.204987622,1.246140446,1.275305271
19	LU,1,1.022971725,1.04594345,1.068915175,1.0918869,1.125944441,1.160001982,1.194059523,1.228117064,1.262174605,1.311903009,1.360831413,1.410159816,1.45948822,1.508816624
20	LV,1,1.021883565,1.04376713,1.065650695,1.08753426,1.107772248,1.128020237,1.148263226,1.168500215,1.188749204,1.21877411,1.248799017,1.278823924,1.308840831,1.338873738
21	MT,1,1.02971175,1.04594345,1.068915175,1.0918869,1.125944441,1.160001982,1.194059523,1.228117064,1.262174605,1.311903009,1.360831413,1.410159816,1.45948822,1.508816624
22	NL,1,1.070951175,1.141902535,1.212853525,1.2388047,1.331702514,1.379900328,1.427498124,1.475395955,1.523293769,1.573505308,1.623716847,1.673928386,1.724439295,1.774351464
23	PL,1,1.029843662,1.059687204,1.089530805,1.119374407,1.158946883,1.182519359,1.214691835,1.24566431,1.277236786,1.338235987,1.399235188,1.460234388,1.521233589,1.58223279
24	PT,1,1.015639166,1.031278212,1.046917319,1.062556425,1.100952656,1.139348871,1.177745118,1.216141349,1.25453758,1.29261171,1.330684763,1.368758354,1.444905537

Figure 47: Yearly variation in electricity demand at different nodes.

Results

To demonstrate the effect of the yearly variation in natural gas import prices, ten years are optimized so that the change in the carrier cost can be seen more clearly (Figure 48).

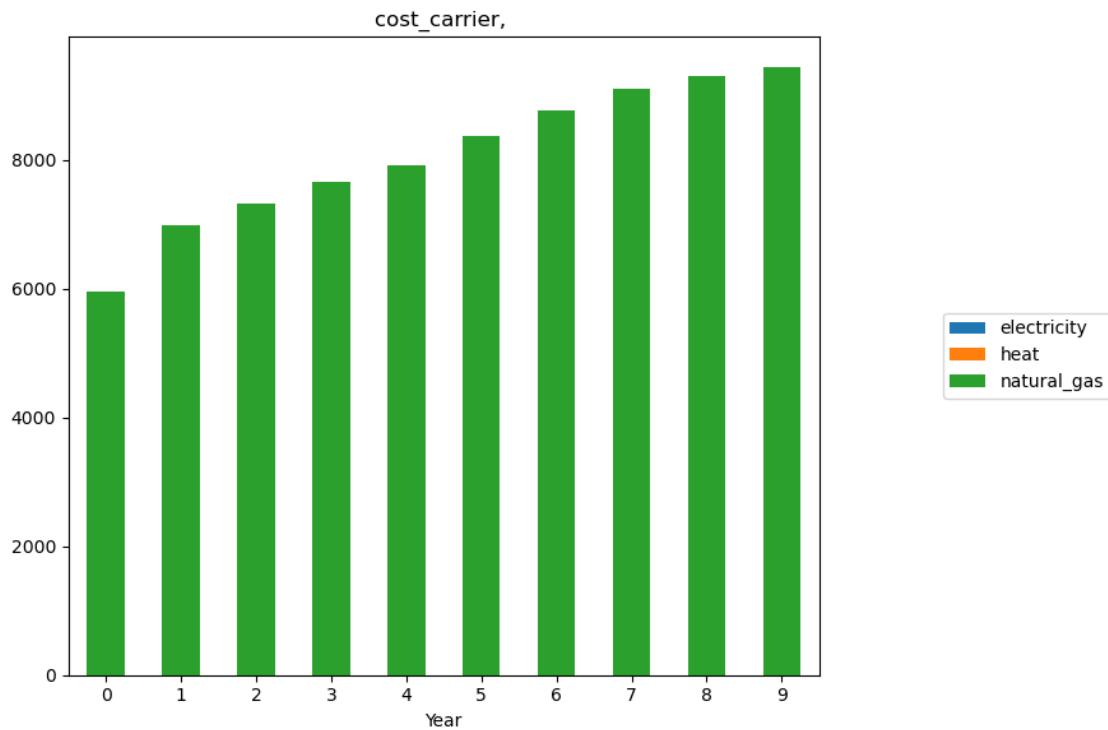


Figure 48: The yearly carrier costs of natural gas increase over the years.

Figure 49 shows the yearly increase in electricity production by photovoltaics due to increasing electricity demand (left) since the electricity input flows of the heat pumps (only other electricity consumer in the energy system) remain constant over the years (right).

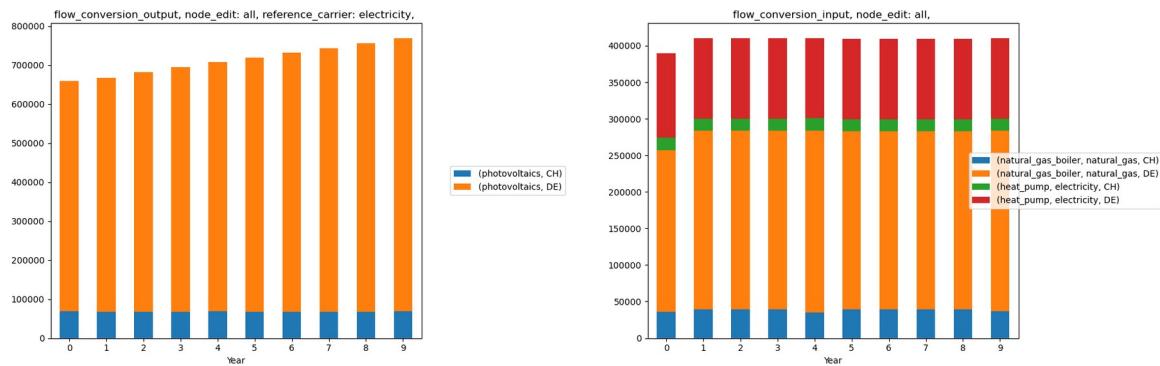


Figure 49: Total yearly output flows of photovoltaics increase over years (left) whereas total yearly input flows of electricity consuming conversion technologies (heat pumps) remain constant.

12_myopic_foresight

All the previous datasets are optimized using so-called perfect foresight, i.e., all years are optimized at once with the assumption that all the future parameter data are known at the time the optimization is conducted. In this example, however, myopic foresight is demonstrated, where the knowledge of future parameter data, the foresight horizon, is limited.

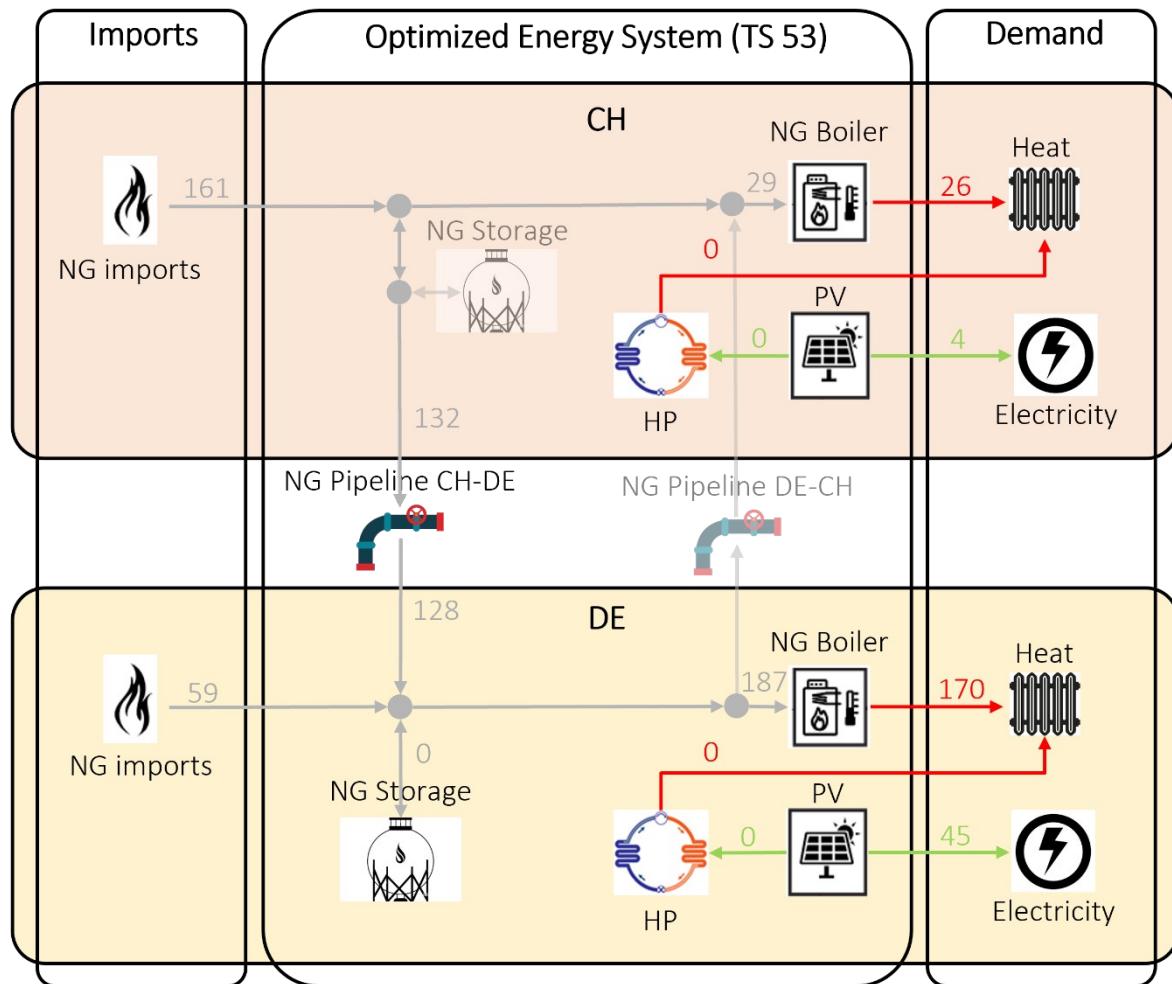


Figure 50: Energy system optimized with myopic foresight at time step 53.

So far, the defined energy system has been optimized for all its years simultaneously (perfect foresight = assumption that all the future parameter data are known already in the starting year). In contrast, the so-called rolling horizon (myopic foresight, see [myopic foresight discussion](#)) optimization approach can be activated as it is done in Figure 51.

```

26     system["optimized_years"]           = 10
27     system["interval_between_years"]    = 1
28     system["use_rolling_horizon"]       = True
29     system["years_in_rolling_horizon"] = 1
  
```

Figure 51: Activation of myopic foresight in the system.py file.

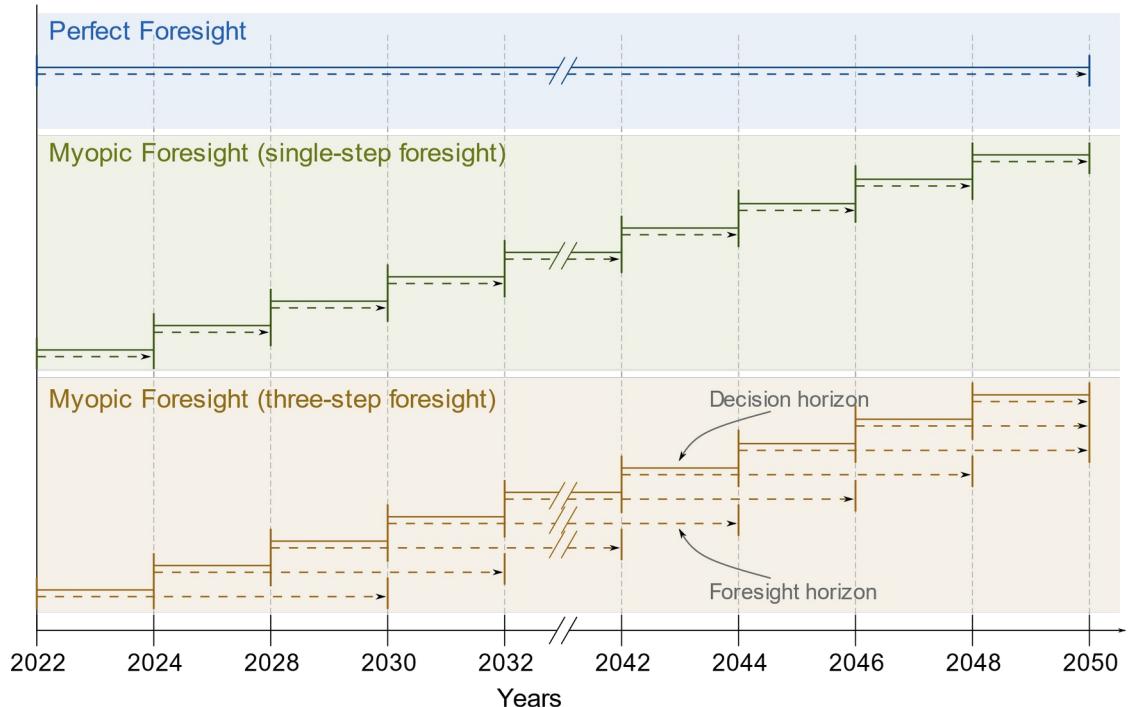


Figure 52: Whereas the decision and the foresight horizon are equal for perfect foresight, they differ for myopic foresight.

By activating myopic foresight, the ten years are all optimized separately and since the number of years in the horizon is set to one, the optimizer only regards the parameter data of the underlying year (single-step foresight) (Figure 52).

Results

The yearly newly added transport (Figure 53), storage (Figure 54) and conversion (Figure 55) technology capacities do differ quite heavily between the myopic foresight approach and the perfect foresight method. Whereas the myopic case installs significantly more transport capacity, the perfect case installs more storage power capacity to compensate. Additionally, the myopic foresight optimization doesn't install any heat pump capacities in year 0, but the perfect foresight optimization does.

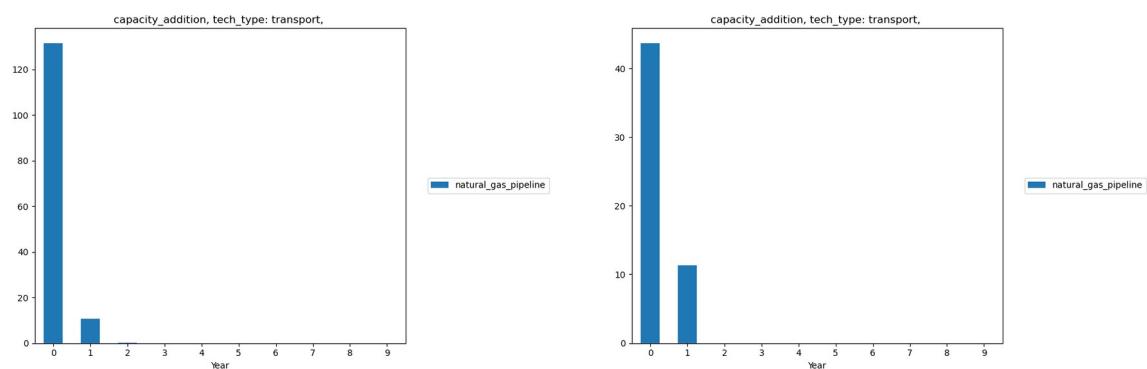


Figure 53: Yearly installed transport capacity for myopic foresight (left) and for perfect foresight (right).

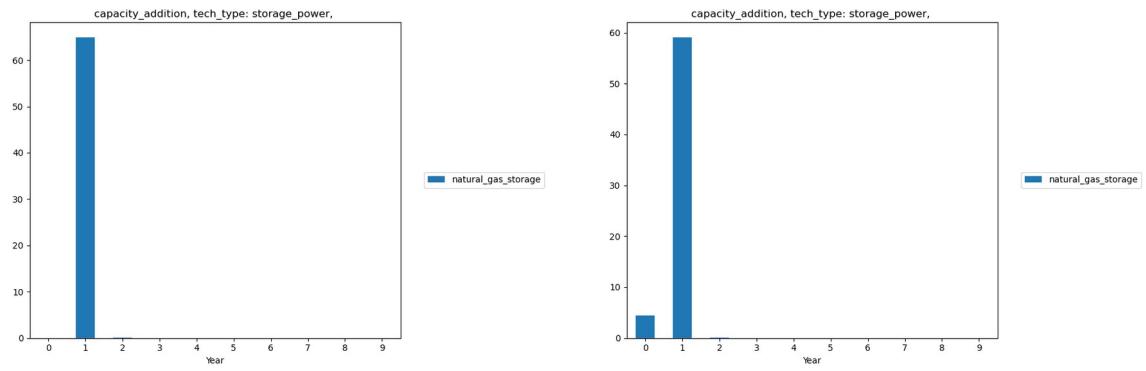


Figure 54: Yearly installed storage power capacity for myopic foresight (left) and for perfect foresight (right).

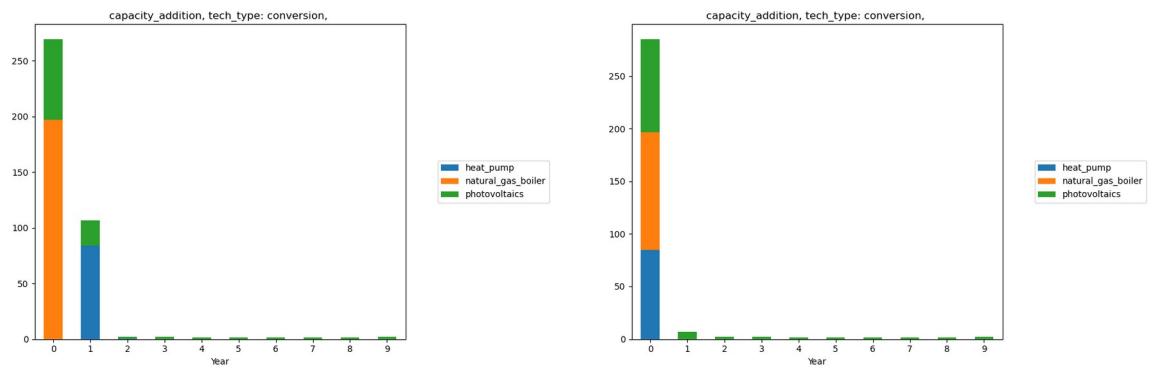


Figure 55: Yearly installed conversion capacity for myopic foresight (left) and for perfect foresight (right).

13_brown_field

This dataset shows how technology capacities which already exist before the optimization horizon starts (brown field) can be specified.

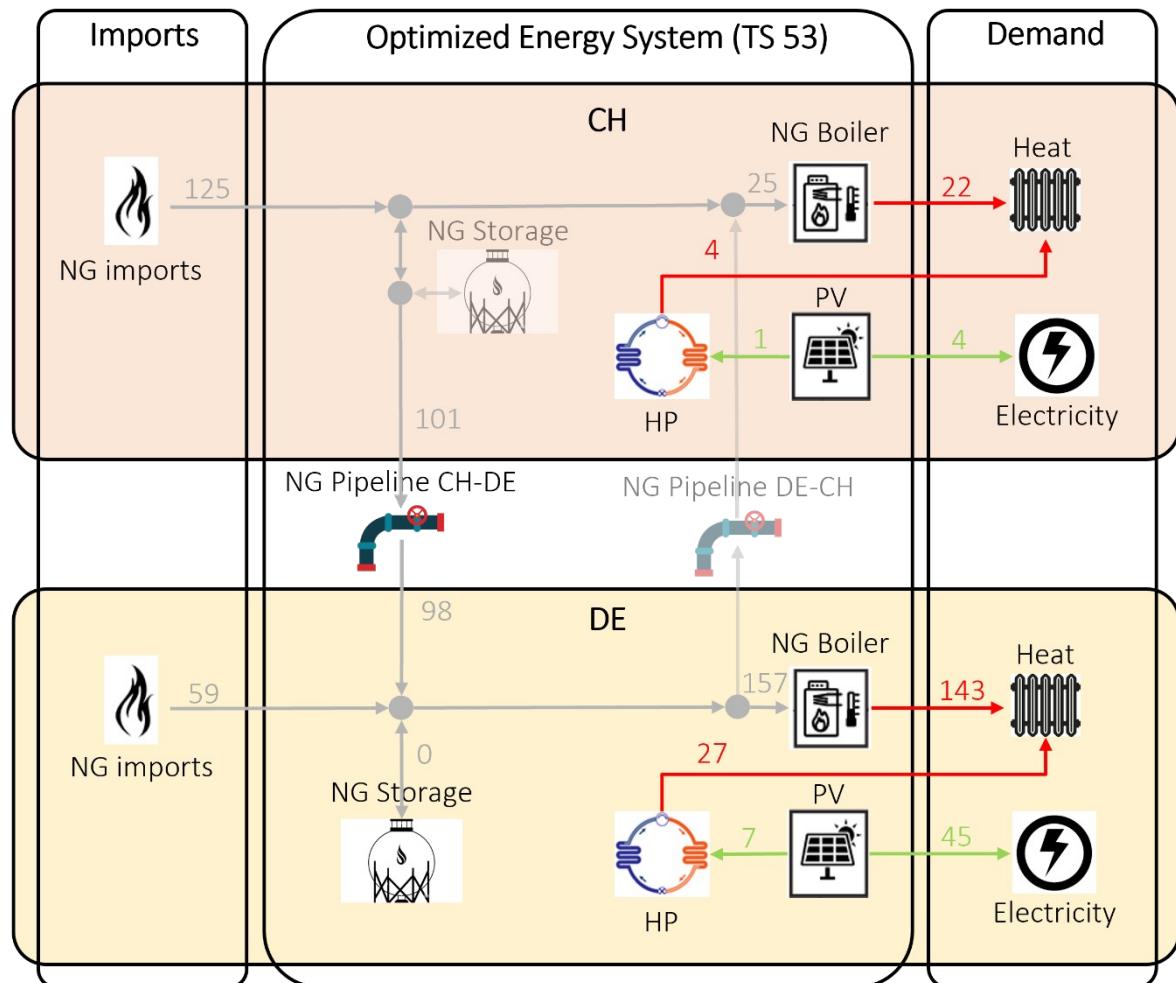


Figure 56: Energy system modelled with the brown field approach (existing capacities) at time step 53.

By defining customized values for the *capacity_existing* parameter of a technology, the capacities of e.g., existing photovoltaics can be considered (Figure 57).

	capacity_existing.csv
1	node,year_construction,capacity_existing
2	CH,2006,0.001
3	CH,2007,0.004
4	CH,2008,0.014
5	CH,2009,0.031
6	CH,2010,0.037
7	CH,2011,0.110
8	CH,2012,0.120
9	CH,2013,0.099
10	CH,2014,0.029
11	CH,2015,0.092
12	CH,2016,0.058
13	CH,2017,0.012
14	CH,2018,0.000
15	DE,1997,0.006
16	DE,1998,0.005
17	DE,1999,0.008

Figure 57: Definition of existing photovoltaic capacities by creating an additional file in the photovoltaics folder.

Results

By comparing the yearly installed conversion technology additions using existing photovoltaic capacities (brown field) with a dataset without existing photovoltaic capacities (green field) (Figure 58), it can be seen that the additions are smaller for brownfield although the overall total capacity is the same for both cases (Figure 59).

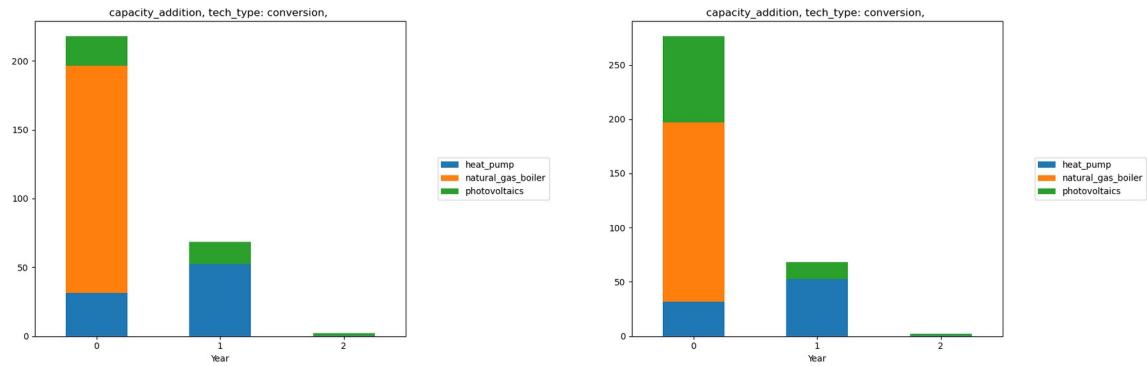


Figure 58: Yearly conversion technology capacity installation for brown field (left) and for green field (right).

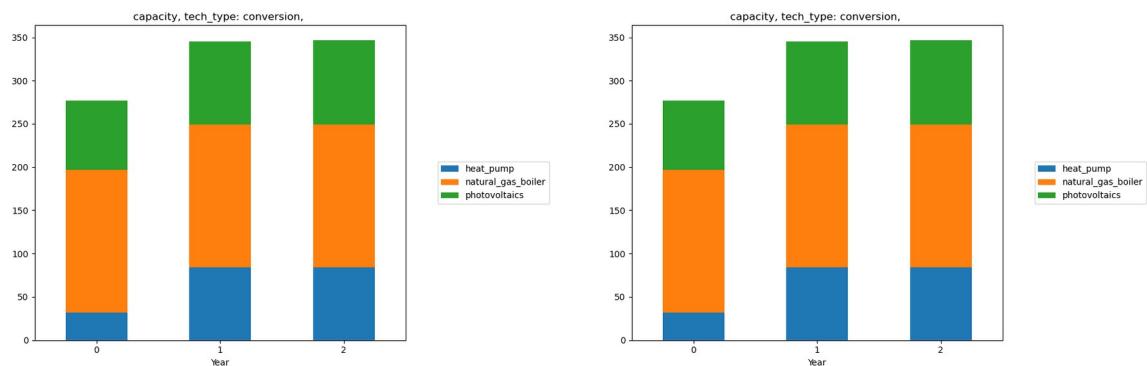


Figure 59: Total conversion technology capacity for brown field (left) and for green field (right).

14_multi_scenario

This dataset introduces the functionality of scenarios, a method to modify individual parameter values during a single framework execution.

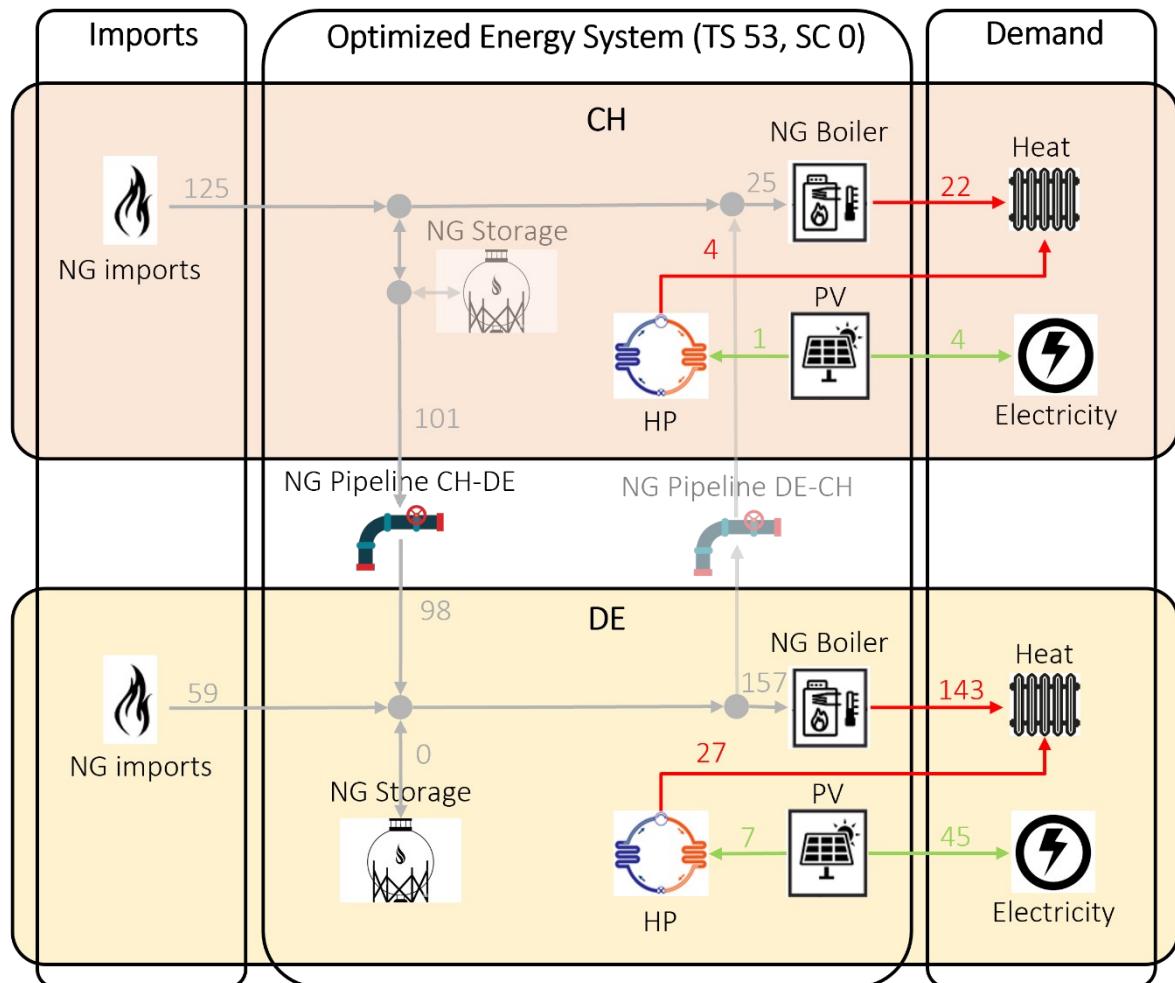


Figure 60: Multi-scenario energy system at time step 53 and the basic scenario, scenario 0.

To optimize several datasets with minor differences at once, the multi-scenario functionality can facilitate the workflow. In this example, the same dataset as in the previous example is optimized three times for different carbon emission prices. Instead of defining and optimizing three individual datasets, the scenario approach is used ([scenario discussion](#)). To activate the scenario method the “conduct_scenario_analysis” value must be set to true in the system file (Figure 61).

```
13 | # scenario analysis:  
14 |     system['conduct_scenario_analysis'] = True
```

Figure 61: Activation of the scenario analysis in the system.py file.

Additionally, the `scenarios.py` file must be provided in the most outer folder of the dataset (Figure 62). Inside this file, the different scenarios must be specified by stating which parameter should be changed with respect to the base configuration.

```

1  """=====
2  Title:      ZEN-GARDEN
3  Created:    October-2021
4  Authors:    Alissa Ganter (aganter@ethz.ch)
5  Organization: Laboratory of Risk and Reliability Engineering, ETH Zurich
6
7  Description: Scenario settings settings.
8 =====
9  scenarios = dict()
10 scenarios["1"] = {"EnergySystem": {"price_carbon_emissions": {"file": "price_carbon_emissions_1"}},}
11 scenarios["2"] = {"EnergySystem": {"price_carbon_emissions": {"file": "price_carbon_emissions_2"}},}

```

Figure 62: Definition of the scenarios by creating the scenarios.py file

In this example, three different carbon emission price scenarios are defined (Figure 63). The first file acts as the base case and the files with the “1” and “2” appendices specify the carbon prices for scenarios one and two, respectively. To minimize the run time needed to solve the optimization problem only three years are optimized again.

year	price	year	price	year	price
1	2023,0	1	2023,0	2	2023,100
2	2024,0	2	2024,100	3	2024,200
3	2025,100	3	2025,200	4	2025,300

Figure 63: The three scenarios differ in their carbon emission prices.

Results

The first row of plots shows the carbon emissions caused by the usage of natural gas (Figure 64). It can be clearly seen that the amount of produced carbon decreases for higher carbon emission prices (the y-axis of scenario 2 is scaled differently). In the second row of plots, the built conversion technology capacities can be seen (Figure 65). Again, the increase in carbon prices introduces a shift from natural gas boilers to heat pumps since they become more cost-effective at higher carbon emission prices.

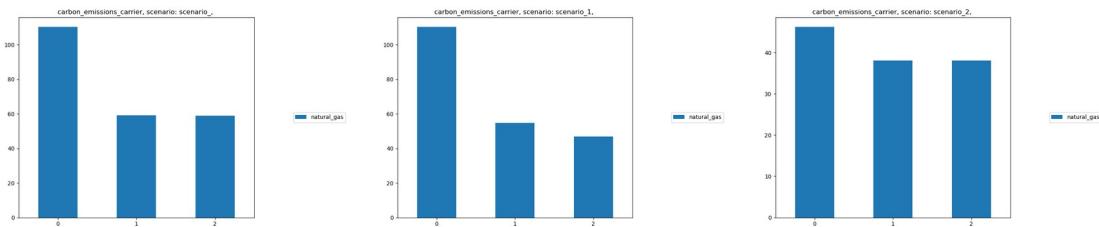


Figure 64: Emitted carbon of energy system for the three different scenarios.

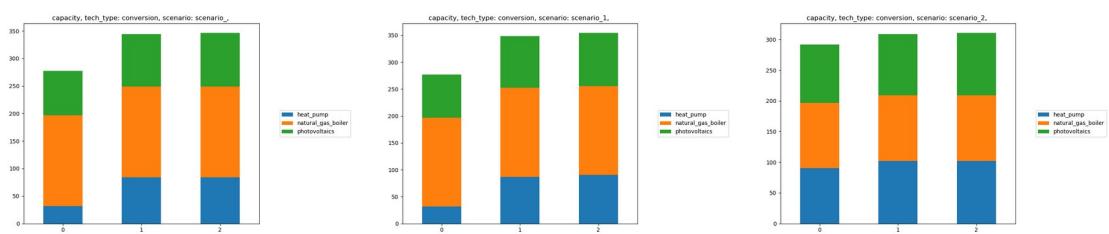


Figure 65: Installed conversion technology capacities for the three different scenarios.

15_multiple_output_carriers_conversion_tech

This dataset shows how conversion technology producing multiple output carriers can be defined, e.g., combined heat and power (CHP) plants (Figure 66).

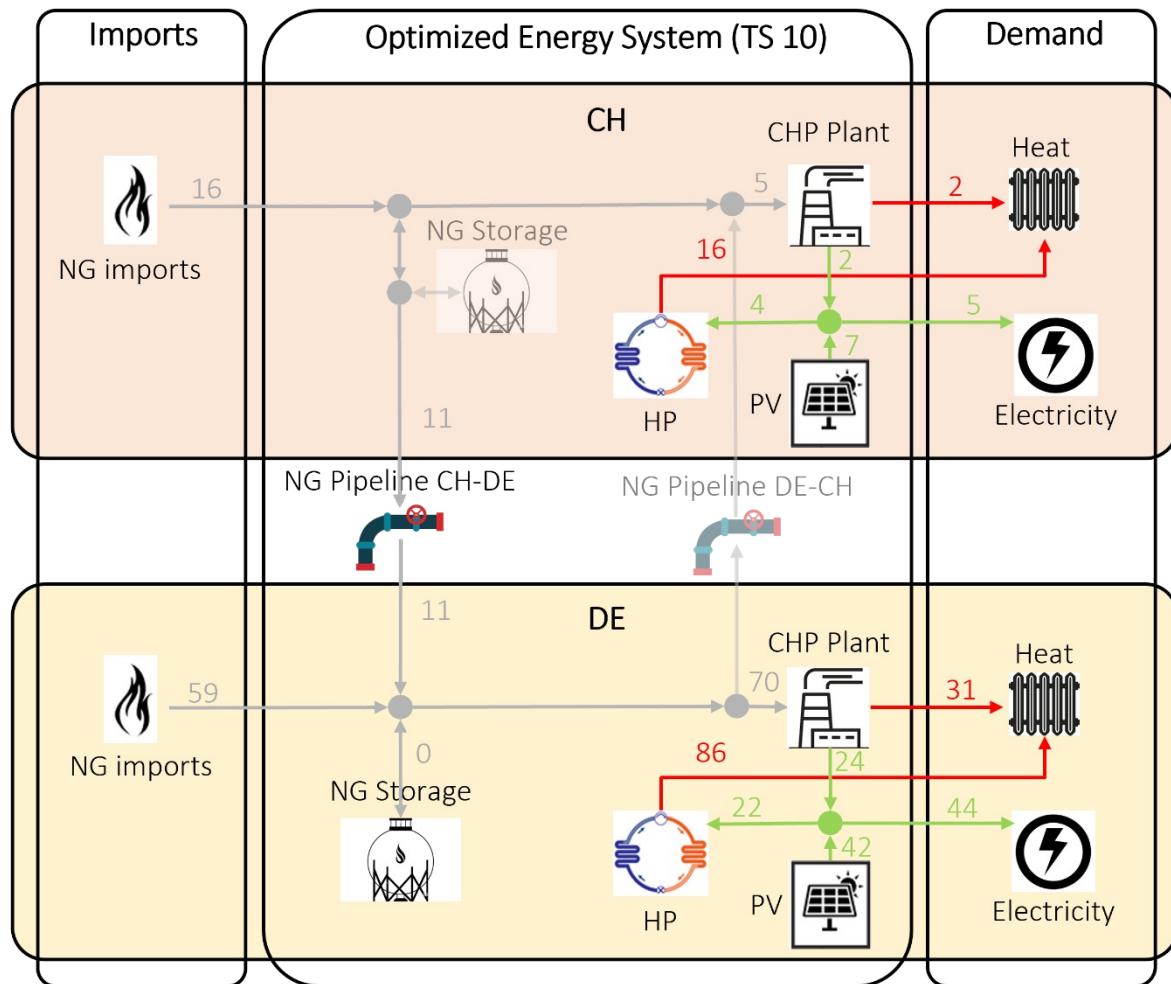


Figure 66: Energy system including CHP plants.

ZEN garden provides the functionality to model conversion technologies with multiple output/input carriers. In this example, a CHP plant (combined heat and power) is introduced, which converts natural gas to electricity and heat. For sake of simplicity, the conversion technology natural gas boiler is no longer included. To define the two output carriers of the CHP technology they need to be defined as output carriers in the attributes file (Figure 67). Additionally, the *conversion_factor.csv* file must be added to the CHP folder which is needed to specify the conversion ratios of the reference carrier to the other carriers. Assuming a CHP plant converts about 35% of the burnt fuel to electricity and 44% to heat, the conversion factors needed for the file can be computed. Since they are defined with respect to the reference carrier (electricity), the conversion factor of natural gas to electricity is 2.875 (how much gas is needed to produce one unit of electricity ($1/0.35$)) and the conversion factor of heat to electricity is 1.257 (how much heat is produced per unit of electricity ($0.44/0.35$)).

The screenshot shows a file explorer window with two main panes. The left pane displays a CSV file named 'attributes.csv' containing various parameters for a CHP plant. The right pane shows a folder structure under '15_multiple_output_carriers_conversion_tech'. Inside this folder are 'set_carriers', 'set_conversion_technologies', and a newly created 'CHP_plant' folder. Within 'CHP_plant', there are 'attributes.csv' and 'conversion_factor.csv' files. A red box highlights the 'conversion_factor.csv' file.

```

1 index,value,unit
2 capacity_addition_min_default,0,GW
3 capacity_addition_max_default,inf,GW
4 capacity_existing_default,0,GW
5 capacity_limit_default,inf,GW
6 min_load_default,0,
7 max_load_default,1,
8 lifetime_default,24,
9 opex_specific_variable_default,0,kiloEuro/GWh
10 reference_carrier,electricity,GW
11 carbon_intensity_default,0,kilotons/GWh
12 construction_time_default,0,
13 capacity_investment_existing_default,0,GW
14 opex_specific_fixed_default,87.6,Euro/kW
15 max_diffusion_rate_default,inf,
16 input_carrier,natural_gas,GW
17 output_carrier,heat electricity,GW
18 conversion_factor_default,1.1,
19 capex_specific_default,876,Euro/kW
20 capacity_addition_unbounded_default,0,GW

```

Figure 67: Definition of the multiple output carrier conversion technology CHP plant by creation of the needed folder and files.

Results

The plot of the conversion technologies' output flows shows that the CHP plants produce electricity and heat (Figure 68).

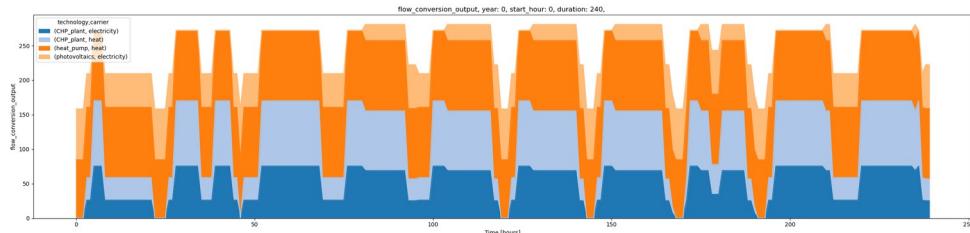


Figure 68: Output flow rates of the conversion technologies showing the different output carriers of the CHP plants heat and electricity.

16_multiple_input_carriers_conversion_tech

This dataset shows how a conversion technology with multiple input carriers can be specified.

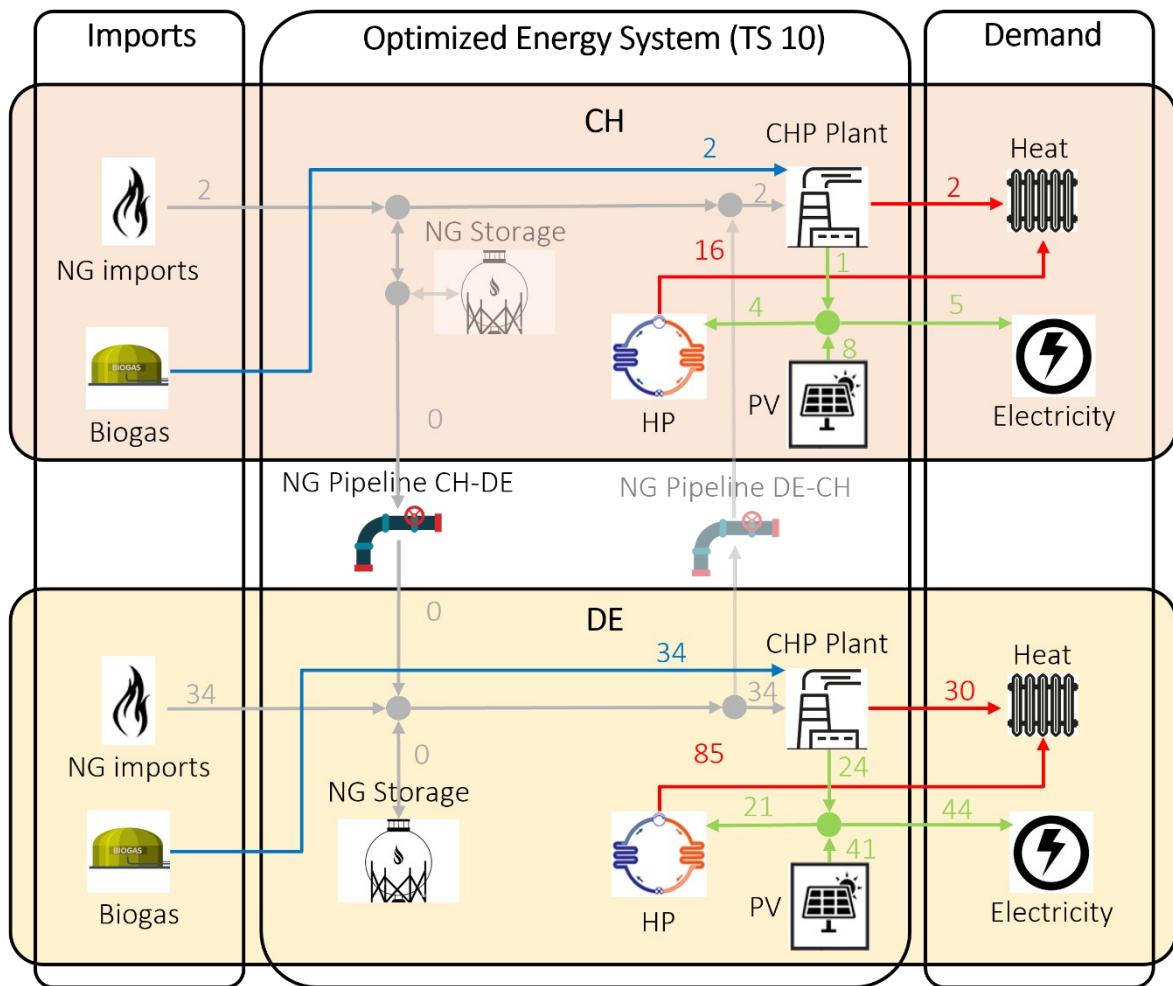


Figure 69: Energy system with multiple input carrier CHP plant.

In the same way as multiple output carriers can be defined, a conversion technology with multiple input carriers can be defined. For this purpose, it is assumed that the CHP plant burns a mixture of natural gas and biogas. Again, the biogas consumed by the CHP plants needs to be related to the reference carrier electricity. Assuming a mixture of 50% natural gas and 50% biogas, the conversion factor defined in the file is about 1.427 GWh of biogas per GWh of produced electricity (Figure 70).

attributes.csv	
1	index,value,unit
2	capacity_addition_min_default,0,GW
3	capacity_addition_max_default,inf,GW
4	capacity_existing_default,0,GW
5	capacity_limit_default,inf,GW
6	min_load_default,0,
7	max_load_default,1,
8	lifetime_default,24,
9	opex_specific_variable_default,0,kiloEuro/GWh
10	reference_carrier,electricity,GW
11	carbon_intensity_default,0,kilotons/GWh
12	construction_time_default,0,
13	capacity_investment_existing_default,0,GW
14	opex_specific_fixed_default,87.6,Euro/kW
15	max_diffusion_rate_default,inf,
16	input_carrier,natural_gas bio_gas,GW
17	output_carrier,heat electricity,GW
18	conversion_factor_default,1.1,
19	capex_specific_default,876,Euro/kW
20	capacity_addition_unbounded_default,0,GW

conversion_factor.csv	
1	year,natural_gas,heat,bio_gas
2	2023,1.427,1.257,1.427
3	unit,GWh/GWh,GWh/GWh,GWh/GWh

Figure 70: Definition of second input carrier bio gas by creating the needed files

Results

The plot of the conversion technologies' input flow rates shows how natural gas and biogas are consumed (Figure 71).

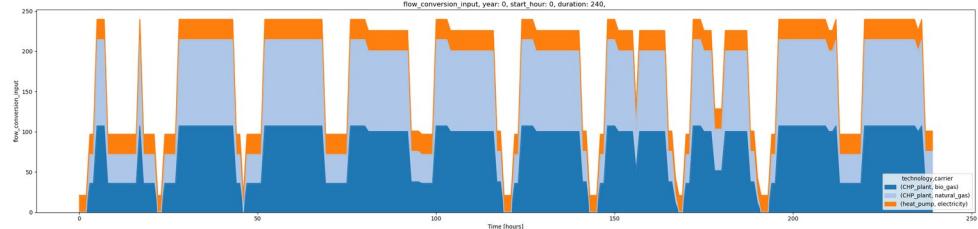


Figure 71: Input flow rates of the conversion technologies showing the multiple input carriers of the CHP plant burning natural gas and biogas.

17_yearly_parameter_missing_values

For hourly time-dependent parameters, data values at missing time indices are set to the default value. In contrast, for year-dependent parameters, data values at missing year indices are interpolated by using the data specified in the corresponding input data file (Figure 72). Therefore, defining customized values for, e.g., the carbon emissions limit for the years 2023 and 2025 automatically specifies non-default values for the years between those years. The right screenshot shows the carbon emissions limit for the years 2023, 2024 and 2025 (units did change during optimization from gigatons to megatons) which were interpolated using the data from the left screenshot. This feature can be handy to define yearly parameters which follow a linear evolution.

carbon_emissions_limit.csv		0
1	year,carbon_emissions_limit	0
2	2023,0.05	50.00000
3	2025,0.0	25.00000

Figure 72: Since the carbon emissions limit is not specified for the year 2024, it is interpolated using the specified data.

Results

Since the carbon emissions limit is reduced over the three years, the total carbon emissions emitted by the energy system decrease (Figure 73). Note that not the entire carbon emissions limit is used in the first time step (total carbon emissions = 40 megatons).

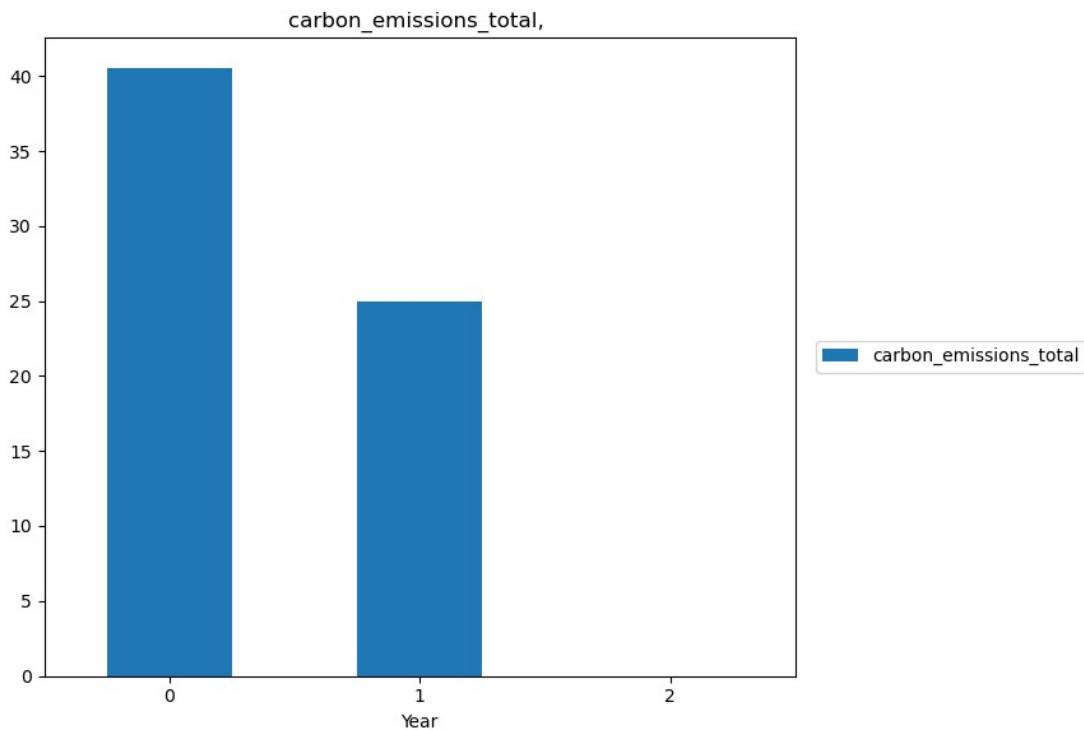


Figure 73: Decrease in total carbon emissions emitted by energy system due to carbon emissions limit.

Similarly, the effect of the introduced carbon emissions limit can be seen in the total yearly output flows of the conversion technologies (Figure 74). The heat and electricity produced by CHP plants must be shifted to carbon-neutral technologies to fulfil the carbon emissions limit.

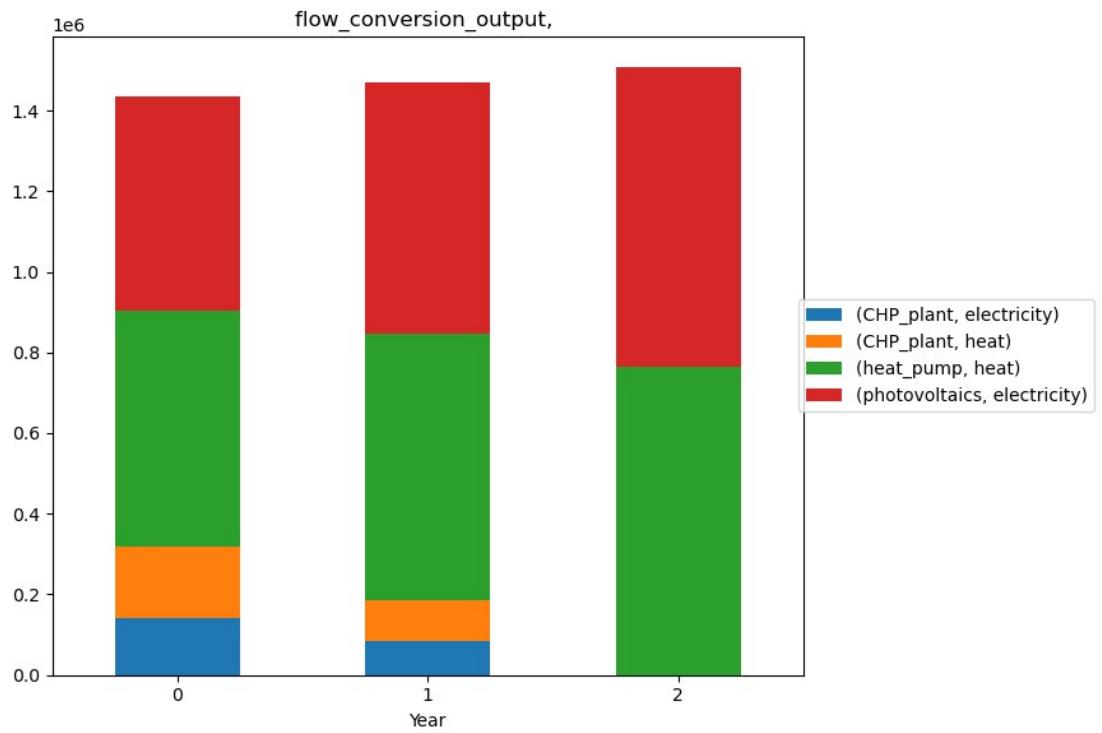


Figure 74: Total yearly output flows of conversion technologies shift from carbon-emitting to carbon-neutral technologies over years.

18_interpolation_off

To switch off the interpolation feature described in the previous example (needed if the years without specified input data should use the default value from the attributes file), the `parameters_interpolation_off.csv` file must be used. By stating the parameter names whose values should not be interpolated at missing years the feature can be turned off (Figure 75).

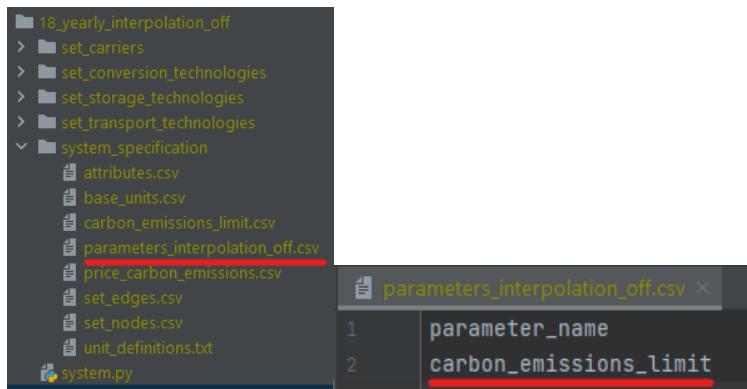


Figure 75: Definition of yearly parameters which should not be interpolated.

Consequently, the carbon emissions limit at year 1 (2024) is set to the default parameter value which is defined as infinity (Figure 76).

	0
0	50.00000
1	inf
2	0.00000

Figure 76: Non-interpolated carbon emissions limit values with default value for year 2024.

Results

By comparing the total carbon emissions of this dataset (where the interpolation feature is switched off and thus the limit for year 2024 is set to infinity) with the carbon emissions of the previous dataset (interpolated case), a difference in carbon emissions in year 1 (2024) can be seen (Figure 77).

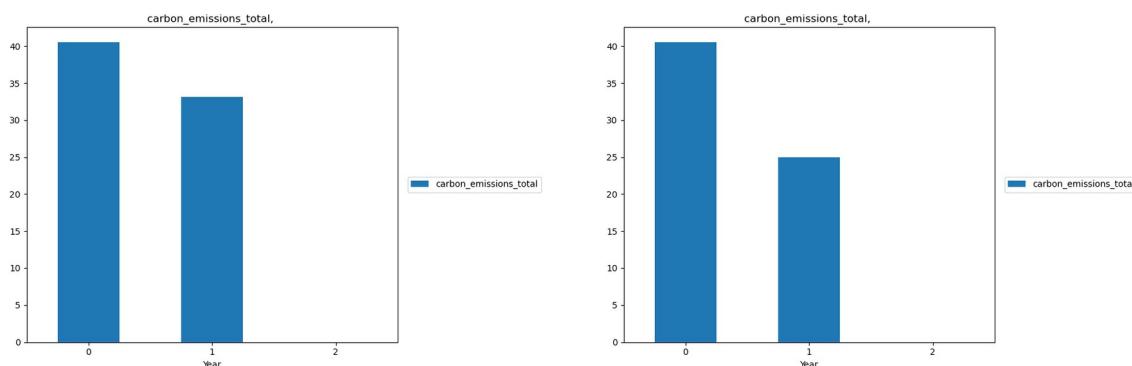


Figure 77: Total carbon emissions without interpolation (left) in year 1 differ from those with interpolation (right)

