

Code Manual for the $N - 1$ Safety Addon to the *Calliope* framework

Christoph Thormeyer

Abstract

The code is an extension to the *Calliope* framework and is realized in python 3.5.

The code extension is structured in subfunctions that execute stepwise analysis, and a main function that calls the subfunctions and manages the options.

Some of the used packages need to be installed beforehand.

Table 1: Python Libraries

Standard Libraries	3 rd Party Libraries
os	yaml
shutil	calliope (including utils)
time (optional)	numpy
	matplotlib (including pyplot)
	pandas

1 Mainfunction and Options

The mainfunction `complete_analysis` has several input options and returns a `pandas Series` with the capacities of the $N - 1$ safe system (except for `mode='check'`).

`path_to_run_directory` is mandatory and requests a string with the path to the directory that contains the original `run.yaml`.

The code will create several other files in that directory and its subdirectories. Mainly the copy model for outage modeling and the $N - 1$ safe model.

`mode=` requests a string and has two options to check a system if it already is $N - 1$ safe (`'check'`) or to design a $N - 1$ safe system (`'design'`).

'check' runs the given model with an outage for each line and checks if the system is $N - 1$ safe and prints a confirmation or the number of lines that could not be compensated.

'design' runs the given model with an outage for each line and designs a model that is able to compensate each individual outage. It enables plotting and provides csv tables with all costs and capacities.

The default setting is `'design'`

`new_run=` requests a boolean and allows to either rerun the original models or use the hdf-solutions of earlier runs (e.g. to run the same setting with different options for

the copy models to save time).

True runs the model again, which costs more time, but accounts for all recent changes.

False reads the hdf-solution of the model that Calliope provides, which costs less time, but doesn't include any changes past the last `new_run`.

The default setting is **True**

`new_copies=` requests a string (or a boolean) and allows to run the models parallel (e.g. on a cluster) (**'parallel'**) or to run all models one after the other (**'sequence'**). If none of the two options are used, the program will take the existing solutions, assuming they already exist (e.g. to run the same setting with different plot options without losing too much time) (boolean **False**).

'parallel' uses the parallel feature of Calliope to run all copy models either on a cluster or on a PC.

'sequence' runs each model separately one after the other. This options should only be used if running the models in parallel causes problems.

False (or any input other than 'parallel' or 'sequence') reads the save location from the run file and searches the hdf solution files there.

The default setting is **'parallel'**

`new_plots=` requests a boolean and allows to save new plots for this run (**True**) or to not create plots (e.g. to save time while trying different settings) (**False**).

True will create and save plots.

False will not create plots.

The default setting is **True**

`get_csv=` requests a boolean and allows to save the values of each model in a respective costs and capacities table (**True**) or not to save them (**False**).

True will create and save the tables.

False will not create the tables.

The default setting is **True**

2 Other Functions & Options

To know the computation time, `get_time` gives you the time from the input time (e.g. `time.time()`) and the moment when `get_time` is called. This can be helpful to optimize your workflow, since you know how much time your models will probably need to finish.

The plot subfunctions have options for the presentation (however they need to be enabled by changing the input in the code):

`plot_bars` has an option `fixed_scale=` which requests a boolean and plots the bar plots either each with axes normalized by their respective largest value in the plot (**False**) or with axes that are normalized by the overall largest value for all plots (**True**).

False allows to read all capacity and cost values directly from the axes.

True shows the capacities in a larger scale, which allows to quickly identify the relevant lines.

The default setting is **False**.

`plot_lines` has an option `fixed_scale=` which is identical to the `fixed_scale=` option of the subfunction `plot_bars`. It also has an option `bars=` which requests a boolean and allows to plot the lines plots without barplots (`False`) or with an additional subplot with respective bars (`True`).

The default setting is `False` for the subfunction, but is `True` for the call in the main function.