Energy System Modelling
Karlsruhe Institute of Technology
Institute for Automation and Applied Informatics
Summer Term 2019

Tutorial VI (Bonus): Clustering Large Power Systems

This is a bonus tutorial and will not be covered by any exercise session.

Problem VI.1 (programming) – clustering the European grid

In the preceding tutorials we have seen that the integration of renewable energy is cheaper when they are integrated over a large geographical area and that its variability requires a fine temporal resolution. In consequence, however, the computational power needed for the optimization of the whole energy system is immense and could be prohibitively high. In this tutorial, we will first compare 2 basic clustering solutions to reduce the grid model size and thereby alleviate the computational burden. Then we will observe the impact of clustering on the lowest installed capacity required to feed the demand.

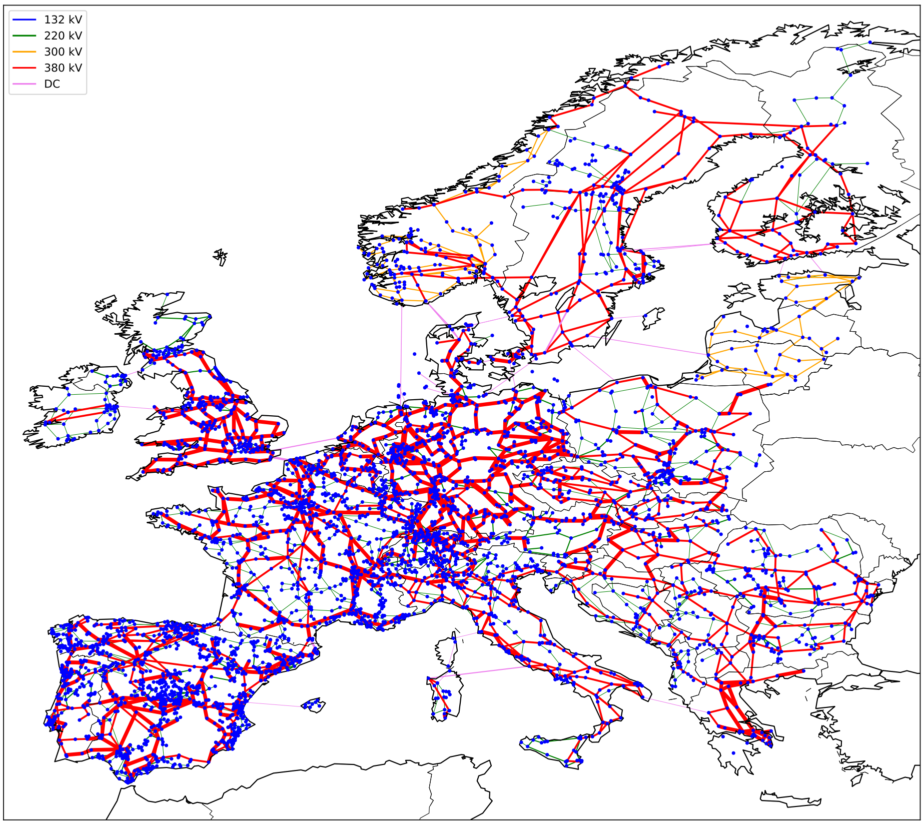

Figure 1: The European power grid.

(a) The file `network_data/base.nc` contains a detailed representation of the European power grid, including the various high voltage AC lines and the DC links.

  (i) Load the file into the provided Jupyter notebook and plot it. The *colours* of the lines represent the voltage type and level, whereas the *width* of the lines are proportional to their transmission capacity.

  (ii) Calculate the order and the size of this graph.

  (iii) To cluster this network, we will use a simplified version of this grid (`network_data/elec_s.nc`) which bases all AC lines to the same voltage level and groups buses which are very close to each other.

  (iv) Calculate the order and the size of the simplified graph.

(b) We will now turn to clustering the European power network using *Sklearn*. Sklearn is a library which allows the clustering of data. For the sake of exposition, we will cluster the network simply by grouping buses according to their geographical locations.

  (i) Perform various clustering configurations of the network by altering the clustering algorithms and the number of clusters k. Find the available clustering algorithms at `http://scikit-learn.org/stable/modules/clustering.html`.

  (ii) Use the `plot_clusters` function of the package `tutorial.py` to observe the outcome of the clustering.

  (iii) Are the buses adequately clustered? Which algorithm is giving the best outcome? What are the limits of using geographically based data clustering on the electrical grid? Comment on your findings!

(c) We will now look at the interaction between the clustering configuration and minimum installed capacity in the network. The network file `network_data/elec_s.nc` also provides solar and wind energy hourly profile per bus for the year 2013 as well as an hourly load profile per bus. The run-of-river hydropower is also fixed.

We will make use of the functions `tutorial.find_kmeans_busmap` and `pypsa.networkclustering.get_clustering_from_busmap` to cluster the European power grid according to the K-Means algorithm, with weights corresponding to the maximum load at the respective bus. These functions aggregate the bus data (i.e. load, renewable energy potential, etc.) to the cluster nodes and group lines from the initial network to form the edges of the clustered graph.

The clustered network is missing some installed capacities for renewables at each bus to be able to run a linear optimal power flow (LOPF). In order to set a capacity of solar, onshore wind andor offshore wind to each bus in a simple way, we use the function `tutorial.P_nom_re(network)` to attribute an installed capacity proportional to the load at each bus. The proportionality coefficients for solar `w_s`, onshore wind `w_on` and offshore wind `w_of` could be changed independently. Keep in might, this is just for illustrative purposes probably not very accurate in reality.

  (i) For a graph with a small number of clusters (e.g. between 10 and 25), determine minimum proportionality coefficients `w_s`, `w_on` and `w_of` to have a feasible solution for the LOPF at a specific day and hour.

  (ii) If you try the same coefficients with a higher number of clusters (e.g. between 100 and 200), is the LOPF still feasible. If not, could you find an explanation?