

Data-driven Adaptive Benders Decomposition for the Stochastic Unit Commitment Problem

Baudouin Vandebussche, Stefanos Delikaraoglou, Ignacio Blanco and Gabriela Hug

This document provides a short description of the Python Files used in the paper “Data-driven Adaptive Benders Decomposition for the Stochastic Unit Commitment Problem” (submitted to IEEE TRPWS).

All the data (generators, lines, windfarms etc.) used in the case study are provided in the folder **Data** of this repository.

Python codes:

- **Cluster_creation.py** is the code used to create the initial clusters for the outer parallelization.
- **comparison.py** is the code used to compare the results of the first step and to create the set of common values.
- **kmedoids.py** is the code which implement the k-medoid method.
- **defaults.py** imports the data and defines some initial values
- **Lib_Cluster.py** is used to create clusters in case of a clustering based on forecasts in the inner parallelization.
- **Data_Load.py** imports and processes the data.

All the following files exist in two versions:

- i. **noB** for the case when the Unit Commitment problem is solved without Benders decomposition and
 - ii. **multiprocess_Parallel** for the algorithm presented in the paper
- **DA_MarketClearing_Main.py** is the main file calling all the other scripts, written for a Linux OS and with the Unit Commitment as output. This file is used in parallel for the first step of the algorithm and for the second step with fixed binary values.
 - **Lib_Constraints.py** defines the constraints of the optimization problem.
 - **Lib_Variables.py** defines the variables of the optimization problem.
 - **Lib_ObjFnct.py** defines the objective function of the optimization problem.