ClustEnergy OpTool

ClustEnergy OpTool is a useful tool for evaluating the energy flexibility potential offered by an aggregate of buildings (e.g., building clusters [1]), which is useful for resource planning in future scenarios. It is a simple Python [2] tool based on linear programming that allows simulation of different demand management strategies in different user-defined clusters of buildings. Through archetype-based approach, it is possible to define a representative cluster of buildings subjected to a demand management strategy during a simulation period and reference location. Then, from the comparison with a baseline scenario (BL), it is possible to study the flexible behavior of a cluster of buildings subjected to a demand response (DR) strategy. Specifically, through centralized optimal control, a peak shaving strategy (for congestion management) or a load shifting strategy (for better integration of PV generation) can be implemented [3]. Although the currently available archetypes refer to the Italian case study, the tool given its simplicity is easily modified and adaptable to the needs of the individual use.

###

Python Libraries

To run simulations, check that the following libraries are installed:

- pandas
- numpy
- scipy
- pvlib
- matplotlib
- os
- sys
- psychrolib
- math
- datetime.

###

How to start

To set up a simulation, insert the following information into the "`ClustEnergy_main.py`" python file:

1. **Reference Location (`Locality`)**

Choice of reference location for outdoor environment simulation. The reference locations currently available in the tool are:

- “Ancona" (IT)
- "Messina" (IT)
- "Milan" (IT)
- "Rome" (IT)
- "Denver" (USA) in reference to the “BESTEST” standard ANSI/ASHRAE 140 [4].

Alternatively, additional locations can be considered through the integration of new climate files in epw format within the tool. For this purpose, it is possible to download the climate files from appropriate sites (such as, Climate.OneBuilding.Org [5]) and add the location within "`CheckMeteoFile`", "`MeteoFile`", and "`PV_load`" functions in MeteoFile.py. For the purposes of the tool, follow the steps below to add new locations from those available:

- insert the climate file in epw format in the "epwFile" folder,
- add the climate file in the check function "`CheckMeteoFile`" in "MeteoFile.py" with the following code line in the if statement

```
and loc != 'NEW_LOCATION'
```

- to extrapolate data useful for simulation of the outdoor environment and calculation of solar gains, in the "`MeteoFile`" function in "MeteoFile.py" insert the following lines of code

```
if loc == 'NEW_LOCATION ':
    psychrolib.SetUnitSystem(psychrolib.SI)
    currentPath     = os.path.dirname("__file__")
    weatherPath     = os.path.join(currentPath, 'epwFile')
    weatherFileName = "NEW_LOCATION.epw"
    weatherFile     = os.path.join(weatherPath, weatherFileName)
    epw             = Weather(weatherFile)
    latitude_deg    = LAT(NEW_LOCATION)
    longitude_deg   = LONG(NEW_LOCATION)
    UTC             = UTC(NEW_LOCATION)
```

- to extrapolate data for the creation of photovoltaic generation profiles via pvlib.py [6], in the "`PV_load`" function in "MeteoFile.py" insert the following lines of code

```
if loc == 'NEW_LOCATION ':
    currentPath     = os.path.dirname("__file__")
    weatherPath     = os.path.join(currentPath, 'epwFile')
    weatherFileName = "NEW_LOCATION.epw"
    weatherFile     = os.path.join(weatherPath, weatherFileName)
```

2. **Type of thermal load (`Thermal_load`)**

   Choose whether to consider space "cooling" or "heating". By choosing "cooling," the thermal loads given as input to the space state modeling formulation will be with negative sign. In addition, the performance characteristics of a heat pump and the thermostat temperature set-point for cooling will be considered. Conversely, by choosing "heating," the heat loads given as input will have a positive sign. Finally, the performance characteristics of a heat pump and the thermostat temperature set-points for heating will be considered.

3. **Simulation period**

   To define the period over which the simulations are to be carried out, define the following parameters:

- **Starting month simulation** (`month`): simulation start month from 1 (January) to 12 (December)
- **Starting day simulation** (`day`): simulation start day
- **Number of simulation days** (`n_days`): number of simulation days (max days of simulation = 365 - simulation start day of the year).
- **Time step** (`mins`) **:** time step in minutes.

4. **Number of buildings composing the cluster**

Number of buildings composing the representative cluster. The representative cluster comprises a varying number of buildings. To represent buildings with different age classes, an archetypal approach was used. This involved considering different building typologies and defining their characteristics (such as surface area, volume, composition of the building envelope, etc.) based on data from the Tabula Project [7] referring to the Italian building stock. In the context of Single-Family Houses, Table 1 outlines the combined archetypes with their corresponding age classes, associated thermal insulation levels, and available emission systems (marked with an "x"). While Table 2 shows the U-values. To represent the energy demand of individual buildings, lumped-parameter modeling based on the thermoelectric analogy (RC networks) was adopted. For additional details on building modeling, please consult [REFs]. Information regarding the construction features of the buildings can be found in the "Archetypes" excel file within the path. While information about tool validation via ANSI/ASHRAE Standard 140 [4] (i.e., BESTEST) is provided in the Annex "ClustEnergy_BESTEST_validation".

Table 1: Archetypes combined with corresponding age classes, associated thermal insulation levels and emission systems available in ClustEnergy OpTool.

Age class	SFH archetype	Insulation level	Air cooling system	Air heating system	Underfloor heating system
before-1970	1946-1960	none	x	x	-
1971-1990	1976-1990	low	x	x	-
1991-2005	1991-2005	medium	x	x	-
2006-today	2006-today	high	x	x	x

Table 2: U-values (W m^{-2} K^{-1}) for archetypes of Single-Family Houses (SFHs) according to Tabula Project [7].

SFH	External walls	Floor	Roof	Windows
1946-1960	1.48	2.00	2.20	4.90
1976-1990	0.76	0.76	1.14	2.80
1991-2005	0.59	0.63	0.57	2.40
2006-today	0.34	0.33	0.28	2.20

Nomenclature: archetypes are referred as SFH06, SFH05, SFH90, and SFH60 to recall SFHs with class years of 2006-present, 1991-2005, 1976-1990, and 1946-1960, respectively. Furthermore, the words "air" or "floor" are added at the end to indicate the air or underfloor emission system.

Note: the code is currently written to consider a maximum of 20 buildings, which is useful for performing simulation on a representative cluster without excessive run times. Considering more than 20 buildings requires the addition of additional occupancy profiles, relative internal gains, and temperature set-points beyond those predefined in 'User_pattern.py'.

5. Heat pump features

The capacity and coefficient of performance of heat pumps are calculated here as the outdoor temperature varies. With "`pre_defined_HP`" in the following "`HP_data_type`" it is possible to choose between some commercial heat pumps available in the library. While, with "`user_defined_HP`" it is possibile to consider normalized performance characteristic curves provided by the user via external Excel file. With reference to commercial heat pumps, the model of the heat pumps available in the library is based on catalog data for capacity (total capacity in heating/cooling mode) and coefficient of performance (COP) provided by the manufacturer [8].

To run a simulation, set the supply temperatures, rated capacities and COP of heat pumps for space cooling/heating. If "`pre_defined_HP`" data, select between the following heat pumps available in the library in reference to the data provided by the manufacturer [8] :

- **HEATING** (with reference to supply temperature of 35°C and outdoor air temperature of 7°C)
  - `rated_capacity`: 8.64 and `rated_COP`: 4.19
  - `rated_capacity`: 14.00 and `rated_COP`: 4.08
  - `rated_capacity`: 18.00 and `rated_COP`: 3.90

- **COOLING** (with reference to supply temperature of 18°C and outdoor air temperature of 35°C)
  - `rated_capacity`: 10.00 and `rated_COP`: 2.70
  - `rated_capacity`: 11.50 and `rated_COP`: 2.94
  - `rated_capacity`: 14.22 and `rated_COP`: 3.43

To consider normalized performance characteristic curves provided by the user, insert the data in the Excel files located in the "userProfiles" path. In particular, "HP_performance_cooling" file for space cooling and "HP_performance_heating" for space heating. See the calculation example in the Excel sheet "HPper_data_calculation" within the Documentation.

Consequently, the cooling/heating capacities useful for defining the boundaries for calculating the thermal demand and Coefficient Of Performance (COP) useful for calculating the electricity demand of individual buildings will be given as the external temperature varies.

6. **Thermostat settings**

Determine the thermostat settings by choosing whether to consider a `"pre_defined"` temperature profile created according to a normal distribution or set a fixed temperature via `"user_defined"`. Then, define the thermostat `setpoint` temperature in case of `"user_defined"` profile (e.g., 20°C for heating or 26°C for cooling). In the case of `"pre_defined"` profiles, new ones can be created by random allocation according to a normal distribution via the function "Tsetpoit_calc.py" inside the folder. Once the mean, standard deviation, and type of choice between hourly and daily are defined, profiles can be created randomly for each individual building.

7. **Thermostat tolerances (baseline scenario)**

Once the temperature profiles are defined, the thermostat tolerances need to be set. Using `th_tolerance_BL_i`, the upper/lower tolerances of the thermostat during the baseline (BL) scenario for air cooling/heating and underfloor heating systems are defined. Accordingly, the indoor air temperature during the BL scenario will be between `upper_BL = tsp_profile + th_tollerance_BL_up` and `lower_BL = tsp_profile - th_tolerance_BL_low`.

8. **Demand response strategy (`Demand_response`)**

Choose the type of Demand Response scenario for managing aggregate loads. It is possible to perform a `"peak_shaving"` strategy for the reduction of peak loads starting from the definition of the peak hour, or a load shifting according to centralized (`"pv_centralized"`) or distributed (`"pv_distributed"`) photovoltaic generation. In particular, centralized production represents the case where a solar photovoltaic generation is available for the entire cluster (sharing of energy resources). While distributed generation represents the case in which each family is provided with its own photovoltaic generation. As a result, during centralized production, excess PV production is made available to other buildings in the cluster. In contrast, during distributed production, solar PV surpluses is considered fed into the electricity grid. To understand the parameters to be set according to the chosen strategy, see the following sections:

- **Peak shaving settings:** starting from the electric demand of the baseline scenario, it is possible to calculate the time when the peak electric demand occurs. Thus, if selected, a DR scenario is applied to reduce the peak by the desired magnitude at the most critical hour (`Demand_response = "peak_shaving"`). For this purpose, a reduction factor needs to be set to bind the electrical demand of the building cluster to remain below a certain percentage for the duration of the peak-shaving event. Specifically, set the peak-load reduction factor `"f_red"` from 0 (no reduction) to 1 (100% reduction) and the duration `"duration_dr"` of the DR event (to be considered from the time step where the maximum electrical demand occurs). Furthermore, to limit rebounds effects it is possible to set a limit to the cluster electricity consumption of a certain `"f_limit"`.
- **Photovoltaic generation settings:** once the PV generation profile and the electricity demand of the building group have been defined, two different DR scenarios of shifting electricity demand as a function of PV generation can be considered. To minimize the demand for electricity drawn from the grid, PV generation settings need to be defined. By defining "`pv_size`", the PV system sizing can be `"calculated"` according to Italian regulation Dlgs. 28/2011 or `"user-defined"`. If "calculated", via `"panel_dim"` specify the rated power in W of the PV panel. While if "user-defined", specify the rated power of the whole PV system in kW for archetypes with air cooling/heating system (`"pv_rp_n_air"`) and for archetypes with floor heating system (`"pv_rp_n_floor"`). Furthermore, to limit rebounds effects it is possible to set a limit to the cluster electricity consumption of a certain `"f_limit"`.
- **Load shifting under price signal settings:** to minimize the demand for electricity drawn from the grid under the influence of a price signal, it is necessary to set the electricity rates. In particular, a bi-hourly Time Of Use rate is considered, by defining the time bands and the electricity price (Eur/kWh) in `"price_ranges"`. Furthermore, to limit rebounds effects it is possible to set a limit to the cluster electricity consumption of a certain `"f_limit"`.
- **Thermostat tolerances:** to unlock the energy flexibility of buildings, upper/lower thermostat tolerances are guaranteed during the Demand-Response scenario. Accordingly, the indoor air temperature during the BL scenario will be between "upper_DR = upper_BL + th_tolerance_DR_up" and "lower_DR = lower_BL - th_tolerance_DR_low". During the peak load reduction scenario for room cooling, increase upper_BL by `th_tolerance_DR_up`. Similarly, during the peak load reduction scenario for space heating, reduce the lower_BL of `th_tolerance_DR_low`. While, during the load shift scenario for space cooling in the presence of PV generation, reduce the lower_BL value of `th_tolerance_DR_low`. Similarly, during the load shifting scenario for space heating under PV generation, increase the upper bound (upper_BL) value of `th_tolerance_DR_up`.

References

1. Vigna I, Pernetti R, Pasut W, Lollini R (2018) New domain for promoting energy efficiency: Energy Flexible Building Cluster. Sustain Cities Soc 38:526–533. https://doi.org/10.1016/j.scs.2018.01.038
2. Python (1991) Python
3. Arteconi A, Polonara F (2018) Assessing the Demand Side Management Potential and the Energy Flexibility of Heat Pumps in Buildings. Energies (Basel) 11:1846. https://doi.org/10.3390/en11071846

4. ASHRAE (2020) ANSI/ASHRAE Standard 140-2017 - Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs

5. (2023) Climate.OneBuilding.Org - Repository of free climate data for building performance simulation. In: Climate.OneBuilding.Org . https://climate.onebuilding.org. Accessed 19 Jan 2024

6. F. Holmgren W, W. Hansen C, A. Mikofski M (2018) pvlib python: a python package for modeling solar energy systems. J Open Source Softw 3:884. https://doi.org/10.21105/joss.00884

7. Corrado V, Ballarini I, Corgnati SP (2012) National scientific report on the TABULA activities in Italy

8. Viessmann (2020) Dati integrativi pompe di calore VITOCAL 250-S per il calcolo delle prestazioni energetiche degli edifici, secondo UNI/TS 11300 parte 4