examples Documentation

Isa

Jan 22, 2020
# Contents

## 1 OMEGAlpes Examples
1.1 Basic example: PV self-consumption .................................................. 4
1.2 Electrical system operation ................................................................. 5
1.3 Storage design .................................................................................. 5
1.4 Waste heat recovery ......................................................................... 6

## 2 OMEGAlpes article study cases
2.1 Basic example: PV self-consumption .................................................. 11
2.2 Multi-actor modelling for MILP energy system optimisation: application to collective self consumption 12
2.3 Other case studies ........................................................................... 13

## 3 How to run an example

## 4 How to run a project with Jupyter Notebook
4.1 I. How to use Jupyter Notebook remotely on Binder ......................... 17
4.2 II. How to use Jupyter Notebook locally .......................................... 17
OMEGAAlpes stands for Generation of Optimization Models As Linear Programming for Energy Systems. It is an Open Source energy systems modelling tool for linear optimisation (LP, MILP).

Various examples and study cases have been developed on OMEGAAlpes. They are stored in the following Gitlab: OMEGAAlpes Examples. They are using a specified graph representation described here: OMEGAAlpes Representation.

Examples are developed to help new omegalpes users. Article study cases are developed for scientific concerns.

To run both, you will first need to install OMEGAAlpes library. To do so, please, have a look to the documentation: OMEGAAlpes Installation. Or to the README.md of OMEGAAlpes Gitlab.

Some examples are also developed on Jupyter Notebook for a better understanding. To know how to run example python codes or notebooks, see run_and_notebook_help.

Note: The examples may be updated with the last developer version which may be different from the OMEGAAlpes user (Pypi) version. Thus, you may have to run the examples with the developer version. Otherwise you have to select the example version corresponding to the current Pypi version. The version used is indicated at the beginning of the example module.

OMEGAAlpes Examples and article study cases
Please have a look to the following examples:

- Basic example: PV self-consumption
- Electrical system operation
- Storage design
- Waste heat recovery

The code is stored at the Gitlab: OMEGAlpes Examples in the folder “beginner_examples” or “examples”. Some of them have also been developed on a Jupyter notebook for a better understanding.

**Note:** To know how to run the example python codes or the notebooks, see: Help run Jupyter Notebook or Help run example

Consider this address to run notebooks with Binder:
https://gricad-gitlab.univ-grenoble-alpes.fr/omegalpes/omegalpes-notebooks

Consider this address to run the examples:
https://gricad-gitlab.univ-grenoble-alpes.fr/omegalpes/omegalpes_notebooks
1.1 Basic example: PV self-consumption

In this PV self-consumption example, a single-house with roof-integrated photovoltaic panels (PV) is studied. Specifically, the study case is about demand-side management in order to maximize the self-consumption, by shifting two household appliances consumption (clothes washing machine and clothes dryer) and using a water tank for the domestic hot water consumption.

The article presenting this example is available at: PV self-consumption article
You can access to this example via an online Notebook at: PV self-consumption online notebook
The code is available here: PV self-consumption code
The Notebook is available at: PV self-consumption notebook

This example leads to a study with:
- 6922 variables (2890 continuous and 4032 binary)
- 79172 non-zeros

This optimization problem has been generated within 1.2 seconds on an Intel bicore i5 2.4 GHz CPU.
An optimal solution was found in 43.6 seconds with the free CBC solver available in the PuLP package, and in the 2.5s with the commercial Gurobi solver.
1.2 Electrical system operation

This first module is an example of decision support for electrical system operations. The electrical system operator needs to decide whether to provide electricity from the grid_production A or B depending on their operating costs. The two grid productions are providing energy to a dwelling with a fixed electricity consumption profile.

The code is available here: Electrical system operation code
The Notebook is available at: Electrical system operation notebook

Fig. 2: Figure 2: Principle diagram of the electrical system operation example \ Author: Lou Morriet

1.3 Storage design

The storage_design module is an example of storage capacity optimization. A production unit and a storage system power a load with a fixed consumption profile. The production unit has a maximum power value and the storage system has maximum charging and discharging power values. The objective is to minimize the capacity of the storage system while meeting the load during the whole time horizon.

The code is available here: Storage design code
The Notebook is available at: Storage design notebook
1.4 Waste heat recovery

In the waste_heat_recovery module, an electro-intensive industrial process consumes electricity and rejects heat. This waste heat is recovered by a system composed of a heat pump in order to increase the heat temperature, and a thermal storage that is used to recover more energy and have a more constant use of the heat pump. This way, the waste heat is whether recovered or dissipated depending on the waste heat recovery system sizing. The heat is then injected on a district heat network to provide energy to a district heat load. A production unit of the district heat network provides the extra heat.

The code is available here: Waste heat recovery code
The Notebook is available at: Waste heat recovery notebook

Technical and decision constraints and objectives can be added to the project. This leads to the following Figure 5.

Applying, multi-stakeholder vision on the waste heat recovery project leads to the Figure 6. One central point is the governance of the storage and heat pump. Who’s financing it? which actor will operate it? This governance needs to be discuss and mutually agreed to be able to go further on the project.

**A technical optimisation over one year on a hourly time step can lead to a study with**
- 228k variables (158k continuous et 70k binaires)
- 316k constraints

It has been solved in 13h with Gurobi, which can be considered as correct considering the high number of variables and constraints.

Considering the 20MWh / 6.7MW storage this can of study can calculate that 60% of the annual needs could be covered by the LNCMI waste heat (which corresponds to 60% reduction in CO2 emissions)! This outputs should be consider regarding the constraints and objectives of the model, which are not totally detailed here, as the goal of this part is to show the possibilities of OMEGAlpes.
Fig. 4: Figure 4: principle diagram of the waste heat recovery example \ Author: Camille Pajot
Fig. 5: *Figure 6: principle diagram of the waste heat recovery example with constraints* | Author: Camille Pajot

Graphics like the following one can also be produced:

**Various studies could be carried out:**

- Balancing between CO2 emissions from the LNCMI and district heating, free profile
- Using HP according to the electricity price, typical profiles
- Study of operational performances under constraints, fixed profile
1.4. Waste heat recovery

Fig. 6: Figure 6: principle diagram of the waste heat recovery example with multi-stakeholder vision | Author: Lou Morriet from Camille Pajot work

Fig. 7: Figure 7: heat provider of the district over a year | Author: Camille Pajot
OMEGAlpes article study cases

Please have a look to the following case study:

- **Basic example: PV self-consumption**
- **Multi-actor modelling for MILP energy system optimisation: application to collective self consumption**
- **Other case studies**

The code is stored at the Gitlab: OMEGAAlpes Examples in the folder “article_case_study”. Some of them have also been developed on a Jupyter notebook for a better understanding.

**Note:** To know how to run example python codes or notebooks, see: Help run Jupyter Notebook or Help run example

Consider this address to run notebooks with Binder: 
https://gricad-gitlab.univ-grenoble-alpes.fr/omegalpes/omegalpes-notebooks
Consider this address to run the examples:
https://gricad-gitlab.univ-grenoble-alpes.fr/omegalpes/omegalpes_notebooks

### 2.1 Basic example: PV self-consumption

In this PV self-consumption example, a single-house with roof-integrated photovoltaic panels (PV) is studied. Specifically, the study case is about demand-side management in order to maximize the self-consumption, by shifting two
household appliances consumption (clothes washing machine and clothes dryer) and using a water tank for the domestic hot water consumption.

The article presenting this example is available at: PV self-consumption article
You can access to this example via an online Notebook at: PV self-consumption online notebook
The code is available here: BS2019 PV self-consumption code
The Notebook is available at: PV self-consumption online notebook

This example leads to a study with
- 6922 variables (2890 continuous and 4032 binary)
- 79172 non-zeros
This optimization problem has been generated within 1.2 seconds on an Intel bicore i5 2.4 GHz CPU.
An optimal solution was found in 43.6 seconds with the free CBC solver available in the PuLP package, and in the 2.5s with the commercial Gurobi solver.

2.2 Multi-actor modelling for MILP energy system optimisation: application to collective self consumption

The shape of an energy project depends on the available technologies but also on stakeholders’ decisions although most energy-support-decision tools only focus on technical issues.
We aim to propose a multi-actor modelling based on actors’ objectives and constraints and to apply it on the model generation tool for optimization OMEGAlpes. This modelling aims to help stakeholders to formalize their constraints and objectives and to negotiate them in a multi-actor design process. This modelling has been applied to a simplified collective self-consumption project.

The code is available here: BS2019 multi-actor Modelling code
The article presenting this example is available at: BS2019 multi-actor Modelling article
You can access to this example via an online Notebook at: BS2019 multi-actor Modelling notebook
The code is available here: BS2019 multi-actor Modelling code

2.3 Other case studies

Camille Pajot’s PhD will be available here in French: OMEGAlpes Outil d’aide a la decision pour une planification energetique multi-fluides optimale à l’échelle des quartiers


Camille Pajot, Nils Artiges, Benoit Delinchant, Yves Marechal “Optimal Heat Pumps Operation For Demand Response of Residential Buildings At District Scale”, BS19, Building Simulation Conference, Roma in September 24, 2019. If needed, have a look to the help on:
CHAPTER 3

How to run an example

You first need to download (or clone) the OMEGAlpes Examples folder (repository) at: OMEGAlpes Examples. In fact, it is better to download the whole folder as most of the examples or article case studies use data located outside the code file. Thus they need to be download as well.

Then, open your development environment, select the example file you want (.py) and run it.

Do not forget:
To run your example, you first need to install OMEGAlpes library.
To do so, please, have a look to the documentation: OMEGAlpes Installation
Or to the README.md of OMEGAlpes Gitlab
CHAPTER 4

How to run a project with Jupyter Notebook

What is Jupyter notebook? This is the web application linked to the OMEGAlpes project which has the ability to execute code from the browser, with the results of computations attached to the code which generated them. If you followed the standard install instructions, Jupyter is actually running on your own computer. It’s your computer acting as the server.

4.1 I. How to use Jupyter Notebook remotely on Binder

1. Go on https://mybinder.org/
2. Type the following repository URL: https://gricad-gitlab.univ-grenoble-alpes.fr/omegalpes/omegalpes_notebooks
3. And select ‘Git Repository’
4. Click on ‘Launch’
5. Click on the Notebooks folder
6. Select the Notebook you want (.ipynb) and run it (see II.2 if needed)

4.2 II. How to use Jupyter Notebook locally

1. Install Jupyter (see II.1.1 - Anaconda - or II.1.2 - pip -)
2. Download the following folder: https://gricad-gitlab.univ-grenoble-alpes.fr/omegalpes/omegalpes_notebooks
3. Open the command prompt
4. Install OMEGAlpes (ex: on the command prompt type ‘pip install omegalpes’ or have a look to the documentation https://omegalpes.readthedocs.io/en/latest/installation_requirements.html)
5. On the command prompt select the folder of the OMEGAlpes Notebooks
6. On the command prompt type:
examples Documentation

    python -m notebook

or:

    jupyter notebook

7. Click on the Notebooks folder
8. Select the Notebook you want (.ipynb) and run it (see II.2 if needed)

4.2.1 II.1.1 Installation through Anaconda

On Linux

To install Anaconda, follow these steps:
1. Download Anaconda 5.2 with Python Distribution Python 3.6 version
2. Save the folder in your “Downloads” (It is according to the language (ex: Telechargements in French))
3. Open your command window and install it by using the following command :
   bash~/Downloads/Anaconda3-5.2.0-Linux-x86_64.sh
4. Agree all the terms and say yes at the end. You can choose or agree to the saving locations for the folder.
5. Close and open a new terminal to make the installation effective
6. To open Jupyter notebook, use the following command : jupyter notebook

On Windows

Just download the package with the latest version of Python 3 and Jupyter will be installed with it. When asked if you want to “add to path”, tick the box even if it’s not recommended. Otherwise, you can add Jupyter to the path -to be able to launch it from the terminal- by . Or if you installed with Anaconda you can open Jupyter directly among your programs (by clicking on the Start button), or from the Anaconda Navigator.

4.2.2 II.1.2 Installation through PIP

On Linux

If you have Python already installed on your computer, you can directly install Jupyter Notebook by using pip on the terminal by using the following command : pip install.

Note that the Python versions 2.7 or 3.3 or higher are required. For upgrading your Python version, type “sudo apt-get install python[the version you want]” on the terminal (example: “sudo apt-get install python3”).

You should also know that, by installing Jupyter with this method, you will not install all the Python libraries that are included on the Anaconda folder, you will only install Jupyter Notebook. So if later you need a library that you don’t have installed already, you will have to download and install it separately.

To install, follow these steps :
1. To make sure that you have the latest pip version, type on the terminal the following command line: “pip3 install --upgrade pip” (otherwise try “python -m pip install --upgrade pip” or “python -m pip install -U pip setuptools” or “sudo apt install python-pip”).
2. Then install Jupyter Notebook by typing “pip3 install jupyter” (or “pip install jupyter”) (you might need to reinitialize the computer after this step if the next step doesn’t work).
3. Finally, to open Jupyter, type “jupyter notebook” on the terminal. This will start a server and Jupyter Notebook will pop-up on a browser, on localhost:8888. Leave the server running on the terminal until you’re finished.

**On Windows**

Run the following commands on the terminal:

1. “py -m install –upgrade pip” to make sure you have the latest version of pip.
2. “py -m pip install jupyter” to install Jupyter Notebook.
3. “py -m notebook” to launch Jupyter from the terminal.

If any of these commands don’t work, try replacing ‘py’ with ‘python’ or ‘python3’.

**4.2.3 II.2 Use Jupyter Notebook locally**

For complete explanations on how to deal with Jupyter Notebook, go to the following link: [https://www.codecademy.com/articles/how-to-use-jupyter-notebooks](https://www.codecademy.com/articles/how-to-use-jupyter-notebooks)

To run:

Choose the case example you want to work in.

Then, run cells after cells after completing the inputs.
Your folder saved in your desktop (local work)

Click on one of the Simulation files to open an example of a microgrid Energy System

All the documents which composed your folder

Fig. 1: Figure 1: Presentation of the dashboard
Fig. 2: Figure 2: How to run the cells

Name of the file you are working on

To run or stop running cells (on after the other)

To save your modifications (when you will have to change the inputs for example)