Technical Report

Abstract:

This short report explains an optimization simulator. The simulator is supposed to find out the optimum energy flow between Grid and eight schools of UPM to maximize the earnings of the schools. This simulator assumes that all the schools are facilitated with rooftop solar panels and storage capacities. At each time interval, which is one hour, each school can buy or sell energy to the grid or other schools. Moreover, each school can charge or discharge its storage unit at each time step. The simulator must make sure each school has enough energy to satisfy the demand at the corresponding time step.

Data:

To run the simulator, we need consumption, generation, storage, prices, and location datasets of each school.

Consumption: these datasets are received from the schools. Each dataset is composed of 8760 records where each record shows the consumption of the school during the corresponding hour.

Generation: the generation datasets are prepared by an online platform called PVWatts Calculator[1]. To measure the potential generation of the schools, the usable surface of the rooftop of the schools along with properties of a typical solar panel is selected and the potential generation dataset of each school is produced by the platform. As same as the consumption, the generation datasets are based on KWh and each dataset is composed of 8760 records.

Storage: Considering the intermittent inherent of renewable energies, these green energies, in our case solar energy, must be combined with storage units. In this simulator, the same storage unit is considered for all the schools. The properties of the storage unit are detailed in table 1. As the table shows, five parameters must be defined for each storage unit. This simulator considers a 10% safety limit for SoC of the batteries. The SoC of the batteries can't be higher than 95% or less than 5%. Moreover, C-Rating is considered to one capacity which means the battery can be charged and discharged totally in each time step. Lastly, maximum degradation of battery is set to 20% of the capacity. If the degradation reaches that level, the battery must be replaced with a new one.

Table 1: Properties of storage unit

Capacity (KWh)	Initial SoC (%)	C-Rating	Maximum Degradation (%)	Price
30	5	1C	20	2000

Prices: Since there are three types of transactions in the considered network, three types of prices must be defined for the simulator. The considered prices are the price of buying energy from the

[1] https://pvwatts.nrel.gov/

grid, the price of selling energy to grid, and finally the price of selling energy between the schools. For the first two prices the data of the year 2022 are downloaded from redelectrica[2]. The last price in this simulator is measured based on the price of buying energy from the grid. The price of selling energy in between schools is 80 percent of the price of buying energy from the grid at the corresponding time step.

Location: we must consider the physical locations of the schools to find out the distances between them. An important factor that affects the transferring of energy is the loss of energy. Regarding the fact that the considered schools are close to each other, the loss of energy is set to zero.

Code:

The simulator is designed in python with pyomo package. There are three main python files, Optimized, linearopt, and Functions that simulator works based on them.

Optimized: to run simulator this file must be run, and the rest of the code will run automatically. In this file, after reading the names of the schools, the initial degradation of the storage units is set to zero. Then, the initial SoC of each storage unit will be read. Afterwards, the main loop (line 27) starts the simulator. The designed simulator works based on the days. It means that it receives the data of each day and optimizes the flow of energy for that day. Line 30 runs the optimization section. This line receives three outputs from the optimization section, which are payment, degradation, and SoC, respectively. The simulator uses degradation and SoC to run the next day optimization.

Linearopt: This file includes a class named LinearOptimization. At the moment this class includes one function, Gurobidays; However, it can be expanded to other functions based on the different solvers. The function, Gurobidays, starts with reading the information like the name and storage capability of the schools and prices. The most important part of this section is line 51 where the cost of degradation is measured. The cost of degradation of each unit is measured based on the price, the capacity, and the maximum degradation of the battery. The rest of this is described as follows:

Line 54: Defining the model. The concrete model of pyomo is considered in this simulator.

Lines 60-97: Define the optimization variables.

Lines 101-107: Define the objective function of the optimization. The objective is maximizing the earnings of the schools.

Lines 111-403: Define 33 constraints of the optimization model (All of them are explained in the code).

[2] https://www.ree.es/es

Lines 405-407: Define solver and solve the model.

Lines 409-419: These values must be saved in a excel sheet.

Functions: All the additional functions which are needed to run the simulator are saved in this file.