Scientific Bulletin of Naval Academy

Mix-generation optimization for electricity market simulation

Mix-generation optimization for electricity market simulation

Adela Bâra, Simona-Vasilica Oprea, Gabriela Ene

Bucharest University of Economic Studies, Romania
E-mail: simona.oprea@csie.ase.ro

Abstract. Owning several types of generating units requires an optimized schedule to cover the negotiated bilateral contracts. This approach will lead to a better electricity market strategy and benefits for an electricity producer. In this paper, we will simulate the operation of five different generators including generators based on Renewable Energy Sources (such as wind turbines and photovoltaic panels) that belong to an electricity producer. The five generators are modelled considering the specificity of their type and primary energy source. For instance, for renewable energy sources, we will consider the 24-hour generation forecast. The objective function of the optimization process is to obtain an optimal loading of generators, while the constraints are related to the capacity and performance of the generators. The output consisting in a generating unit optimized operation schedule will be further used for day-ahead or balancing market bidding process. Hence, the producer will be able to adequately bid on the future electricity markets knowing the commitment of generators for negotiated bilateral contracts market. The simulations are tested for more than five generators considering the connection to a relational database where more data for generators is stored.

1. Introduction and literature review

Different electricity markets require various approaches. Producers can take advantage of all types of electricity markets, such as bilaterally negotiated contracts centralized market (BNCCM), day-ahead market (DAM), ancillary service market (ASM) and balancing market (BM). On BNCCM, the producers negotiate in advance long-term contracts with electricity suppliers or large consumers considering the specificity of this market.

In Romania, the BNCCM is characterized by several standard products for trading electricity. They are set for certain time intervals: year, semester, month, day; and it consider certain days (from Monday to Friday or from Monday to Sunday) and hour intervals, for peak or off-peak [1]. The quantities are fixed for the entire interval and the prices are usually lower than the prices that could be obtained from other markets, but the risks associated with this market are the lowest.

DAM is characterized by 24 hour-auctions based on uniform price mechanism that identifies the most competitive generating units that match the demand and are accepted for trading [2]. Two to four intraday markets also operate to tune the results of the DAM.

In certain condition, if they qualify, the producer may approach the ASM that has two income components: first is the reserving capacity and second is the activation of the reserve. 3% of the available power is considered mandatory for each producer to contribute to the primary reserve. For ASM, the market pricing in Romania is given by uniform-price auction system. This approach is common for most of the capacity markets over the world.

BM is mandatory for electricity producers; they have to offer in both directions (to increase generation up to maximum capacity or decrease generation to zero or to minimum technic). Hence, double sets of hourly offers are sent to the transmission system operator that manage the BM. The

market mechanism is based on pay-as-bid pricing system [3]. Two sets of prices are obtained: for increasing and decreasing the generation. The difference between deficit and surplus prices could be significant.

Considering an electricity producer with several generating units know as portfolio, the engagements assumed on the BNCCM could be fulfilled in different ways taking into account the diversity of generators. However, the producer is interested to minimize the operational costs and therefore obtain the optimal loading of his generators. Also, the daily operation scenario may change, so that the producer can not rely on the same operating schedule of his generators.

In this paper, we will model the generators and perform a simulation to get the optimal loading that covers the contracts obligations signed on the BNCCM. The output of the simulation will be used as input data for the next electricity markets (DAM, ASM or BM). Thus, after using the simulator, the producer knows the rest of the generation potential for each generating unit for the rest of the electricity markets.

A similar approach was presented in [4], considering a producer that approaches different electricity markets. It proposed a stochastic programming methodology for a producer to obtain the optimal bidding strategies for DAM.

In case of microgrids with isolated operation sometimes, the optimal scheduling of the generating units, storage devices and modern appliances can be performed using Mixed-Integer Linear Programming (MILP), General Algebraic Modeling Systems (GAMS) [5], Genetic Algorithms, etc.

2. PYPSA package description

Acknowledging the complexities of power systems tools and the variety of needs that emerge at the consumer level, the availability of models for managing formulations of entire power flows and optimization of power storages has become a requirement for fast implementation and analysis of systems. Given the demand increase in varied sectors in regards with electrification, the need for elaborate software tools that can model the power system has grown accordingly. Another important key in the development of models for decentralized systems that use renewable energy is related to the optimization of electrical power systems over different periods of time.

Python for Power System Analysis (PyPSA) performs simulations and optimizes the power systems [6]. As described in the official documentation, the library encapsulates multiple components: Network, SubNetwork, Bus, Carrier, GlobalConstraint, Line, LineType, Transformer, TransformerType, Link, Load, Generator, StorageUnit, Store and ShuntImpendace. The Network component is the container of all components, and every component is attached to the nodes, which are represented by the Bus component. The power balance of one bus is determined by the following components: Loads, Generators, StorageUnit, Store and ShuntImpendance. The Store offers optimization and control over the energy capacity and it is used only for storing the energy. Two buses of given impedance are connected by lines and transformers. The power balance of one bus is determined by the following components: Loads, Generators, StorageUnit, Store and ShuntImpendance. Energy enters and leaves the model through certain components and flow is determined by the level of energy before and after the simulation or by checking the efficiency. The Link component controls flow between two buses with different energy carriers, while efficiency and marginal costs can be handled. Therefore, PyPSA can resemble a more general energy network but proposes also optimization for other energy sectors, such as transport or heating. Other components can be easily added by extending the existing ones, so the toolbox scales well over long time series or large networks.

Data related to the network components are stored in DataFrame objects that belong to the pandas library available in Python. This provides efficient methods of data representation at every component level and all matrix calculations are handled using additional Python libraries such as NumPy and SciPy. Each DataFrame object can have either static attributes or a dictionary of time-varying attributes, that can be accessed from the network object. The problem of optimal power flow is

handled under two modules pypsa.opf and pypsa.linopf that support linearized power flow equations for AC and DC networks, offering optimization of the dispatch of generation and storage and the transmission infrastructure. The objective function used is the total system cost for the optimized snapshots, while each snapshot can be given a weight, to represent multiple hours. Considering the weights as probabilities, a stochastic optimization can be obtained [7].

In [8], PyPSA was found to be amongst the four open-source tools that are considered to be mature enough in comparison with proprietary tools for serious use. Also, a big test suite built under PyTest is available, including tests that compare the results between PyPSA and other similar software (pandapower, PYPOWER).

3. Simulation and results

Using the Python library PYPSA [6], a module can be created to optimize the operation of the operating generators that belong to an electricity producer. Similarly, in Matlab, the optimal operation of the generating units can be performed [9].

To determine the optimal operation of the generators, the generators are modeled, considering the following attributes (as in Table 1):

Table 1. The attributed of the generator modelled in PYPSA

No.	Attribute
1	bus
2	control
3	type
4	p_nom
5	p_nom_extendable
6	p_nom_min
7	p_nom_max
8	p_min_pu
9	p_max_pu
10	p_set
11	q_set
12	sign
13	carrier
14	marginal_cost
15	capital_cost
16	efficiency
17	committable
18	start_up_cost
19	shut_down_cost
20	min_up_time
21	min_down_time
22	up_time_before
23	down_time_before
24	ramp_limit_up
25	ramp_limit_down
26	ramp_limit_start_up
27	ramp_limit_shut_down
28	p_nom_opt
29	Name:

In the simulation, five generators were considered, of which three are conventional (one on coal (coal) and two on gas (gas1 and gas2)) and two based on renewable sources (wind and solar or photovoltaic).

A virtual generator is also considered to adjust the situation when the existing generators are not able to supply the demand or satisfy the contracts signed on BNCCM. It will provide an indication when the producer has to buy on the DAM to fulfil the BNCCM engagements.

To cover a certain consumption (Figure 1) for the next day, we aim to determine the optimal operation of the generators. The consumption or demand represented in Figure 1 could be the daily consumption curve for the next day of a microgrid, a large consumer or the sum of engagements taken

on BNCCM. After supplying this demand, the producers find out the remaining generating potential for each generating unit.
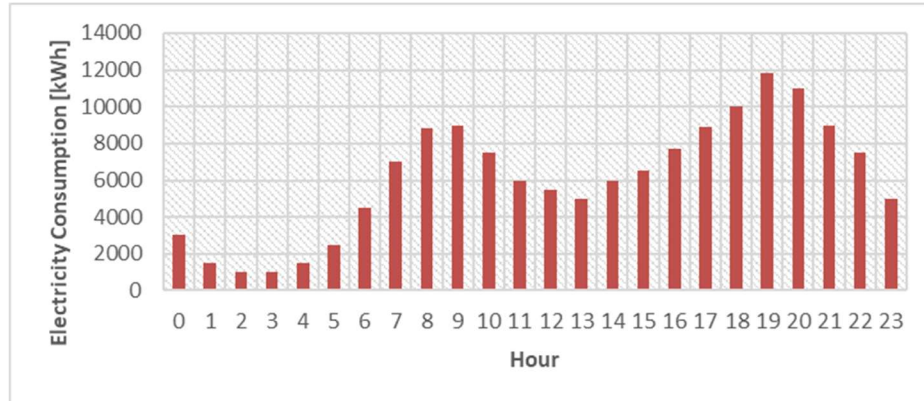

Figure 1. Daily consumption curve for the next day

The five generators were modeled, according to their specificity, according to the source code provided in Table 2. For setting the marginal costs, we consulted data provided in [10], [11].

Table 2. Modelling of the generators in PYPSA

```python
import pypsa
nu = pypsa.Network()
nu.set_snapshots(range(24))
nu.add("Bus","bus")
nu.add("Generator","coal",bus="bus",
        committable=True,
        p_min_pu=0.25,
        marginal_cost=30,
        start_up_cost=1000,
        shut_down_cost=50,
        min_up_time=1,
        min_down_time=1,
        ramp_limit_up=0.1,
        ramp_limit_down=1,
        p_nom=10000)

nu.add("Generator","gas1",bus="bus",
        committable=True,
        marginal_cost=70,
        p_min_pu=0.1,
        start_up_cost=0,
        shut_down_cost=0,
        min_up_time=0,
        min_down_time=0,
        p_nom=1500)
nu.add("Generator","gas2",bus="bus",
        committable=True,
        marginal_cost=50,
        p_min_pu=0.1,
        start_up_cost=0,
        shut_down_cost=0,
        min_up_time=0,
        min_down_time=0,
        p_nom=1500)
#variable generation
nu.add("Generator", "wind", bus="bus",
                committable=True,
                p_nom=600,
                marginal_cost=10,

p_max_pu=[0.3,0.2,0.3,0.6,0.4,0.5,0.7,0.6,0.6,0.4,0.8,0.7,0.6,0.7,0.6,0.6
,0.7,0.8,0.5,0.5,0.6,0.5,0.3,0.4])
nu.add("Generator", "solar", bus="bus",
                committable=True,
                p_nom=300,
                marginal_cost=10,

p_max_pu=[0,0,0,0,0,0,0,0.2,0.3,0.5,0.6,0.7,0.8,0.9,0.98,0.9,0.7,0.5,0.4,
0.2,0,0,0,0])
nu.add("Load","load",bus="bus",p_set=[3000,1500,1000,1000,1500,2500,4500,
7000,8800,9000,7500,6000,5500,5000,6000,6500,7700,8900,10000,11800,11000,
9000,7500,5000])
print(nu.lopf(nu.snapshots))
print(nu.generators_t.status)
print(nu.generators_t.p)
```

The renewable generators, known as variable generation, are modelled differently compared with conventional generations. In their case, we considered the generation forecast, providing hourly values for attribute *p_max_pu*. In Table 3 and Figure 2, we obtained the optimal loading of the generators. The contractual engagements are modelled as hourly load (using the attribute *p_set*) that should be supplied. In case the five generators are not able to supply the load, the producer has to buy electricity to supply the difference.

Table 3. Optimal loading of the generators using PYPSA

hour	coal	gas1	gas2	wind	solar
1	0	1320	1500	180	0
2	0	0	1380	120	0
3	0	0	820	180	0
4	0	0	640	360	0
5	0	0	1260	240	0
6	0	700	1500	300	0
7	4500	0	0	0	0
8	5500	0	1080	360	60
9	6500	350	1500	360	90
10	7500	0	1110	240	150
11	6840	0	0	480	180
12	5370	0	0	420	210
13	4900	0	0	360	240
14	4346	0	0	384	270
15	5346	0	0	360	294
16	6270	0	0	0	230
17	7270	0	0	420	10
18	8270	0	0	480	150
19	9270	0	310	300	120
20	10000	0	1440	300	60
21	10000	0	640	360	0
22	8700	0	0	300	0
23	7320	0	0	180	0
24	4760	0	0	240	0
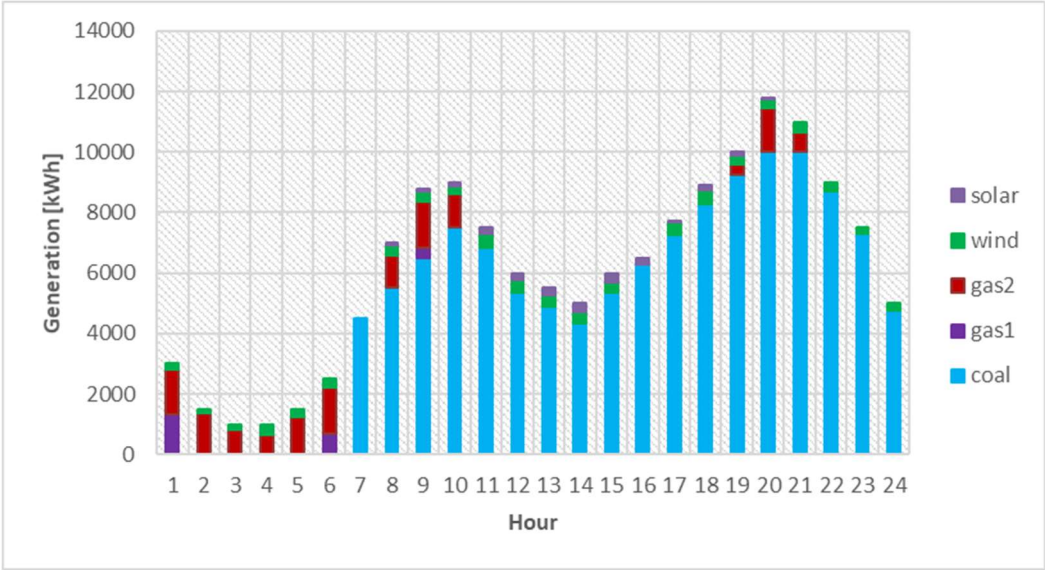
Figure 2. Optimal loading or operation of the five generation

As in Figure 2, the coal-fired generator covers most of the day's consumption, except for peaks and night gaps where gas generators and renewable energy sources generators operate (solar and wind at morning / evening peak, wind at night-time).

References
[1] OPCOM, "Bilaterally negociated contracts centralized market - standard products."
[2] OPCOM, "Day-Ahead Market."
[3] OPCOM, "Balancing Market."
[4] M. A. Plazas, A. J. Conejo, and F. J. Prieto, "Multimarket optimal bidding for a power producer," *IEEE Trans. Power Syst.*, 2005.
[5] H. Morais, P. Kádár, P. Faria, Z. A. Vale, and H. M. Khodr, "Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming," *Renew. Energy*, 2010.
[6] T. Brown, J. Hörsch, and D. Schlachtberger, "PyPSA: Python for power system analysis," *J. Open Res. Softw.*, 2018.
[7] "PyPSA: Python for Power System Analysis."
[8] M. Groissböck, "Are open source energy system optimization tools mature enough for serious use?," *Renewable and Sustainable Energy Reviews*. 2019.
[9] MathWorks, "Optimal Dispatch of Power Generators: Problem-Based."
[10] Wikipedia, "Cost of electricity by source."
[11] LAZARD, "Lazard ' s levelized cost of energy analysis-version 12.0."