A Web Application for the Simulation of Day-Ahead Energy Markets Part II Back-end Implementation

Martyn van Dijke, *Department of Electrical Engineering, Electrical Energy Systems (EES), TU/e*

***Abstract*—Market clearing algorithms have been invented as early as the 1920's and have been improved upon the decades after. Advancements in market clearing algorithms have been made under the rising development of computer-aided computation algorithms and now a days have formed the go to standard for solving the day ahead market clearing. While many market clearing algorithms have been invented and have been developed into an application there is not an interactive way to use market clearing models, without installing complicated and often, closed source software that is hard to modify or even impossible to modify. The purpose of this paper is to present different models by which market clearing can be achieved and that are embedded in the back-end of the developed web application.**

***Index Terms*—Day ahead, Mixed Integer Linear Programming, market clearing, modelling, network, Python, stochastic, simulation, web**

NOMENCLATURE

A. Indices

t	Index of time instances running from 1 to N_T
i	Index of generators running from 1 to N_I.
j	Index of loads running from 1 to N_J.
n	Index of bus labels running from 1 to N_N.
l	Index of branches running from 1 to N_L
w	Index of wind output power scenarios running from 1 to N_W

B. Variables

c_i	Marginal cost for generator i [€/MW].
$u_{i,t,n}$	Binary status variable $\in \{0,1\}$ for generator i attached to bus n on period t, if unit i is scheduled to be on during period t the binary status variable is set to 1.
K_{nl}	Incidence matrix taking values $\in \{-1,0,1\}$ depending if branch l starts or ends at bus n
$p_{i,t}$	Power generated by generator i in time period t [MW].
$P^W_{w,t}$	Wind power generation [MW].
$P^{\varphi}_{i,t,w}$	Random variable modeling the wind power generation for unit i during period t and scenario w.
$l^s_{j,t}$	Involuntary load shedding on consumer j during period t [MW]
$l_{j,t}$	Power consumed by load j in time period t [MW].
$\theta_{n,t}$	Voltage angle at bus 0.
$\theta_{m,t}$	Voltage angle at bus 1.
$s_{i,t,w}$	Wind power generation spillage during period t and scenario w [MW].
t^{mu}	Minimum time periods generator needs to be up [h].
t^{md}	Minimum time periods generator needs to be down [h].
$rd_{i,n}$	Ramp down of generator i attached to bus n [MW/h].
$ru_{i,n}$	Ramp up of generator i attached to bus n [MW/h].
$r^u_{i,t}$	Spinning upward reserve scheduled for generator i during period t limited by $r^{d,max}_{i,t}$ in [MW].
$r^d_{i,t}$	Spinning downward reserve scheduled for generator i during period t, limited by $r^{u,min}_{i,t}$ in [MW].
$r^u_{i,t,w}$	Deployed reserve up by generator i during period t and scenario w
$r^d_{i,t,w}$	Deployed upward reserve by generator i during period t and scenario w t and scenario w [MW].
$r^{u,min}_{i,t}$	Minimum spinning upward reserve in [MW].
$r^{d,max}_{i,t}$	Maximum spinning downward reserve in [MW].
x_l	Reactance between bus 0 and bus 1 [Ω].

C. Constants

p^{min}_i	Minimum power output of generator i [MW].
p^{max}_i	Maxium power output of generator i [MW].
F_l	Capacity of branch l [MVA].
$c^{sd}_{i,n}$	Shutdown costs of generator i attached to bus n on time period t [€].
$c^{su}_{i,n}$	Start up costs of generator i attached to bus n on time period t [€].
$v^{lol}_{j,t}$	Value of load shedding for consumer j during period t [€/ MWh].
π_w	Probability of wind scenario w.
v^{ws}_w	Cost of wind spillage [€/ MWh].

I. INTRODUCTION

WHOLESALE electricity markets have been cleared with the aid of computer algorithms for more than 35 years [1]. While the core basis of all market clearing algorithms used nowadays is the same, different models are used depending on the country in question [2].

Three main techniques for market clearing are distinguished in this paper: 1) basic model: which relies on the solution of a simple economic dispatch problem with a reduced set of constraints. 2) network model: which incorporates the technical constraints of the generating units and the limitations of the network and is based on solving a network-constrained unit commitment problem. 3) stochastic model: a model that

expands the previous models by considering the stochasticity in the generation of renewable energy based production units and further considers the joint scheduling of reserve services. Due to time limitations, only the two first models are incorporated in the current version of the web-application.
However, this paper investigates the advantages and disadvantages of all three models.
These models, as well as their extensions, are programmed in Python [3] and form the back-end of the interactive web application that has been developed within this project.
It is to be noted that the front-end of the web application is explained in-depth in part I, authored by Y. Li.

II. Literature Survey

Market clearing algorithms are periodically ran by Market or System Operators in order to ensure that the electricity demand is covered in an economically optimal fashion, while ensuring that the power system variables are kept within a narrow range around their nominal operating values.
As a result, numerous models have been presented so far and this research area is well-studied field [4],[5] for network-constrained unit commitment problems and [6],[7] for stochastic two-stage reserve clearing problems.

Literature regarding only the basic market clearing model has proven to be hard to find, since it has no real applications other than explaining the basic theory behind market clearing. However, this model is essentially the basis of most market clearing engines used in centralised European electricity markets where bids and offers are cleared on the basis of welfare maximization. Nevertheless literature in the form of books [8],[9] have been used to research the algorithm.
Next to the books found, literature from the network constrained case has also been used for the basic economic dispatch model since it requires less constraints then the full network-constrained unit commitment while largely optimizing the same objective function.

Literature regarding the network constrained unit commitment case, is more commonly available.
A large number of countries all over the world [10] have implemented a network constrained model in their day ahead market clearing to dispatch electrical energy accordingly.
This is even done in real time, to ensure that imperfections in the predicted load consumption or mechanical failures are covered. The problem area is not novel [8] and is still in active research.

Literature regarding stochastic models is not uncommon and relatively new in the market clearing history.
Market clearing algorithms using stochastic optimisation have seen a sharp increase in research over the last decade.
This is mainly due the enormous increase in renewable electrical energy such as wind energy in today's society.
The main goal of joint energy and reserve clearing model is to provide a prediction in the number of scheduled real time reserves that are needed in order to cope with irregularity in the predicted wind output power production [11].
In order to achieve a prediction in the amount of scheduled reserves a two-stage stochastic optimization process [12] needs to be integrated with the network-constrained unit commitment.

III. Back end implementation

The web application can be split into two parts, one part consisting of the back-end where the mixed integer linear programming (MILP) optimization is done and the second part consisting of the frontend where the browser renders the interface. Within the back-end implementation, two working models can be distinguished, and one researched model 1) basic model: basic constraints to match the power output to the load. 2) network model: network constrained unit commitment problem where technical generator constraints are included. 3) stochastic model: joint energy and reserve clearing model based on a two-stage stochastic programming.
All models extend their previous model, the network-constrained unit commitment extends the basic economic dispatch model with more sophisticated constraints and the two-stage stochastic programming model extends the network-constrained unit commitment model and the basic economic dispatch model.
Due to time limitations, no working version of the stochastic model has been developed, and thus only the basic and network models are available in the frontend.
The back-end is implemented entirely in Python and uses the Pyomo [13],[14] modelling language within Python.
The network-constrained unit commitment is made using the PyPSA framework [15].
Both problems are formulated as an MILP problem and can both be solved using the free GLPK (GNU Linear Programming Kit) [16], Gurobi [17], CPLEX [18], Cbc (Coin-or branch and cut) [19] or any other MILP compatible Pyomo solver.

A. Basic economic dispatch

The basic economic dispatch model can be described by an objective function as seen in eq. (1) and considering the constraints from eq. (2) and eq. (3).
Eq. (2) ensures that the generated power matches the load consumption and eq. (3) ensures that the generator does not exceed its operating limits.
From these constraints, it follows that only basic constraints are considered that do not incorporate the full spectrum of technical constraints, such as upward and downward generator limits and network constraints.

$$\underset{c}{\text{minimize}} \quad \sum_t \sum_i c_i \cdot p_{i,t} \tag{1}$$

$$\text{subject to} \quad \sum_i p_{i,t} = \sum_j l_{t,j} \quad \forall\, t \tag{2}$$

$$p_i^{min} \leq p_{i,t} \leq p_i^{max} \quad \forall\, i, t \tag{3}$$

B. Network-constrained unit commitment

The objective function for the network-constrained unit commitment model is given by eq. (4).
The network model extends the elementary case and thus is the full range of constraints given by eqs. (5) to (13) and by eqs. (2) to (3).
Eq. (8) forms the unit commitment constraint and forces the generator to be completely on or off. Eq. (9) formulates the minimum up time a generator has to fulfill once it has been turned on, similarly eq. (10) formulates the minimum down time a generator has to down once it has been turned off.
Eq. (11) formulates the maximum amount of power increase and decrease between the time periods considered.
Eq. (6) and eq. (7) formulate the maximum amount of power a branch (line) is able to transport.
Furthermore, eq. (13) and eq. (12) describe the shutdown costs and the startup costs respectively and have been included in the optimization process to improve modelling of the total costs a of the generator and hence the electricity market.
From above constraints, it can be deducted that the network-constrained unit commitment incorporates multiple technical generators constraints and hence models a more real world generator behaviour that cannot output any desirable power level on any given moment.

$$\underset{c}{\text{minimize}} \quad \sum_t \sum_i \sum_n c_i \cdot p_{i,t,n} + \sum_t \left(c^{sd}_{i,t,n} + c^{su}_{i,t,n}\right) \tag{4}$$

$$\text{subject to} \quad \sum_l K_{nl} f_{l,t} = \sum_i p_{i,t,n} - \sum_j l_{j,t,n} \tag{5}$$

$$f_{l,t} = \left| \frac{\theta_{n,t} - \theta_{m,t}}{x_l} \right| \tag{6}$$

$$f_{l,t} \leq F_l \tag{7}$$

$$u_{i,t,n} \cdot p^{min}_{i,n} \leq p_{i,t,n} \leq u_{i,t,n} \cdot p^{max}_{i,n} \qquad \forall i,t,n \tag{8}$$

$$\sum_{t'=t}^{t+t^{mu}} u_{i,t',n} \geq t^{mu}(u_{i,t,n} - u_{i,t-1,n}) \qquad \forall i,t,n \tag{9}$$

$$\sum_{t'=t}^{t+t^{md}} (1 - u_{i,t',n}) \geq t^{md}(u_{i,t-1,n} - u_{i,t,n}) \qquad \forall i,t,n \tag{10}$$

$$-rd_{i,n} \leq (p_{i,t,n} - p_{i,t-1,n}) \leq ru_{i,n} \qquad \forall i,n \tag{11}$$

$$c^{sd}_{i,t,n} \geq c^{sd}_{i,n}(u_{i,t-1,n} - u_{i,t,n}) \qquad \forall i,t,n \tag{12}$$

$$c^{su}_{i,t,n} \geq c^{su}_{i,n}(u_{i,t,n} - u_{i,t-1,n}) \qquad \forall i,t,n \tag{13}$$

C. Stochastic programming based market clearing

As stated before the stochastic network has not been implemented and thus not tested. However, many works [11] are making use of this model and therefore, a working back-end model is expected to be added in the web app in the future.
The objective function for the stochastic case is given by eq. (14) where SIC is defined by eq. (23) and SDC_w is defined by eq. (24), ε represents the modeling penalties to force the right outcome of the optimization process, and is not further described since no working model has been developed. Since the stochastic case is an extension of the network case and thus also the elementary case the full spectrum of constraints is given by eq. (3), eqs. (6) to (13) and eqs. (15) to (24). Where eq. (22) is the equivalent balance equation eq. (2) from the basic economic dispatch.

$$\underset{c}{\text{minimize}} \quad SIC + \sum_w \pi_w \cdot SDC_w + \epsilon \tag{14}$$

$$\text{subject to} \quad 0 \leq r^u_{i,t} \leq ru_i \cdot u_{i,t} \qquad \forall i,t \tag{15}$$

$$0 \leq r^d_{i,t} \leq rd_i \cdot u_{i,t} \qquad \forall i,t \tag{16}$$

$$0 \leq r^u_{i,t} \leq r^{u,max}_{i,t} \qquad \forall i,t \tag{17}$$

$$0 \leq r^d_{i,t} \leq r^{d,max}_{i,t} \qquad \forall i,t \tag{18}$$

$$0 \leq l^s_{j,t} \leq l_{j,t} \qquad \forall j,t,w \tag{19}$$

$$0 \leq s_{i,t,w} \leq p^{\varphi}_{i,t,w} \qquad \forall i,t,s \tag{20}$$

$$0 \leq p^{wp}_{w,t} \leq +\infty \qquad \forall w,t \tag{21}$$

$$\sum_i p_{i,t} + \sum_w \left(p^{\varphi}_{i,t,w} - s_{i,t,w}\right) = \sum_j \left(l_{j,t} - l^s_{j,t}\right) \tag{22}$$

$$SIC = \sum_t \sum_i \left(c_i \cdot p_{i,t} + c^{ru}_i \cdot r^u_{i,t} + c^{rd}_i \cdot r^d_{i,t}\right) + \sum_t \sum_i \left(c^{sd}_{i,t} + c^{su}_{i,t}\right) \tag{23}$$

$$SDC_w = \sum_t \Big(\sum_i c_i \cdot \left(r^u_{i,t,w} + r^d_{i,t,w}\right) + \sum_j v^{lol}_j \cdot l^s_{j,t} + \sum_w v^{ws}_w \cdot s_{w,t,s} \Big) \tag{24}$$

Eq. (15) formulates the scheduled upwards reserve, similarly eq. (16) formulates the scheduled downwards reserve.
Eq. (18) limits the scheduled downwards reserve to their maximal value, equivalently eq. (17) limits the scheduled upwards reserve to their maximal value.
Eq. (18) ensures that the load shedding is not greater then the load power consumption. Eq. (19) describes the cost of load shedding. Eq. (20) limits the wind spillage during a scenario to the amount of wind power generated.
Eq. (21) describes the power generation from a renewable wind power unit. Eq. (22) describes the balance equation that matches the load consumption to the power output generation.
From above constrains it can be deducted that the main focus of the stochastic model is to schedule for real time reserves in the form of upwards and downwards reverses to cover unpredicted wind output power fluctuations.
To dispatch this model a two stage stochastic optimization has to be preformed, where the first stage optimization is more or less the same as the network case. And the second optimization stage where wind power output production and reserves are included in the market clearing, to clear a more economically efficient market, since the cost of wind power output is zero in an ideal case.

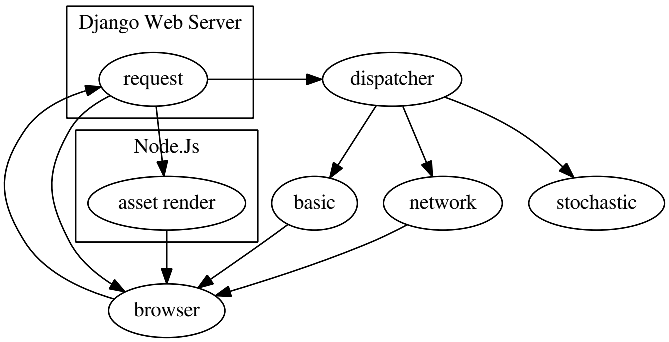


Fig. 1: Overview of the frontend & back-end

IV. FRONT-END IMPLEMENTATION

The frontend is implemented in React [20], while Django [21] ensures the communication to the back end.
Node.Js [22] is further used to compile the React files into browser renderable JavaScript files.
Fig. 1 shows an global overview of the entire system with the respective sub systems. Two subsystems can be identified in fig. 1 the Django web server and the separate Node.Js server for asset compiling. The Node.Js server is a separate server, only used once to compile the React files into one JavaScript file that the browser is able to render.
Once the React files are compiled the Node.Js server is not called anymore and Django uses the compiled JavaScript file that has been generated by the Node.Js server.
Fig. 2 shows the flowchart of the main system, without the separate subsystems for storing and retrieving data from and to the back-end.
Fig. 2 depicts the separate subparts for the front-end and the back-end respectively and indicates the data flow from the front-end to the back-end and vice versa.

V. RESULTS

The results of this back-end part of the two-part project are given below and focus on the basic economic dispatch model and the network-constrained unit commitment.

A. Basic economic dispatch model

Testing of the basic economic dispatch model is done using the data provided in table I for the generator and table II for the load properties. The power set points generated by the algorithm for the generators are depicted in fig. 3.
From fig. 3 it follows that the basic economic dispatch model performs well since the model is only considering a small amount of constraints the solution time normally lay in the order of $0.5 - 1.5\,s$ and are very acceptable optimization times.

B. Network-constraint unit commitment

Testing of the network constraint algorithm is done using the IEEE RTS test data [23].
Gurobi, GLPK and the Cbc solvers been used to solve the IEEE RTS problem. Fig. 4 demonstrates the slowdown in solution time of incorporating the unit commitment constraints

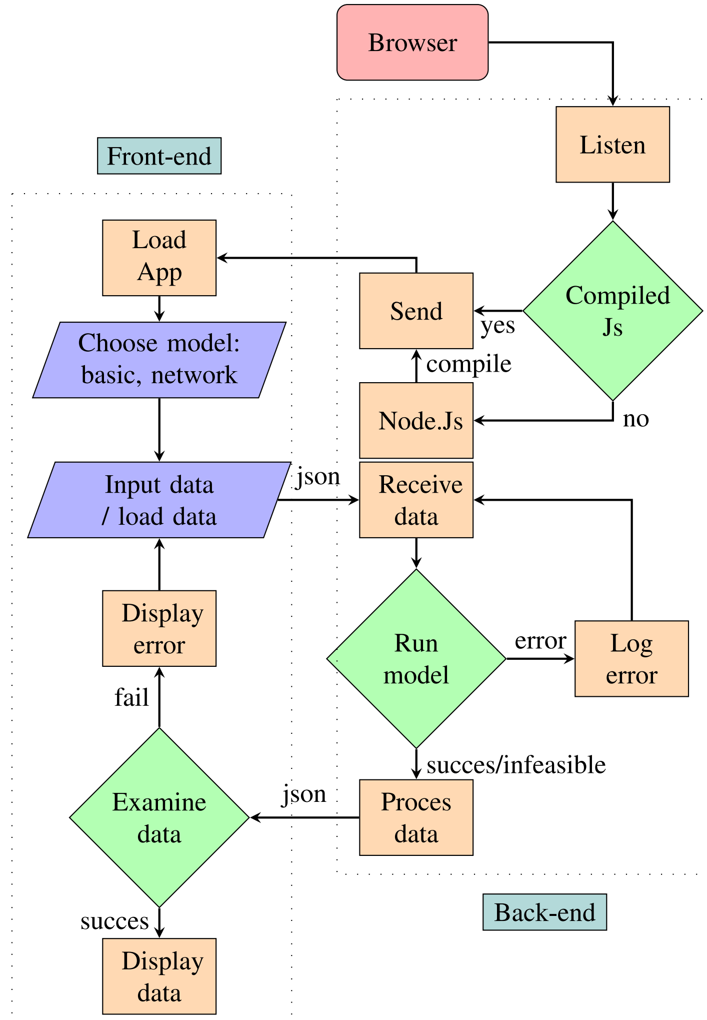


Fig. 2: Flowchart of the main system

TABLE I: Generator data for the basic economic dispatch model

Generator	c	p^{min}	p^{max}
1	30	20	100
2	2	10	30

TABLE II: Load data for the basic economic dispatch model

Time	t_1	t_2	t_3	t_4	t_5	t_6
Load	30	40	50	60	70	80

eq. (8) till eq. (13). The optimization problem has been run 100 times on all solvers with and without unit, commitment enabled, with the exception of GLPK in the unit commitment case where the problem has been run 10 times.
The differences in solution time are depicted in fig. 4a where unit commitment is enabled and in fig. 4b where unit commitment is disabled, the resulting data is also given by table III.
Where case 1 solves the problem with the unit commitment equations enabled and case 2 where the problem is solved without the unit commitment equations.
The simulations have been excuted on an Intel Core i7-4700MQ, with a clock frequency of 2.4GHz and 3.4GHZ boost

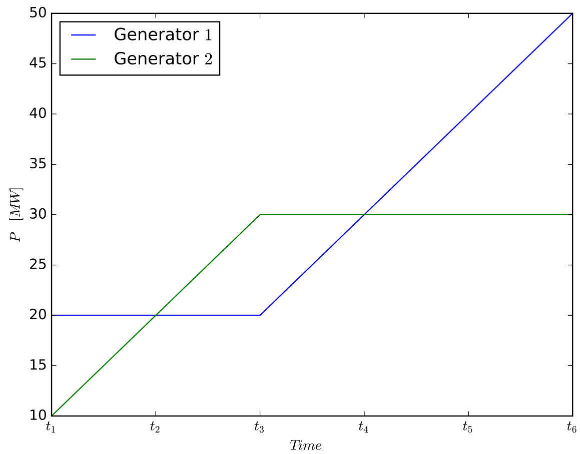
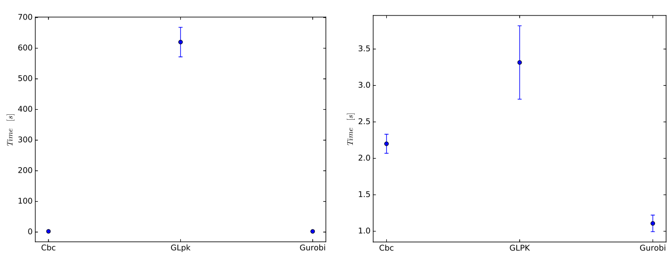


Fig. 3: Generators power output over time with basic economic dispatch model



(a) With unit commitment (b) Without unit commitment

Fig. 4: Solution time with Cbc, Gurobi and GLPK solvers

frequency. Furthermore Python 3.6, Pyomo 5.1.1 and PyPSA 0.9 have been used to solve the IEEE RTS problem. During the simulations the computer is kept idle.

From the data provided in table III it may be concluded that the Gurobi solver is the fastest in both cases.

While it must be noted that the differences with the free Cbc solver are insignificant in the first case and only slightly notable in the second case. For yet unknown reasons GLPK drastically increases the solution time once unit commitment has been enabled.

To verify that the IEEE RTS test data works on the network-constrained unit commitment fig. 5 and fig. 6 have been plotted. Fig. 5 depicts the generator power output set point over time and fig. 6 depicts the bus power flow over time.

From these figures it may be concluded that the algorithm works and dispatches the electrical energy correctly.

TABLE III: Solver speed

Solver	Average time $[s]$	Standard deviation $[s]$
Cbc case 1	2.3730 s	0.6472 s
Gurobi case 1	2.2373 s	0.6910 s
Glpk case 1	619.9920 s	48.2153 s
Cbc case 2	2.2004 s	0.1306 s
Gurobi case 2	1.10783 s	0.1138 s
Glpk case 2	3.3160 s	0.5036 s

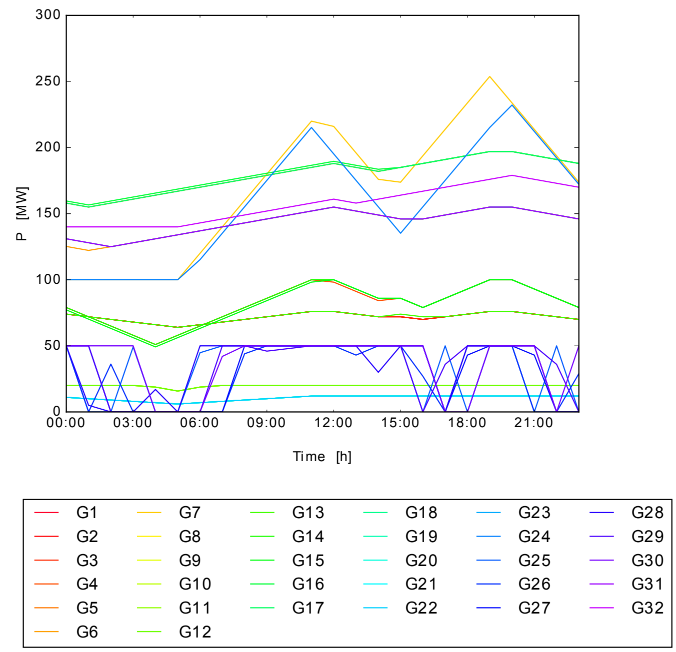


Fig. 5: Generator output power over time

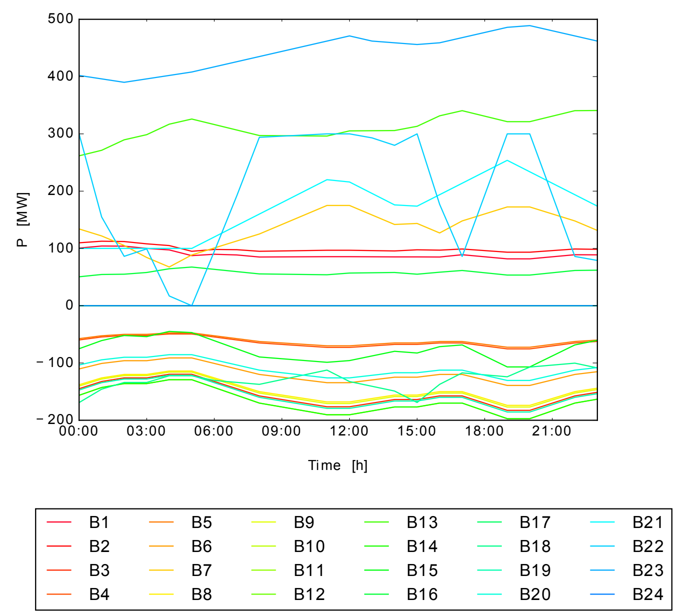


Fig. 6: Bus power flow over time

VI. Conclusion & Future work

While the network-constrained unit commitment model delivers a more completed simulation of the day ahead market, it requires more network and technical generator constraints to know or have an appropriate estimate for those constraints.

For demonstration purposes and for creating a "general feeling" how market clearing algorithms work the basic economic dispatch model is often sufficient and provides a more easily understandable way to learn the underlying mechanisms of market clearing.

While it does not perfectly reflect the real day ahead market clearing algorithm used by today's market operators it is believed that it is often a good approximation.
In order to get a real understanding of the day ahead market clearing the network constrained model should instead be used since it provides a better simulation of the real world electricity grid. The principles behind the network-constrained unit commitment are still in use nowadays in many countries all over the world.
Joint energy and reserve planning using a two-stage stochastic model while harder to understand provides an even more complete picture of market clearing especially regarding the future where more and more renewables are installed and need to be dispatched during the market clearing.
Since greenhouse gases are toxic to the environment and therefore must be limited to use more environmentally friendly options other than the old fashioned coal and gas generators are installed by Transmission Line Operators (TSO's), this global trend towards renewable energy sources is an excellent extension of the web application in the future.
Next to this two-stage stochastic joint energy and reserve planning extension a model for incorporating smart grids could also be integrated with the web application to provide a complete future-proof vision of the electricity grid.

References

[1] A. Elacqua and S. Corey, "Security Constrained Dispatch at the New York Power Pool," pp. 58–59, 1982.
[2] H.-p. Chao and R. Wilson, "Design of Wholesale Electricity Markets."
[3] Python Software Foundation, "Python 3.6.1 Documentation." [Online]. Available: https://docs.python.org/3/
[4] R. Palma-Benhke, A. Philpott, A. Jofré, and M. Cortés-Carmona, "Modelling network constrained economic dispatch problems," *Optim. Eng.*, vol. 14, no. 3, pp. 417–430, 2013.
[5] C. K. Simoglou, P. N. Biskas, and A. G. Bakirtzis, "Optimal self-scheduling of a thermal producer in short-term electricity markets by MILP," *IEEE Trans. Power Syst.*, vol. 25, no. 4, pp. 1965–1977, 2010.
[6] P. Xiong and P. Jirutitijaroen, "A stochastic optimization formulation of unit commitment with reliability constraints," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2200–2208, 2013.
[7] D. Lee, J. Kim, and R. Baldick, "Stochastic optimal control of the storage system to limit ramp rates of wind power output," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2256–2265, 2013.
[8] Gomez-Exposito, *Electric Energy Systems*.
[9] Steven A. Gabriel, Antonio J. Conejo, J. David Fuller, Benjamin F. Hobbs, Carlos Ruiz, S. A. Gabriel, A. J. Conejo, J. D. Fuller, B. F. Hobbs, and C. Ruiz, "Complementarity Modeling in Energy Markets," *Media*, vol. 180, 2013.
[10] E. G. Read and D. Chattopadhyay, "Electricity Market Models : Lessons from Australasian Experience," *Read*.
[11] J. M. Morales, A. J. Conejo, and J. Pérez-Ruiz, "Economic valuation of reserves in power systems with high penetration of wind power," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 900–910, 2009.
[12] L. Puglia, D. Bernardini, and A. Bemporad, "A multi-stage stochastic optimization approach to optimal bidding on energy markets," *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, pp. 1509–1514, 2011.
[13] W. E. Hart, J.-P. Watson, and D. L. Woodruff, "Pyomo: modeling and solving mathematical programs in Python," *Math. Program. Comput.*, vol. 3, no. 3, pp. 219–260, 2011.
[14] W. E. Hart, C. Laird, J.-P. Watson, and D. L. Woodruff, *Pyomo–optimization modeling in python*. Springer Science & Business Media, 2012, vol. 67.
[15] T. Brown, J. Hörsch, and D. Schlachtberger, "Python For Power System Analysis (Pypsa) Version 0.9.0," jan 2017. [Online]. Available: https://zenodo.org/record/582307
[16] Free Software Foundation, "GLPK." [Online]. Available: https://www.gnu.org/software/glpk/
[17] I. Gurobi Optimization, "Gurobi Optimizer Reference Manual," 2016. [Online]. Available: http://www.gurobi.com
[18] "No Title."
[19] J. Forrest, "Welcome to the Cbc home page." [Online]. Available: https://projects.coin-or.org/Cbc
[20] Facebook Inc., "React." [Online]. Available: https://facebook.github.io/react/
[21] Django Software Foundation, "Django." [Online]. Available: https://www.djangoproject.com/
[22] Node.js Foundation, "Node.js." [Online]. Available: https://nodejs.org/en/
[23] C. Grigg and P. Wong, "The IEEE reliability test system -1996 a report prepared by the reliability test system task force of the application of probability methods subcommittee," *IEEE Trans. Power Syst.*, vol. 14, no. 3, pp. 1010–1020, 1999.