SoftwareX 16 (2021) 100870

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Power Market Tool (POMATO) for the analysis of zonal electricity markets

Richard Weinhold [a,*,1], Robert Mieth [b,c,1]

[a] Fakultät VII Wirtschaft und Management, TU Berlin, 10623 Berlin, Germany
[b] Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, New York, NY 10012, USA
[c] Fakultät IV Elektrotechnik und Informatik, TU Berlin, 10587 Berlin, Germany

ARTICLE INFO

Article history:
Received 23 November 2020
Received in revised form 17 June 2021
Accepted 19 October 2021

Keywords:
Flow-based market coupling
FBMC
Economic dispatch problem
Transmission system
Optimal power flow
Security constrained optimal power flow

ABSTRACT

The proposed open-source *Power Market Tool* (POMATO) aims to enable research on interconnected modern and future electricity markets in the context of the physical transmission system and its secure operation. POMATO has been designed to study capacity allocation and congestion management policies of European zonal electricity markets, especially flow-based market coupling (FBMC). For this purpose, POMATO implements methods for the analysis of simultaneous zonal market clearing, nodal (N-k secure) power flow computation for capacity allocation, and multi-stage market clearing with adaptive grid representation and redispatch. The computationally demanding N-k secure power flow is enabled via an efficient constraint reduction algorithm. POMATO provides an integrated environment for data read-in, pre- and post-processing and interactive result visualization. Comprehensive data sets of European electricity systems compiled from Open Power System Data and Matpower Cases are part of the distribution. POMATO is implemented in Python and Julia, leveraging Python's easily maintainable data processing and user interaction features and Julia's well readable algebraic modeling language, superior computational performance and interfaces to open-source and commercial solvers.

Code metadata

Current code version	0.3
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_79
Code Ocean compute capsule	
Legal Code License	LGPL v3
Code versioning system used	git
Software code languages, tools, and services used	Python and Julia
Compilation requirements, operating environments & dependencies	Python and Julia, OS independent (continuous automated tests on Windows and Ubuntu)
If available Link to developer documentation/manual	https://pomato.readthedocs.io/en/latest/
Support email for questions	riw@wip.tu-berlin.de

1. Motivation and significance

Europe's increasing electricity production from renewable energy resources in combination with a significant decline of conventional generation capacity, has spawned political and academic interest in the transmission system's ability to accommodate this transition [1]. Specifically, higher levels of coordination in a pan-European electricity market to facilitate meeting emission targets, while ensuring reliable power supply at acceptable cost [2], require the internalization of the power transmission capabilities into trading processes. Central to this discussion is the efficiency of *capacity allocation* and *congestion management* (CACM) policies between and within electricity market areas that share transmission infrastructure [3]. Capacity allocation (CA) summarizes regulatory and market mechanisms that constrain electricity trading volumes between two adjacent

[*] Corresponding author.
E-mail addresses: riw@wip.tu-berlin.de (Richard Weinhold), robert.mieth@nyu.edu (Robert Mieth).

[1] R. Weinhold and R. Mieth contributed equally to this work.

https://doi.org/10.1016/j.softx.2021.100870
2352-7110/© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

market areas with respect to the expected available cross-border transmission capacity at time of delivery. Congestion management (CM), on the other hand, refers to methods ensuring that the physical system state at time of delivery indeed remains within its security margins, e.g., does not cause transmission line overloads. Noticeably, well defined CA based on suitable forecasts can reduce CM measures, such as out-of-market generator redispatch coordinated by the transmission system operator (TSO). Previously implemented CA policies based on static *net-transfer capacities* (NTCs) consider tie-line capacities between markets, but neglect restricting transmission assets within market zones, thus leading to often non-feasible market outcomes, increased CM (redispatch) or overly conservative market results [4,5]. To overcome these deficiencies and in an effort to "*move towards a genuinely integrated [European] electricity market*" [3], the central-western European (CWE) countries inaugurated *flow-based market coupling* (FBMC) [6], a more complex CA policy that aims to increase the potential volume of cross-border electricity trading while decreasing CM requirements by explicitly accounting for cross-border and zone-internal transmission limits.

The proposed *Power Market Tool* (POMATO) has been designed to enable further research on the status-quo and future policies of practical zonal electricity markets, especially FBMC. While the theory of a centrally coordinated zonal electricity market is somewhat mature, see e.g. [7], practical implementation with imperfect coordination between market and system operators requires ongoing analyses. The current European FBMC is a multi-stage process coordinated by multiple TSOs and involves detailed zone-specific load and generation forecasts and network models, which are typically not, or only partially, disclosed by the TSOs [8]. However, the medium- and long-term evolution of the FBMC design requires public decision support from independent FBMC analyses, that, e.g., study the impact of more countries joining the coupled market or bidding zone layouts (which have been declared inefficient by a recent TSO study [9]). Notably, the yearly federal report on the future of the grid in Germany ("*Netzentwicklungsplan*"), included a rudimentary FBMC representation for the first time in its 2018 edition [10], three years after FBMC implementation.

There are few model implementations of the FBMC process available to the academic community [11–13]. In [11] Aravena et al. compare various CA policies and demonstrate the benefits of FBMC traditional approaches using real-world data sets. Also, this study proposes an iterative approach to meet the practical requirement that any dispatch has to be robust against unplanned transmission equipment outage (*N-1 security*). However, data-sets and model implementation are not published along the paper. Similarly, Matthes et al. [12] present a FBMC formulation for estimating future FBMC parameters and studying the impact of regulations that require additional security margins on critical transmission lines. This model uses an external convex hull reduction for a N-1 secure FBMC solution and is implemented in the MILES framework [14], which, to our knowledge, is also not publicly available. Schönheit et al. [13] extend the classic GAMS[2] model ELMOD [15] to facilitate FBMC analyses and study the impact of regulations requiring a minimum availability of interzonal trading volumes [16]. Here, N-1 security is approximated through static security margins. Despite comprehensive documentation in [13] the code extension itself is undisclosed.

POMATO aims to overcome some of the caveats of [11–13] by providing a documented and open-source framework that is easy to use and goes beyond the implementation of an academic optimization model. Some main features and contributions are:

(a) Separation of data processing (implemented in Python[3]) and optimization (implemented in Julia[4]) to achieve a flexible Python-based user interface that is familiar to many users, while creating a lean implementation of the central optimization model in the well-readable JuMP algebraic modeling language [17].
(b) Open-source available and documented on GitHub[5] [18,19].
(c) Comprehensive data set of the European electricity market and transmission infrastructure based on Open Power Systems Data[6] and Matpower[7] data sets.
(d) Electricity market model with zonal and nodal market clearing and a module to synthesize the FBMC process, including heat sector coupling.
(e) Exact N-1 secure dispatch implementation suitable for large-scale networks and multi-period analyses. The used algorithm removes redundant constraints, similar to the convex hull approach in [12], but has been optimized for optimal power flow (OPF) analyses, yielding improved computation times.
(f) Stochastic OPF using chance-constraints to analyze the impact of forecast errors from renewable energy sources.

2. Software description

POMATO's architecture is structured in three layers as shown in Fig. 1. The *model core* collects the mathematical formulations of the necessary optimization problems and provides an interface to the required solvers. To ensure a lean model implementation and efficient re-runs without re-calculating large parameter sets, the model core is encapsulated in a *data processing* layer. This layer automates parameter calculation and validation, provides parameters to the model core, and validates and processes the resulting model output. Finally, all functionality of POMATO is collected in the outer *user interface* layer via readable API-like commands.

2.1. Model core

The core of POMATO is an electricity market model. This model is a linear optimization problem (LP) that finds the least cost generation schedule that matches system demand with respect to all technical constraints. Eq. (1) shows the general structure of such a model[8]:

$$\min \ \text{OBJ} = \sum COST\_G + COST\_H + COST\_CURT + OOM\_PEN \tag{1a}$$

$$\text{s.t.}$$

$$\text{Cost definition} \tag{1b}$$

$$\text{Generation constraints} \tag{1c}$$

$$\text{Heat constraints} \tag{1d}$$

$$\text{Storage constraints} \tag{1e}$$

$$\text{Energy balances} \tag{1f}$$

$$\text{Network constraints} \tag{1g}$$

[2] www.gams.com.

[3] www.python.org.

[4] www.julialang.org.

[5] www.github.com/richard-weinhold/pomato.

[6] www.open-power-system-data.org/.

[7] www.matpower.org/.

[8] The exact mathematical equations can be found in the POMATO documentation under http://pomato.readthedocs.io/en/latest/model_formulation.html. See also in [8,20].

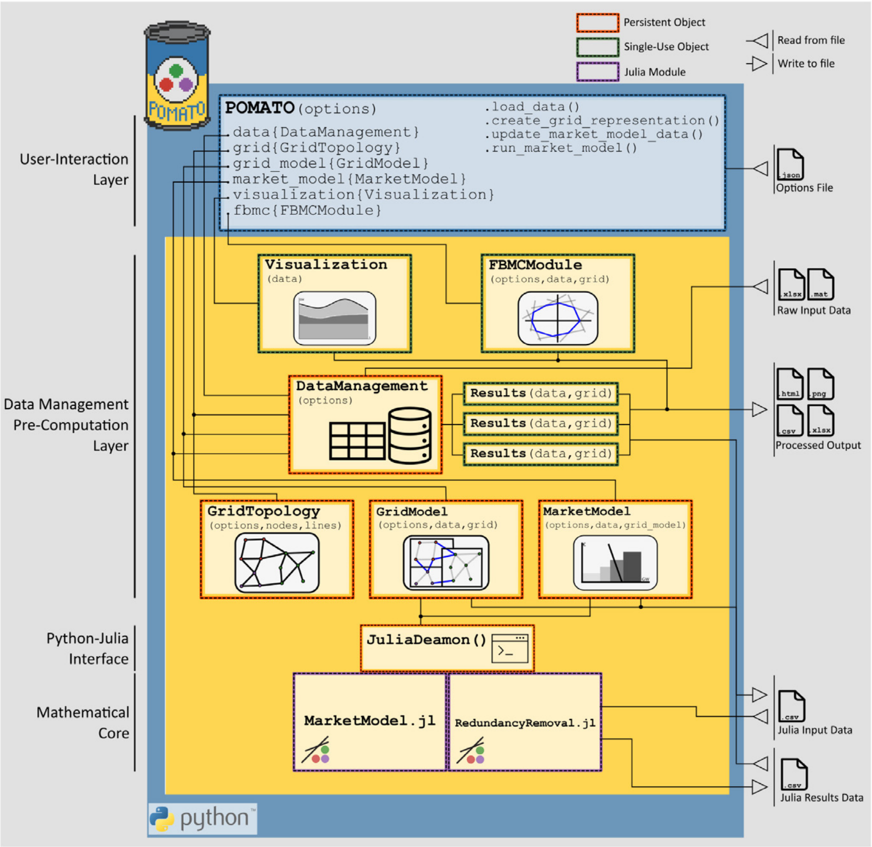


Fig. 1. POMATO's layered architecture. The inner model core is colored purple, the data processing layer is colored yellow, and the outer user interface is colored blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Objective (1a) minimizes the cost of electricity generation ($COST\_G$), heat generation ($COST\_H$), load curtailment ($COST\_CURT$) and out-of market penalties (OOM_PEN, e.g. cost of redispatch) over all units and time steps. Constraints in (1b) define these cost components as sets of linear functions. Depending on the modeled market framework, some cost may be zero. Constraints (1c) and (1d) enforce limits on the available electricity and heat generation, respectively. Constraints (1e) enforce the time-coupled energy constraints for storages and limit their efficiency-corrected charging and discharging power. Eqs. (1f) ensure that for all market areas generation equals demand plus net export (export minus import) at all times. Network constraints (1g) capture the transmission system topology as nodes (buses) and lines and enforce exchange capacity limits. Physical flows are modeled using the linearized (DC) power flow equations so that nodal net injections can be mapped to power flows either directly using power transfer distribution factor (PTDF) matrix or implicitly via auxiliary voltage angle variable, see e.g. [21,22].

The specific definitions of constraints (1b)–(1g) are activated and parametrized by POMATO "on the fly" based on user-defined options. For example, nodal market clearing enforces an energy balance in (1f) for each *node* and exchanges are limited by the physical power flow and the thermal transmission system capacities. Zonal markets, on the other hand, ensure an aggregated energy balance for an entire *zone* (i.e. multiple connected nodes). Exchanges with neighboring zones are limited either explicitly by NTCs, which constrain individual cross-border exchanges, or implicitly through constraints on the net positions (total net export) of each zone. The latter is used to model FBMC by calculating zonal PTDFs, see e.g. [8], that map zonal net positions to power flows on (critical) lines. Additionally, constraints to model CM (redispatch) and external security requirements (contingency robustness) can be included. For redispatch models, a nodal market is solved subject to additional out-of-market penalties based on results of a previously solved zonal market. Specifically, the optimal redispatch is the least-cost change from a given generation schedule with predefined out-of-market penalties and fixed *zonal* balances so that the resulting *nodal* balances are feasible for the given transmission network. Preventive system security requirements based on contingency (N-1) analyses, are enforced through a suitable extensions of zonal and nodal PTDF matrices.

Optimal dispatch with preventive robustness against all contingencies (unplanned outages) is computationally demanding and requires preliminary solution heuristics [21]. Otherwise, because the number of constraints grows exponentially with the network size, typical multi-period dispatch problems for real-world economic analyses cannot be solved in reasonable time or with commonly available computation resources. However, it has been shown that many (in fact most) of these constraints are *redundant*, i.e. never binding in the optimal solution due to the existence of more restrictive constraints [23]. To ensure feasible solution times for real-world networks over longer time horizons (e.g. weeks to years), POMATO's model core includes additional functionality to identify these redundant constraints. Implementation and performance of this redundancy-removal algorithm has been presented in our previous work [21], where we showed that for most of the studied transmission networks over 99% of the constraints can be removed from the optimization problem. This reduced the solve time of smaller problems significantly and enabled the solution of previously infeasible larger models.

The methods and algorithms implemented in the model core require a suitable framework that enables fast robust computations and facilitates the use of interchangeable solver back-ends.

The open-source Julia Language provides a competitive combination of performance and readability [24], as well as suitable libraries for implementing the required mathematical programming (electricity market model and redundancy removal). Here, we opted for the popular, well-readable and flexible JuMP package [17]. POMATO's Julia modules (`MarketModel.jl` and `RedundancyRemoval.jl`) are parametrized and called automatically by the higher POMATO layers, as indicated in Fig. 1.

Feature Remark 1 (*Risk-Aware Economic Dispatch*)**.** When modeling systems with high shares of stochastic renewable generation, feed-in uncertainty with respect to capacity reserve, network constraints and risk-perception of the system operator can be internalized in the marker clearing using *chance constraints* [25, 26]. This capability is released with POMATO as an experimental feature. See [20] for additional information.

Feature Remark 2 (*Solvers*)**.** JuMP allows to easily communicate with a wide range of solvers. Per default, POMATO will install and use the open CLP solver.[9] However, if available on the user's system, POMATO can be configured to use commercial solvers like Mosek[10] or Gurobi.[11] Other solvers can be implemented as outlined in the documentation.[12]

2.2. Data management and pre-processing

POMATO's central Data-Management layer, see Fig. 1, manages the functionality of the model core by providing data and model parametrizations, calling and monitoring the Julia processes, and validating their results. To enable flexible data-handling that avoids redundant calculations, we leverage Python and its object-oriented programming paradigm, which resolves issues typically related to static scripts. In such linear approaches, e.g. in GAMS, all parameters are handled in the global scope of the program so that any desired conditional functionality renders the code overly complex and hard to maintain. Alternatively, encapsulating the required functionally in different objects (modules), ensures data consistency and compatible methods.

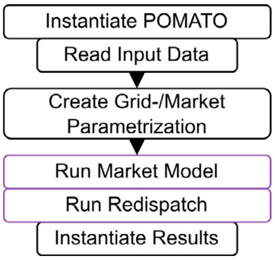
The data-management layer consists of four main modules, `DataManagement`, `GridTopology`, `GridModel` and `MarketModel`. These modules are created once in every instance of POMATO and each provides specific methods to the user-interface and other modules. Additionally, single-use modules are used for result processing, e.g. FBMC calculations and result visualization. `DataManagement` is the central module to handle all data and their corresponding options. All input data is validated with respect to predefined data structures that are required to run the desired market model and are made available to other modules. Due to the persistence of `DataManagement`, changes in data, e.g. from user manipulations or model results, are propagated throughout POMATO.

Pre-processing and validation of data related to the transmission network topology requires specific methods that are collected in the `GridTopology` module. It uses network data (bus/node and line/branch information) and provides the parameters for subsequent power flow calculations. Additionally, `GridTopology` verifies the properties of the network graph, sets reference nodes for multiple disconnected networks, creates contingency scenarios and manages settings of phase shifting transformers.

The verified and pre-processed data is then used by the `GridModel` and `MarketModel` modules to provide the input data for the model core. Based on the market model that is desired from the user (and that is feasible given the available data), `GridModel` uses the data provided from `GridTopology` and `DataManagement` to compile the correct grid representation with all physical properties. This includes the parametrization of the network constraints, as discussed in Section 2.1 above, as well as generator locations, cost models, and zonal configurations. Also, `GridModel` calls `RedundancyRemoval` in the process of compiling the set of parameters for a N-1 secure dispatch. If this algorithm has been performed in previous runs, `GridModel` loads available parameters to avoid time-consuming re-runs. These parameters define the constraints under which the market has to be cleared. Finally, `MarketModel` collects these constraints and manages the required model runs. After the optimization in the model core has obtained an optimal solution, `MarketModel` instantiates a new `Results` object in the `DataManagement` module. Communication between the POMATO instance and the mathematical core is achieved via a threaded process within the `JuliaDaemon` module, see Remark 4.

The decoupling of model runs and data processing allows to efficiently solve multi-stage market clearing processes that iterate between different market model configurations. Two relevant examples are itemized below.

Ex. 1 - *Redispatch Analysis:* Zonal electricity markets are cleared as a single price zone without any internal network constraints. In a second stage, potential transmission line overloads are rectified by out-of-market CM measures, i.e., redispatch. For this purpose, a second model run with nodal resolution is required.



Ex. 2 - *FBMC:* FBMC uses a forecasted nodal dispatch (base case) to derive zonal PTDF matrices that establish the relation between zonal net-positions and flows on critical network elements (CNEs). The resulting feasible region of net-positions with respect to the technical limits of the CNEs are called *FBMC domains*. These domains are used to solve the zonal market clearing that is then corrected by subsequent CM

[9] www.github.com/jump-dev/Clp.jl.
[10] www.github.com/MOSEK/Mosek.jl.
[11] www.github.com/jump-dev/Gurobi.jl.
[12] http://pomato.readthedocs.io/en/lastest/installation.html#solvers.

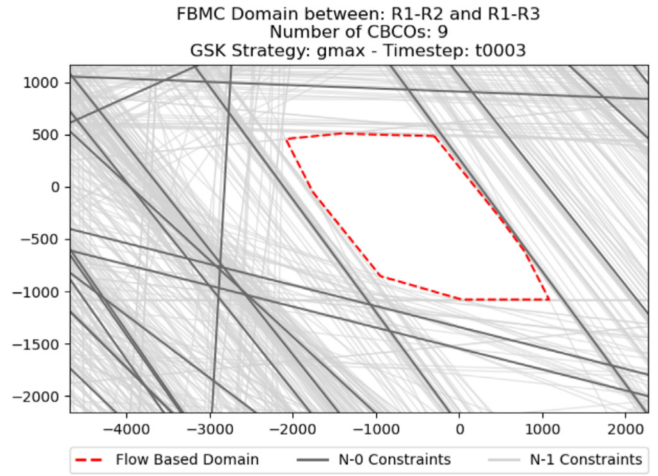
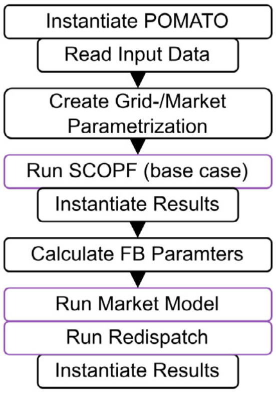


Fig. 2. Exemplary flow based domain.

measures. See also [20] for more information the FBMC process. The supplementary `FBMCModule` automatizes the necessary re-parametrization of parts of the model data, including the generation of the *FBMC domains* depicted in Fig. 2, and facilitates the necessary re-runs of the optimization model.



Feature Remark 3 (*Contingency Scenarios*)**.** The user can specify customized contingency scenarios instead of performing a strict N-1 contingency analysis (i.e., protecting the system against the outage of every single transmission line). Such scenarios can either ignore certain lines as potential contingencies or define contingency groups of outages that are likely to occur simultaneously, e.g., parallel lines.

Feature Remark 4 (*Python–Julia Interface*)**.** Communication between the Julia language and Python is implemented by a `JuliaDaemon` class that maintains a threaded Julia program and executes its functions when requested by `MarketModel` or `GridModel`. Instructions are transferred via *json* files that are written to the POMATO package folder. This way, the Julia process has to be started only once and remains alive after a model is completed, which allows fast re-solving of a model with different parameters.

2.3. User interface

All modules of the data processing layer presented in previous Section 2.2 are attributed to the overall POMATO object, which provides a comprehensive set of functions to the user. Although the methods of the lower level modules are accessible to the user, the POMATO module provides predefined methods for the typical use cases of this program. A brief overview is presented in the illustrative application in Section 3. Additionally, POMATO provides a collection of visualization features. Its `create_geo_plot` method shows market and redispatch results including prices on an interactive map using the *Ploty*[13] library. Many results, such as the FBMC domains or generation schedules, can be automatically plotted as a collection of interactive 2D diagrams. In addition, POMATO can start a *Dash*[14] server, featuring a curated view of interactive plots and visualizations. See documentation[15]

2.4. Publication

POMATO is available under the *LGPLv3* license on GitHub including a comprehensive documentation [18,19]. To facilitate testing, maintenance and readability, POMATO's Python and Julia components are stored in separate repositories. However, all installation is handled by the central Python module in [18]. See also Remark 5. The repository also includes exemplary data-sets that cover the core functionality. These include a data-set for the German power system, that has been compiled to showcase POMATO's functionality, as well as the NREL 118 bus network [27], and Matpower case files. More details on how to create custom data-sets can be found in the accompanying documentation.[16]

Feature Remark 5 (*Installation*)**.** POMATO requires Python and Julia to be installed. When both are available on the user's system the installation of POMATO is managed through Python via pip.[17] The Python installer will detect the Julia distribution on the system and automatically pull the two Julia packages, `MarketModel` and `RedundancyRemoval` from their separate repositories. They will be installed in a Julia virtual environment to isolate any dependencies from possible user-installed Julia packages.

3. Illustrative application

This section illustrates the application of POMATO. The corresponding example code is shown below and its line numbers are referenced in the subsequent description.

[13] www.plotly.com.

[14] www.plotly.com/dash/.

[15] http://pomato.readthedocs.io/en/latest/model_functionality.html#the-dashboard.

[16] http://pomato.readthedocs.io/en/latest/input_model_data.html.

[17] www.pip.pypa.io.

```
 1 from pomato import POMATO
 2
 3 mato = POMATO(
 4     wdir=wdir,
 5     options_file="options/de.json"
 6 )
 7 mato.load_data(’data_input/DE_2020.zip’)
 8
 9 mato.create_grid_representation()
10 mato.run_market_model()
11
12 # Return market results (m)
13 # and redispatch results (r)
14 m, r = mato.data.return_results()
15
16 # Check for Overloaded lines N-0
17 n0_m, _ = m.overloaded_lines_n_0()
18 print("Number of N-0 Overloads: ",
19       len(n0_m))
20 n0_r, _  = r.overloaded_lines_n_0()
21 print("Number of N-0 Overloads: ",
22       len(n0_r))
23 n1_r, _  = r.overloaded_lines_n_1()
24 print("Number of N-1 Overloads: ",
25       len(n1_r))
26
27 gen = r.redispatch()
28 print("Total Redispatch in MWh: ",
29       gen.delta_abs. sum())
30
31 mato.visualization.create_geo_plot(
32     redispatch_result,
33     show_prices=True,
34     show_redispatch=True)
```

After importing the POMATO module in line 1, a new POMATO class is instantiated in line 3 with the requested working directory (`wdir`) and an options file in the JSON[18] format that configures the model run. In this example the options file specifies a zonal market clearing, subject to NTCs, and subsequent redispatch with redispatch-cost and model horizon:

```
{"optimization": {
        "type": "ntc",
        "model_horizon": [0, 168],
        "redispatch": {
                "include": true,
                "cost": 50}}
```

A comprehensive list and descriptions of possible configurations can be found in the online documentation.[19] In line 4 the POMATO object loads the required data, which automatically instantiates the data management objects as described in Section 2.2 above. The command in line 6 creates the grid representation for the specified model run and line 7 starts the actual model run. After successful completion, two `Results` are instantiated into `DataManagement` and are available to the user (see line 9) for further analysis, e.g., to check for overloaded lines with and without contingencies (see lines 11–20). Finally a geo-plot is created in line 22, which is shown in Fig. 3.

Execution time of this example is around 7.5 min, including data read-in, Julia start-up, solving both the market- and the redispatch problem, as well as the final result analysis and plotting. Subsequent runs benefit from already active Julia processes, which saves 2 to 4 min of computation time.

4. Impact

POMATO offers features that may be of interest for practitioners, researchers and students alike, all of which benefit from POMATO's open-source availability.

- The high-level API-like functionality of the top-layer POMATO class (see Section 2.3) facilitates the use of POMATO without a detailed understanding of power engineering and economics. With minimal understanding of the required data structure, POMATO offers a low barrier for users to obtain relevant results from their own data sets and to produce insightful visualizations of these results.
- By separating the model core from the user interface, more advanced users will be able to extend or modify the mathematical formulations for their needs, which we hope to enable new research in the area of zonal market clearing.
- Advanced interactive visualizations can support educators to convey some concepts of power system operations and electricity markets.

5. Conclusion and further development

This paper presents a tool that aims to enable the broad analyses of zonal electricity markets. The proposed Power Market Tool POMATO is open-source, provides essential data-sets and comprehensive online documentation. Central to POMATO's functionality is its ability to synthesize the flow-based market-coupling (FBMC) process. Further, POMATO includes a redundancy removal algorithm that enables security constrained (N-k) dispatch with feasible computation times. While the current version of POMATO already includes prototypical options for the analyses of stochastic (chance-constrained) market clearing, extensions to stochastic and risk-aware markets and power flow analyses along the lines of [25,26] are subject to ongoing development.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) in the project LKD-EU (03ET4028A) and European Union's Horizon 2020 research program (No. 773406). The work of Robert Mieth was supported by the Reiner Lemoine-Foundation. The work of Richard Weinhold was supported by the Danish Energy Agency.

[18] www.json.org.

[19] http://pomato.readthedocs.io/en/latest/options.html.

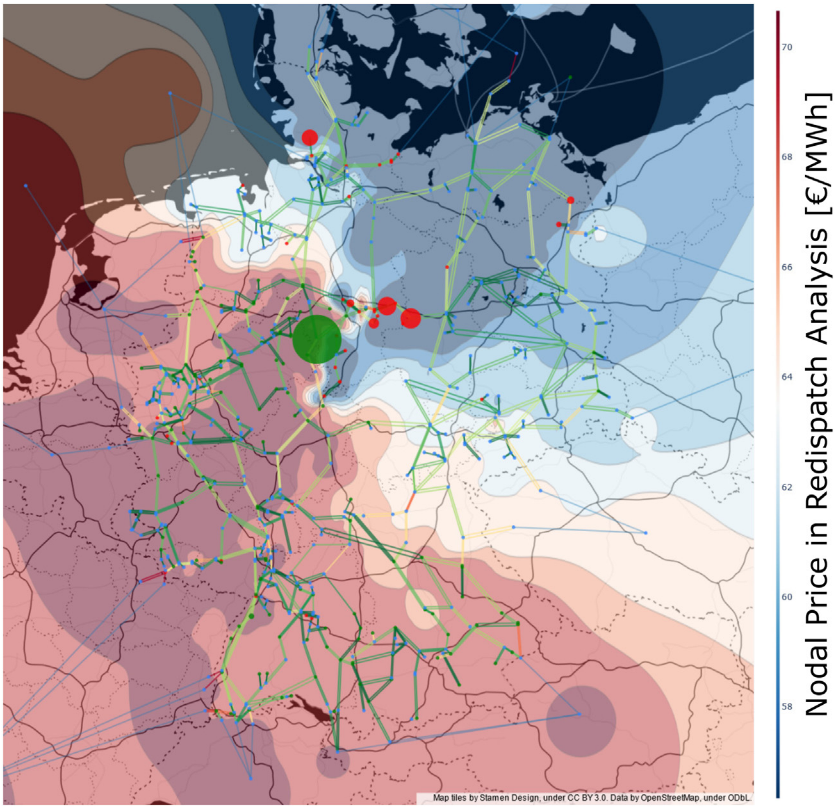

70
68
66
64
62
60
58
Nodal Price in Redispatch Analysis [€/MWh]
Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

Fig. 3. Geo-plot of market- and redispatch result for Germany. Red and green circles indicate negative and positive changes in generation schedules, respectively, due to redispatch. Change magnitude are proportional to the circles' radius. Line colors indicate average line load where green is low loads and red is high load. Prices as dual multipliers on the nodal power balances of the redispatch run are indicated by the contour plot overlay. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

References

[1] Amprion. Flow based market coupling – Development of the market and grid situation 2015–2017. 2018, amprion.net/Dokumente/Dialog/Downloads/Studien/CWE/CWE-Studie_englisch.pdf.

[2] European Parliament. Internal energy market. 2020, www.europarl.europa.eu/factsheets/en/sheet/45/internal-energy-market.

[3] The European Commission. Commission regulation (EU) no 2015/1222. 2015, eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32015R1222&from=EN.

[4] Amprion, APX-ENDEX, Belpex, Creos, Elia, EnBW, EPEX SPOT, RTE, TenneT. CWE Enhanced flow-based MC feasibility report. 2011, www.amprion.net/Dokumente/Strommarkt/Engpassmanagement/CWE-Market-Coupling/CWE_FB-MC_feasibility_report.pdf.

[5] The European Commission. Commission regulation (EU) 2016/1719 of 26 september 2016 establishing a guideline on forward capacity allocation. Off J Eur Union 2016. September 26.

[6] Van den Bergh K, Boury J, Delarue E. The flow-based market coupling in Central Western Europe: Concepts and definitions. Electr J 2016;29(1):24–9.

[7] Ehrenmann A, Smeers Y. Inefficiencies in European congestion management proposals. Util Policy 2005;13(2):135–52.

[8] Schönheit D, Weinhold R, Dierstein C. The impact of different strategies for generation shift keys (GSKs) on the flow-based market coupling domain: A model-based analysis of Central Western Europe. Appl Energy 2020;258:114067.

[9] ENTSO-E. First edition of the bidding zone review – Final report. 2018, eepublicdownloads.azureedge.net/clean-documents/news/bz-review/2018-03_First_Edition_of_the_Bidding_Zone_Review.pdf.

[10] Genehmigung des Szenariorahmens für die Netzentwicklungsplanung. Netzentwicklungsplanung 2019-2030, Bonn: Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen (BNetzA). 2018.

[11] Aravena I, Papavasiliou A. Renewable energy integration in zonal markets. IEEE Trans Power Syst 2016;32(2):1334–49.

[12] Matthes B, Spieker C, Klein D, Rehtanz C. Impact of a minimum remaining available margin adjustment in flow-based market coupling. In: 2019 IEEE milan powertech. IEEE; 2019, p. 1–6.

[13] Schönheit D, Hladik D, Hobbie H, Möst D. ELMOD documentation: Modeling of flow-based market coupling and congestion management, Working paper of the Chair of Energy Economics (TU Dresden).

[14] Spieker C, Klein D, Liebenau V, Teuwsen J, Rehtanz C. European electricity market and network simulation for energy system analysis. In: 2016 IEEE international energy conference. IEEE; 2016, p. 1–6.

[15] Leuthold FU, Weigt H, von Hirschhausen C. A large-scale spatial optimization model of the European electricity market. Netw Spat Econ 2012;12(1):75–107.

[16] Schönheit D, Dierstein C, Möst D. Do minimum trading capacities for the cross-zonal exchange of electricity lead to welfare losses? Energy Policy 2020.

[17] Dunning I, Huchette J, Lubin M. JuMP: A modeling language for mathematical optimization. SIAM Rev 2017;59(2):295–320.

[18] Pomato - GitHub repository. 2020, github.com/richard-weinhold/pomato.

[19] Pomato - Documentation. 2020, pomato.readthedocs.io/en/latest.

[20] Weinhold R, Mieth R. Uncertainty-Aware Capacity Allocation in Flow-Based Market Coupling. 2021, arXiv preprint arXiv:2109.04968.

[21] Weinhold R, Mieth R. Fast security-constrained optimal power flow through low-impact and redundancy screening. IEEE Trans Power Syst 2020;35(6):4574–84.

[22] Hörsch J, Ronellenfitsch H, Witthaut D, Brown T. Linear optimal power flow using cycle flows. Electr Power Syst Res 2018;158:126–35.

[23] Bouffard F, Galiana FD, Arroyo JM. Umbrella contingencies in security-constrained optimal power flow. In: 15th power systems computation conference, vol. 5. 2005.

[24] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. SIAM Rev 2017;59(1):65–98.

[25] Dvorkin Y. A chance-constrained stochastic electricity market. IEEE Trans Power Syst 2019.

[26] Mieth R, Kim J, Dvorkin Y. Risk-and variance-aware electricity pricing. Electr Power Syst Res 2020;189:106804.

[27] Pena I, Martinez-Anido CB, Hodge B-M. An extended IEEE 118-bus test system with high renewable penetration. IEEE Trans Power Syst 2017;33(1):281–9.