Capturing Electricity Market Dynamics in the Optimal Trading of Strategic Agents using Neural Network Constrained Optimization

Mihaly Dolanyi
ESIM, KU Leuven

Kenneth Bruninx
ESIM, KU Leuven

Jean-François Toubeau
PSMR, University of Mons

Erik Delarue
ESIM, KU Leuven

Abstract

In competitive electricity markets the optimal trading problem of an electricity market agent is commonly formulated as a bi-level program, and solved as mathematical program with equilibrium constraints (MPEC). In this paper, an alternative paradigm, labeled as mathematical program with neural network constraint (MPNNC), is developed to incorporate complex market dynamics in the optimal bidding strategy. This method uses input-convex neural networks (ICNNs) to represent the mapping between the upper-level (agent) decisions and the lower-level (market) outcomes, i.e., to replace the lower-level problem by a neural network. In a comparative analysis, the optimal bidding problem of a load agent is formulated via the proposed MPNNC and via the classical bi-level programming method, and compared against each other.

1 Introduction and Background

Electricity markets, the cornerstone of energy transition, are governed by systematic principles such as technical constraints and economic objectives. Consequently market actors and policy makers are interested in defining a mathematical model that well-captures these market dynamics inside their reward (profit, or social welfare) maximization problem. In power systems, this goal is mostly achieved by modeling a Stackelberg-game [1] via a bi-level program, which can be further translated into a single-level mathematical program with equilibrium constraints (MPEC) [2, 3]. This modeling strategy poses many numerical challenges due to the inherent non-convex and nonlinear nature of the market clearing (lower-level). Additionally, the lower-level parameters are usually partially known for the modeler, such as the constraints of the market agents or the "true" (often imperfect) objective function of the participants.

Reinforcement learning (RL) may be seen as an alternative to the model-based MPEC approach, leading to a fully data-driven solution method wherein the optimal policy is directly learned from interactions with the physical environment, as utilized in the works of [4–6]. The RL theory assumes that the system dynamics do not depend explicitly on time [7]. In a market environment wherein market dynamics are quickly varying this may jeopardize the speed and quality of the learning process. Moreover, most RL algorithms consider that the state of the system is fully observable. In reality, a market agent ignores the operating conditions of its competitors, and must thus cope with a partially observable system state, which complicates the problem.

Motivated by the above challenges in existing model-based and model-free optimal bidding strategies, this paper proposes a hybrid modeling strategy that combines the strengths of both approaches by

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2021.

recasting the problem as a Mathematical Program with Neural Network Constraint (MPNNC). In power systems research, solution strategies combining optimization methods with neural networks (NNs) have been only formulated to capture the feasible space of non-convex optimal power flow problems [8–11], thus reducing the computational burden. In this work, we instead aim to leverage the learning abilities of neural networks to propose a data-driven representation of the (imperfectly known) LL problem in a bi-level structure. In order to enhance the solution quality of the proposed MPNNC, we employ input convex neural networks (ICNNs) [12], in a similar fashion as they were exploited for optimal control applications [13–15]. The results of the MPNNC models are benchmarked against the classical MPEC approach in an ideal simulation environment, where the MPEC can explicitly capture the market dynamics.

2 Methodology

2.1 The chained input convex NN for capturing the electricity market (represented as an economics dispatch problem)

Let us assume that the electricity market clearing, formulated as an optimization problem subject to a set of constraints, characterizes (i) the mapping between its inputs x (e.g. the participant's quantity bids: $\{\overline{G_i} \cdot AF_{i,t}, D_t\}$, and price bids: c_i^{gen}) and the resulting dispatch ($y^{DP} = g_{i,t}$) of the generation $f^{ED} : x \mapsto y^{DP}$, (ii) the mapping between the resulting dispatch and corresponding price (λ), which is defined by the merit order curve $f^{MO} : y^{DP} \mapsto \lambda$. After connecting the two functions, the relationship between the price formation of the electricity market (EM) and its inputs may be captured as: $f^{EM} : x \mapsto \lambda$. The detailed ED problem, describing the simulated environment, is presented in Appendix A. Since the electricity market price formation may be seen as a composition of two sub-problems, the proposed NN model also contains two chained learning models.

$NN(x, \theta)$ denotes the neural network, where θ refers to the weights W and biases b parameters, and x to the vector of input features. The optimal weights and biases are obtained in the ex-ante training that minimizes the mean-squared-error (MSE) loss function: $\theta^* = \arg\min_y ||\hat{y} - y||_2^2$. The training is split into two parts, (i) fitting the dispatch forecast $y^{ED} = NN^{ED}(\theta^{ED}, x)$, (ii) fitting the price forecast given the results of of the forecasted dispatch $\lambda = NN^{EM}(\theta^{EM}, y^{ED})$. The structural scheme of the two connected networks and the optimization of inputs is shown in Figure 1. The resulting chained model is fitted to forecasts the prices (the output of the lower-level model) the following way: $\theta^* = \arg\min_\lambda ||\big(\arg\min_{y^{DP}} ||\hat{y}^{DP} - y^{DP}||_2^2\big) - \hat{\lambda}||_2^2$. By decoupling the forecaster NN into two sub-models (dispatch and price formation), the true electricity market mechanisms are better reflected, and thus the behaviour of the forecaster ($NN(x, \Theta)$) can be better interpreted. Moreover, the intermediate dispatch variables can be constrained via embedding generator constraints into the first NN, in a similar fashion as in [16], to enhance both computational performance and credibility of the LL model. In order to improve the solution quality of the MPNNC model in the subsequent optimization step, we employ the input convex NNs for both NNs, as introduced in [12], and applied for OPF optimization in [11].

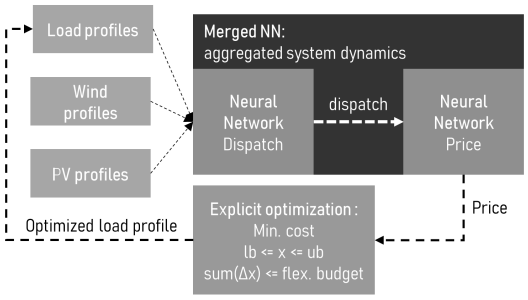

Fig. 1: Scheme showing the generic setting for the NN-based optimization.

2.2 Mathematical program with neural network constraints for the optimal bidding problem of the demand agent

In the bi-level setting, the decision vector of the UL agent D_t, parametrizes the decision space, $y \in Y$, of the LL economic dispatch problem (depicted in Appendix A). The demand agent's optimization problem (the UL problem) is defined below by Eq. (1-4). Objective function (1) is the summed cost of the load agent for procuring electricity, i.e., the market price λ_t multiplied by the requested electricity D_t. Note that the market price depends on the inputs (x), including the decided quantity bids (D_t) of the UL agent. Therefore, including information about how D_t affects λ_t is beneficial for the UL agent. Eq. (3) enforces that the chosen demand bid does not exceed a predefined maximum ($\overline{D_t}$), or be negative. Eq. (4), enforces a budget constraint on the aggregated flexibility activation by imposing that 5% of the overall demand can be shed throughout the day.

$$\text{MPNNC}: \min_{D_t} \mathcal{C}(D_t, \lambda_t(x)) = \mathcal{C}(D_t, \lambda_t(x)) \tag{1}$$

subject to

$$\mathcal{C}(D_t, \lambda_t(x)) = \lambda_t(x) \cdot D_t \tag{2}$$

$$0 \leq D_t \leq \overline{D_t} \tag{3}$$

$$\sum_{t \in T} (\overline{D_t} - D_t) \leq \sum_{t \in T} 0.05 \cdot \overline{D_t} \tag{4}$$

$$\lambda_t(x) = NN(x, \theta) \tag{5}$$

The above proposed MPNNC model consists the optimization problem of the UL agent constrained by the chained $NN(x, \theta)$ in LL (Eq. 5) that captures the mapping between input variables ($D_t \subseteq x$) and the resulting price (λ_t). Contrary, the classical MPEC model (see in Appendix C) has the set of KKT conditions (Appendix B) in the LL level.

Compared to the training step, in the MPNNC problem the weights (θ) are fixed in $NN(x, \theta)$, whereas a subset of input variables, i.e., the chosen demand bids are optimized. Given the significantly reduced number of decision variables compared to the training phase, the Sequential Least Squares Programming (SLSQP) algorithm is used, which is a second-order (quasi-Newton) optimization method. It captures curvature information that can lead to quick convergence and high robustness [17], while reducing the number of manually tuned hyper parameters [18].

3 Solution procedure and Results

3.1 Training the Neural Network model

The training and test results were generated by altering the inputs of the ED model, introduced in Section A. The CO_2 price (p^{CO_2}) is set to 25 €/t, while the Value of Lost Load (VoLL) is 1000 €/MWh. The inputs time series of the ED model, namely the load profile and the wind and solar availability factors are multiplied at each time-step of the day by a normally distributed coefficient ($\mu = 1.0, \sigma = 0.05$). This way, 500 altered dispatch profiles, and market price profiles, are generated. The first 400 are used for training the NN and the last 100 to perform the optimization with both MPEC and MPNNC models.

The NN weights are optimized by a SGD algorithm using the Adam optimizer [19] to result in the lowest mean squared error (MSE) w.r.t. the output of the ED model. Each day is handled as an independent batch, leading to overall 400 batches of size 24. The number of epochs is chosen as 50, whereas the learning rate is 0.001, for both NNs. Each NN is composed of three layers, from which the first one is a dense layer and the other two others are input convex layers. All hidden layers have 64 neurons. The NN models are implemented in the Julia programming language [20] using the Flux package [21]. To interoperate with Python packages, the PyCall package is used.

3.1.1 Learning the economic dispatch model

The first NN is trained to best predict the dispatch profile of the 16 generators, including 14 conventional unit (summarized in Table 1 of Appendix A) and 2 renewable generators. The foretasted dispatch for the 100th test day (last in the test set), along with the overall (unaltered) system load

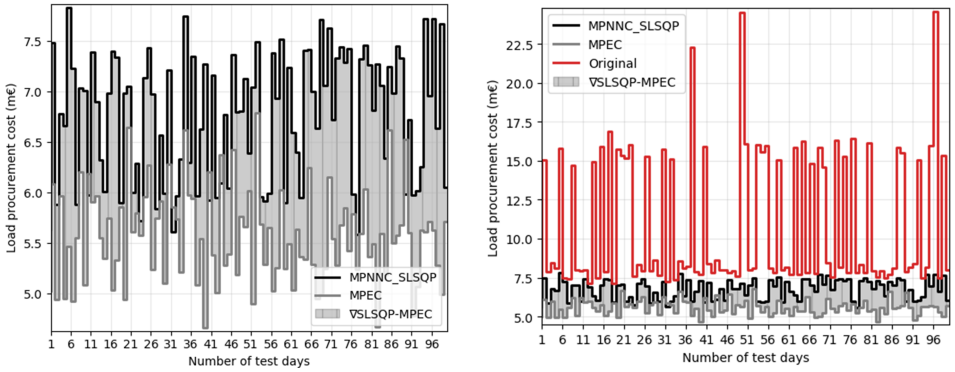

Fig. 2: Comparing the daily cost reductions of both models (MPEC, MPNNC) to the original costs (right figure). For better illustration, the left figure, only depicts the results of the two optimized costs.)

is shown in Fig. 3 in Appendix D. In order to maintain system balance the stacked dispatch needs to be equal to the load. While the error for the individual generator's predicted dispatch may differ, especially for the ones with smaller capacity as their contribution to the overall MSE is smaller, the aggregated predicted generation follows closely the load curve. This suggests sufficient accuracy of the intermediate dispatch forecast. The small error in the predicted generation leads to acceptable levels of final price predictions, discussed in Appendix E.

3.2 Optimal bidding

In this Section, we compare the results of the proposed MPNNC model to the classical MPEC, reformulated as MINLP. Opposed to real electricity markets, for the stylized simulation setting, the MPEC's lower-level problem representation is exact and the KKT conditions are meaningful, thus can be used as a performance benchmark. The MINLP is solved by the Gurobi solver [22], using the non-convex optimization method that guarantees global optimality. However, due to the (i) optimistic LL response assumed in the MPEC, and (ii) the possibly erroneous initialization of M values, the MINLP model still might not result in global optimum in some cases. The MPNNC model is solved via using the optimizer of the *scypy* library [23], selecting the SLSQP algorithm, introduced in [24]. The tolerance in the MPNNC models is set to the optimality gap of Gurobi: 0.0001.

Table 2 in Appendix F summarizes the average solution time to solve the MPNNC and MPEC models and the summarized daily cost attained by the two models over the 100 test days. The final costs are obtained by integrating the load's quantity bids (i.e., decisions of the optimization) in the ED model. This process eliminates the inherent biases of the MPNNC and MPEC models, e.g., the optimistic LL response in the MPEC model. The costs of procuring the demand are substantially reduced by both models compared to the original (no flexibility activation) case. The strategy obtained by the MPNNC to procure the load leads to costs around 18 % higher than the MPEC model. Although, the difference may seem substantial the MPEC can achieve this performance only in the stylized setting. In addition, there is a clear advantage of the MPNNC model's solution time.

Fig. 2 shows the evolution of daily costs over the test set (100days). The right hand side shows the optimized costs (by both MPNNC and MPEC models) in comparison with the original outcomes. Similarly as in the summarized results of Table 2, the improvement attained by both models is significant and their differences are not substantial compared to the overall decrease of the cost. Despite the overall better solutions of the MPEC model, the MPNNC attains lower costs 5 % of the time. The best performance of the MPNNC model is achieved on day 90, with an improvement of 7.27e5 € w.r.t. the MPEC, and the worst outcome occurred on day 65, with total costs 1.49e6 € higher than the MPEC. For these extreme days, the optimized bids and resulting market price profiles are discussed in Appendix F.

4 Conclusion

In this paper, a learning-based MPNNC model was presented to formulate the optimal bidding problem of a strategic load agent. Compared to the classical MPEC formulation, the presented approach has the flexibility to capture real electricity market behaviour in the lower-level, while the upper-level allows for including various types of constraints and objective functions. In out numerical analysis, the MPNNC model gets close to the theoretically optimal MPEC, and their difference is limited compared to the overall improvement in the objective value. As future work, the MPNNC model should be tested on various less stylized, non-convex, market environments, to support the proposed method's real-life applicability.

References

[1] H. Von Stackelberg, *Market structure and equilibrium*. Springer Science & Business Media, 2010.

[2] S. A. Gabriel, A. J. Conejo, J. D. Fuller, B. F. Hobbs, and C. Ruiz, *Complementarity modeling in energy markets*. Springer Science & Business Media, 2012, vol. 180.

[3] A. J. Conejo, L. Baringo, S. J. Kazempour, and A. S. Siddiqui, "Investment in electricity generation and transmission," *Cham Zug, Switzerland: Springer International Publishing*, vol. 106, 2016.

[4] I. Boukas, D. Ernst, T. Théate, A. Bolland, A. Huynen, M. Buchwald, C. Wynants, and B. Cornélusse, "A deep reinforcement learning framework for continuous intraday market bidding," *Machine Learning*, pp. 1–53, 2021.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[6] G. Bertrand and A. Papavasiliou, "Adaptive trading in continuous intraday electricity markets for a storage unit," *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 2339–2350, 2019.

[7] D. Ernst, M. Glavic, and L. Wehenkel, "Power systems stability control: reinforcement learning framework," *IEEE transactions on power systems*, vol. 19, no. 1, pp. 427–435, 2004.

[8] X. Pan, M. Chen, T. Zhao, and S. H. Low, "Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems," *arXiv preprint arXiv:2007.01002*, 2020.

[9] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 630–637.

[10] A. Venzke, D. T. Viola, J. Mermet-Guyennet, G. S. Misyris, and S. Chatzivasileiadis, "Neural networks for encoding dynamic security-constrained optimal power flow to mixed-integer linear programs," *arXiv preprint arXiv:2003.07939*, 2020.

[11] Y. Chen, Y. Shi, and B. Zhang, "Data-driven optimal voltage regulation using input convex neural networks," *Electric Power Systems Research*, vol. 189, p. 106741, 2020.

[12] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 146–155.

[13] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7559–7566.

[14] Y. Chen, Y. Shi, and B. Zhang, "Modeling and optimization of complex building energy systems with deep neural networks," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2017, pp. 1368–1373.

[15] ——, "Optimal control via neural networks: A convex approach," *arXiv preprint arXiv:1805.11835*, 2018.

[16] B. Chen, P. Donti, K. Baker, J. Z. Kolter, and M. Berges, "Enforcing policy feasibility constraints through differentiable projection for energy optimization," *arXiv preprint arXiv:2105.08881*, 2021.

[17] J. Martens, *Second-order optimization for neural networks*. University of Toronto (Canada), 2016.

[18] P. Xu, F. Roosta, and M. W. Mahoney, "Second-order optimization for non-convex machine learning: An empirical study," in *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 2020, pp. 199–207.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, "Julia: A fast dynamic language for technical computing," *arXiv preprint arXiv:1209.5145*, 2012.

[21] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, and V. Shah, "Fashionable modelling with flux," *CoRR*, vol. abs/1811.01457, 2018. [Online]. Available: https://arxiv.org/abs/1811.01457

[22] Gurobi Optimization, LLC, "Gurobi Optimizer," 2021. [Online]. Available: https://www.gurobi.com

[23] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[24] D. Kraft *et al.*, "A software package for sequential quadratic programming," 1988.

[25] J. Lago, F. De Ridder, and B. De Schutter, "Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms," *Applied Energy*, vol. 221, pp. 386–405, 2018.

[26] D. S. Kirschen and G. Strbac, *Fundamentals of power system economics*. John Wiley & Sons, 2018.

[27] M. Dolányi, K. Bruninx, and E. Delarue, "Exposing the variety of equilibria in oligopolistic electricity markets with strategic storage," in *INFORMS 2020, Location: Online*, 2020.

[28] J. Fortuny-Amat and B. McCarl, "A representation and economic interpretation of a two-level programming problem," *Journal of the Operational Research Society*, vol. 32, no. 9, pp. 783–792, 1981.

[29] T. Kleinert, M. Labbé, F. a. Plein, and M. Schmidt, "There's no free lunch: on the hardness of choosing a correct big-m in bilevel optimization," *Operations research*, vol. 68, no. 6, pp. 1716–1721, 2020.

A The ED problem describing the LL

Nomenclature

Parameters

$\overline{AF_{i,t}}$	Availability factor of unit i in time step t (-)
$\overline{G_i}$	Capacity of unit i (MW)
c_i^{gen}	Average generation cost of unit i (€/MWh)
Dt	Demand in time step t (MWh)
em_i	Average emissions of unit i (tCO_2/MWh)
p^{CO_2}	Carbon price (€/tCO2)
VOLL	Value of lost load (€/MWh)

Sets and indices

$i \in I$	Set of generators
$t \in T$	Time steps

Variables

λ_t	Electricity market price at time t (€/MWh)
ens_t	Energy not served at time t (MWh)
$g_{i,t}$	Generation of unit i in time step t (MWh)

As discussed in the introduction, the input vector x is a result of many individual agent's bidding behaviour, which are driven by different preferences. In addition the feasible region of the problem, characterized by, e.g., the technical constraints of the dispatched generators $y^{DP} \in Y^{DP}$ may not be fully known by the modeler. Although the NNs in the introduced MPNNC model have the capability to capture these complex often nonlinear and non-convex dynamics [25], we use a stylized economic dispatch (ED) model to simulate the environment. The motivation is to have an ideal benchmark to the presented data-driven MPNNC model, namely the classical bi-level model (MPEC), which can be solved to global optimality for the stylized environment. The ED model is formulated below:

$$\text{ED}: \min_{g_{i,t}, ens_t} \mathcal{C}(g_{i,t}, ens_t) = \tag{6a}$$

$$\sum_{i \in I^D} \sum_{t \in T} (c_i^{gen} \cdot g_{i,t} + p^{CO_2} \cdot em_i \cdot g_{i,t})$$

$$+ \sum_{t \in T} ens_t \cdot VOLL$$

subject to

$$\sum_{i \in \mathcal{I}} g_{i,t} + ens_t = D_t \quad (\lambda_t) \quad \forall t \in T \tag{6b}$$

$$0 \leq g_{i,t} \leq AF_{i,t} \cdot \overline{G}_i \quad \forall i \in I, \forall t \in T \tag{6c}$$

The objective function, Eq. (6a), accumulates the generation cost along with emission costs of operating the system and the social cost of the curtailed load, i.e., the energy that is not served. Constraint (6b) enforces the energy balance for all time steps, while Constraint (6c) represents the physical limits of the generation units. λ_t shown on the RHS of Eq. (6b) is the corresponding dual variable that, by definition, is the market price resulting from the ED model [26]. Problem (6) represents the electricity market clearing in its simplest form. Firstly, the solution space of the above ED problem, defined by Eq. (6b-6c) is fully convex, while in reality, due to the unit commitment decisions, it is nonconvex. Another difference originates from the parametrization of the objective function that, in practice, results from the imperfect decision making of the market participants. Representing such strategies is particularly challenging due to the existence of information asymmetry and the irrational (and possibly dynamic) risk-attitude. Even if the modeler could capture all non-convexities and represent the "true" underlying decision model of the participants, there may exist multiple equilibrium to this problem, and they can significantly differ [27].

Tab. 1: Generator data, used in the economic dispatch model.

	number of units	c_i^{gen}	em_i	G_{min}	G_{max}
CCGT	4	36.36	0.38	0	450
Nuclear	3	9.09	0.0	0	1200
Coal	2	25.0	0.85	0	800
OCGT	5	50.0	0.53	0	50

B KKT conditions of the ED problem

Let us assume that μ, γ denote the dual variables belonging to the LL's equality and inequality constraints respectively. Then, the KKT conditions are formulated for a generic optimization problem subject to equality and inequality constraints $(h(y), g(y))$ as shown below. The KKT conditions derived for the ED formulation (6) are presented in Appendix B.

$$\Delta_y \mathcal{L}(y, \mu, \gamma) = 0 \tag{7a}$$
$$\gamma \cdot g(y) = 0 \tag{7b}$$
$$\gamma \geq 0 \tag{7c}$$
$$g(y) \leq 0 \tag{7d}$$
$$h(y) = 0 \tag{7e}$$

Eq. (7a) is called the Lagrangian stationarity, while Eq. (7b) the complementary slackness conditions. Eq. (7c) enforces dual feasibility and Eq. (7d-7e) the primal feasibility of the original optimization problem. When all Equations of Problem (7) hold, plus the Hessian of the Lagrangian $(\Delta_{yy} \mathcal{L}(y, \mu, \gamma))$ is positive definite, the KKT conditions are necessary and sufficient for optimality.

C Bi-level formulation of the optimal bidding problem

In order to maximize its own objective, the UL agent recognizes the dependency between its actions and the LL market clearing's outcome $f^{EM} : x \mapsto \lambda$. This hierarchical relation is a typical instance of a Stackelberg-game. The leader anticipates the reaction of the follower $y(x), \lambda(x)$ to its decision x to maximize its utility (or minimize its cost) $\mathcal{C}(x, y)$. Formally, such a game may be written as:

$$\text{MPEC} : \min_{D_t} \mathcal{C}(D_t, \lambda_t(x)) = \lambda_t(x) \cdot D_t \tag{8a}$$

subject to

$$0 \leq D_t \leq \overline{D_t} \tag{8b}$$
$$\sum_{t \in T} (\overline{D_t} - D_t) \leq \sum_{t \in T} 0.05 \cdot \overline{D_t} \tag{8c}$$
$$\lambda_t(x) = \arg\max_y \left\{ f(x, y) : y \in \mathcal{Y}(x) \right\} \tag{8d}$$

The difference of the above formulation from the one in Section **??** is that the LL's response is explicitly captured as the optimum of Eq. (8d).

Standard practice in the literature is to model and solve such games as mathematical programs with equilibrium constraints (MPECs). The solution procedure's first step is to transform the bi-level optimization problem into the single-level form, via reformulating the LL (the market clearing). This reformulation may be done by using the Karush-Kuhn-Tucker (KKT) conditions, or by the primal-dual optimality conditions [2].

For the introduced ED Problem (6), the KKT conditions are necessary and sufficient for optimality. However after including the non-convex constraints of the participating generators, or modeling their possibly imperfect bidding behaviour, the KKT conditions would often become neither necessary nor sufficient.

Assuming that one can find meaningful KKT conditions for the underlying LL problem, it still may be difficult to solve the MPEC to global optimality due to two numerical issues both the complementary slackness conditions and the objective function consists of nonlinear and non-convex terms. To

approximate the bi-linear constraints, we use a mixed integer reformulation that is often referred to as bigM method [28]. The resulting problem is an exact reformulation of the original problem, if the M parameters are appropriately chosen. However, as pointed out in [29], this is a non-trivial task. In our case, the nonlinearities in the objective function are not further linearized as in [2], because Gurobi's noncovnex solver handled the resulting MINLP efficiently and it guarantees global optimality.

D Predicted generation dispatch

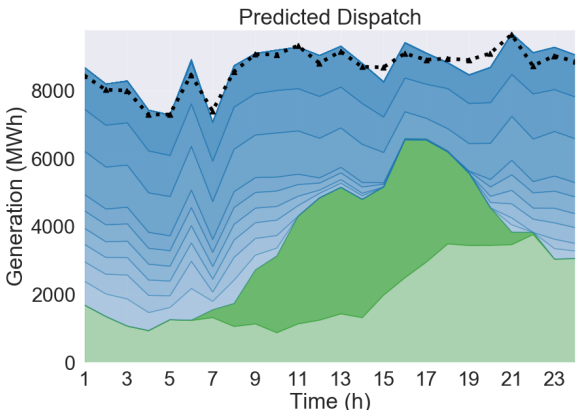

Fig. 3: Stacked plot of a given arbitrarily selected day's predicted dispatch for all conventional generators (shown by blue colors) and for all renewable generators (shown by green color). The aggregated load is indicated by the black dotted line.

E Learning the price model

As second learning step, the resulting market prices are predicted from the dispatch prediction generated by the first NN. After training the second network, it is chained with the first one to capture the relation between the input series (load, RES generation), and the resulting prices (Fig. 1). While achieving the best forecasting performance is not the main focus of this paper, a certain accuracy is desired for a well-performing MPNNC model as the dynamics of the underlying price formation are based on the trained NN. Fig. 4 shows that the price profile of the 50th test day is very closely

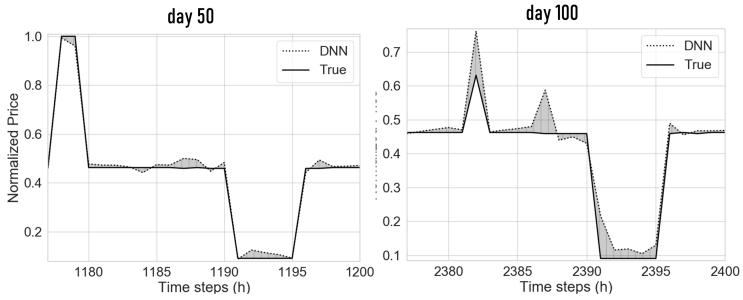

Fig. 4: The scaled price forecast of the 50th and 100th day of the test set. The scaling was done by dividing all values by the maximum observed price (the VoLL).

captured. Both the peak and the off-peak prices are well-represented by the prediction, suggesting that price fluctuations are captured by the trained NN. On the contrary, the predictions made for the 100th test day show larger errors, by overestimating the first peak and creating another price spike that did not occur. Overall the average daily MSE was 0.00289 with a standard deviation of 0.00214, while day 50 had a MSE of 0.00040, and day 100 had a MSE of 0.00232.

F Bidding behaviour of specific days

For day 65, where the MPNNC model had the best performance, it is visible that the submitted load bids (top Figure) deviate from the original load at around the same time intervals (at around noon and in the evening). The resulting price profiles (bottom Figure) show that the MPNNC model reduces the price to below 10 € per MWh from 45 € per MWh between 11:00 and 20:00, while the MPEC achieves the same reduction in a shorter time window (19:00-22:00). Looking at the opposite case, in which the MPNNC performed the worst compared to the MPEC model, it can be noted that the price cap (1000 € per MWh) that was reached in the beginning of the day, was reduced by both models. In the second half of the day, however, the MPEC model manages to keep the prices low for a longer time horizon, resulting in superior performance. The fact that the MPNNC model did not manage to exploit the same price reduction opportunity in the second half of the day, is due to utilizing more flexibility than needed to (i) avoid the price hitting the price cap, (ii) to achieve a small price decrease at 8:00.

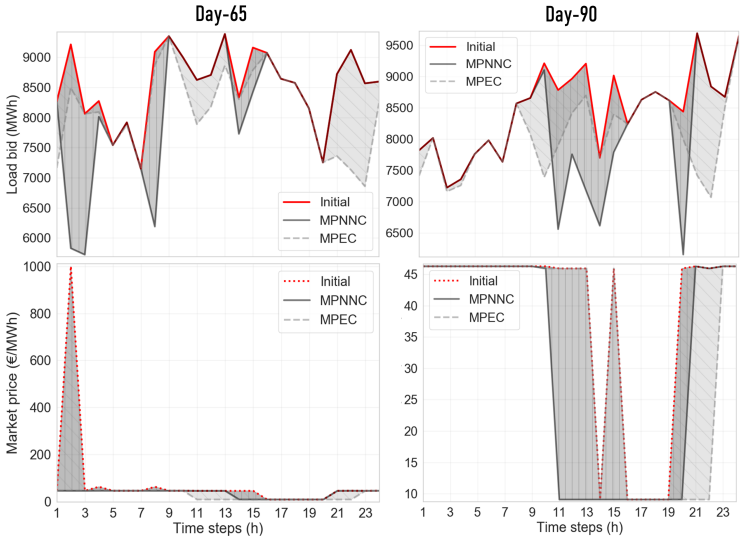

Fig. 5: The best (right two sub-figures) and worst performing days of the MPNNC model compared to the MPEC's performance.

Tab. 2: The summarized and mean solution times and summarized costs of the MPNNC and MPEC models.

	Original	**MPNNC**	**MPEC**
Sum Time (s)	-	52	1196
Avg. Time (s)	-	0.5	12
Sum Cost (€)	1.09e9	6.76e8	5.58e8