Learning in Stackelberg Games with Application to Strategic Bidding in the Electricity Market

Francesco Morri, Hélène Le Cadre, Luce Brotcorne, Pierre Gruet

Learning in Stackelberg Games with Application to Strategic Bidding in the Electricity Market

Francesco Morri
Univ. Lille, CNRS, Inria,
Centrale Lille, UMR 9189 CRIStAL,
F-59000 Lille, France

Hélène Le Cadre, Luce Brotcorne
Univ. Lille, Inria, CNRS,
Centrale Lille, UMR 9189 CRIStAL,
F-59000 Lille, France

Pierre Gruet
EDF R&D and FiME, France

***Abstract*—We formulate a two-stage electricity market involving conventional and renewable producers strategically bidding in the day-ahead market to maximize their profits, while anticipating the market clearing performed by an Independent System Operator (ISO), as a multi-leader single follower Stackelberg game. In this game, producers are interpreted as leaders, while the ISO acts as a follower. To compute an equilibrium, the classical approach is to cast the Stackelberg game as a Generalized Nash Game (GNG), replacing the ISO's optimization problem by its KKT constraints. To solve this reformulated problem, we can either rely on the Gauss-Seidel Best-Response method (GS-BR), or, on the Alternating Direction Method of Multipliers (ADMM). However, both approaches are implemented in a centralized setting since they require the existence of a coordinator which keeps track of the history of agents' strategies and sequential updates, or, is responsible for the Lagrange multiplier updates following the augmented Lagrangian. To allow the agents to selfishly optimize their utility functions in a decentralized setting, we introduce a variant of an actor-critic Multi-Agent Deep Reinforcement Learning (MARL) algorithm with provable convergence. Our algorithm is innovative in that it allows different levels of coordination among the actors and the critic, thus capturing different information structures of the Stackelberg game. We conclude this work by comparing GS-BR and ADMM, both used as benchmark, to the MARL, on a dataset from the French electricity market, relying on metrics such as the efficiency loss and the accuracy of the solution.**

I. Introduction

The study of systems composed of multiple strategic agents, learning simultaneously, is notoriously difficult – in particular for a large number of agents and complex environments. A key topic lies on the identification of conditions for an equilibrium to exist and on the design of distributed algorithms to reach it. This problem can be well framed in the setting of computational game theory, and has been recently applied to perform efficient market clearing on the day-ahead electricity market. Multiple techniques have been introduced to solve the problem, and we can split them into theoretically-driven approaches and agent-driven simulations, which are those we are interested in (see [1] for a more complete list of works in this area). These comprehends approaches that represent the electricity producers as agents following some sets of predefined rules, such that there is no explicit need to solve an optimization problem and it is possible to model the agents' behaviour in dynamic settings. The drawback of these approaches is being limited by the handmade set of rules. Recently, different Reinforcement Learning (RL) techniques have been applied (see [2] and references therein for a survey) to tackle various problems such as the planning of building energy management systems, hybrid or electric vehicle charging, economic dispatch problems, clearings and operations of energy markets [3]. The implementation of RL algorithms to strategic bidding within auctions is promising, because in these settings the interactions are complex and usually modelled in such ways that it is impossible to explicit them in closed form, furthermore the environment evolves dynamically and can change very rapidly. The literature in this area can be classified depending on the number of agents considered, the complexity of the market structure, the shared information patterns, and whether renewable sources are considered. In [4, 5, 6], simulations involving multiple conventional producers are studied under various market designs. Among these articles, only [4] implements continuous actions and states, while the others focus on discrete settings. Most articles in the literature dealing with renewable producers focus on forecasting issues, storage planning and control. Questions related to equilibrium seeking in peer-to-peer electricity markets have been recently considered, e.g. in [7, 8].

A. Main Contributions

We study a market formulated as a multi-leader single-follower Stackelberg game, which we reformulate as a single level non-cooperative game with coupling constraints. In addition, we introduce a Multi-Agent Reinforcement Learning (MARL) algorithm with provable convergence, to dynamically simulate the market with learning energy producers. Finally, we test our model on real data from the French electricity market. MARL allows full decentralization of the agents' decision process, since every producer acts selfishly, maximizing its own utility function. Furthermore, MARL is easily adaptable to settings involving different coordination schemes and information structures.

The paper is organized as follows. In Sec. II, we formulate the strategic bidding problem as a generalized Stackelberg game, we analyse it through classical approaches in Sec. III, while in Sec. IV we introduce the main concepts of Deep RL and the algorithm we use. Finally in Sec. V numerical

979-8-3503-8174-0/24/$31.00 ©2024 IEEE

experiments on the French electricity system illustrate the results.

II. Strategic Bidding as a Multi Leader Single Follower Stackelberg Game

We now formulate our electricity market as a Stackelberg game, describing the agents, their utility functions and decision variables.

A. Stackelberg Game Definition

A Stackelberg game involves a hierarchy between players [9]: a *leader* acts first and a *follower* reacts rationally by computing its best response after observing the leader's action. In our settings we are interested in games with multiple leaders and a single follower. We assume the we have a set $\mathcal{L}$ of L leaders, each leader's objective function is denoted $f_l(x_l, x_{-l}, y),\ \forall l \in \mathcal{L}$, where x_l is the decision variable for leader l, x_{-l} are the other leaders' decision variables and y is the decision variable of the follower. A multi-leader single follower Stackelberg game can be formulated as follows:

$$\min_{x_l \in \mathcal{X}_l} \{f_l(x_l, x_{-l}, y') | y' \in \operatorname*{argmin}_y g(\boldsymbol{x}, y)\},\ \forall l \in \mathcal{L},$$
$$\min_{y \in \mathcal{Y}} \{g(\boldsymbol{x}, y)\}.$$

B. Agents

We consider a set of strategic learning-based and non strategic (static) producers, taking positions on the day-ahead electricity market by submitting offer curves corresponding to their forecasts of the power production. The non-strategic producers always bid at their production costs (also called marginal costs), while the strategic learning-based producers can adapt their strategies, learning from the outcome of the market clearing. The learning producers are made of a mix of conventional and renewable producers.

a) Conventional Producers: The set of learning-based producers is $\mathcal{I}$. The decision variable for each producer $i \in \mathcal{I}$ is its bid $b_i \in \mathcal{B}_i$, where $\mathcal{B}_i := \{x \in \mathbb{R}_+ : c_i \leq x \leq b_i^{max}\}$, and they have two parameters: the marginal cost c_i and the capacity v_i^{max}. Their objective function is: $J_i^C(b_i, b_{-i}, v_i) = (c_i - \lambda(\boldsymbol{b}))\, v_i$, which is the negative of their profit. $\lambda(\boldsymbol{b})$ is the uniform price of the market, dependent on all bids, v_i is the volume requested to producer i. We can write the problem of producer $i \in \mathcal{I}$ as follows:

$$\min_{b_i}\ J_i^C(b_i, b_{-i}, v_i),$$
$$\text{s.t. } b_i \in \mathcal{B}_i.$$

The set of non strategic conventional producers is $\mathcal{J}$, with $b_j^{max} = c_j, \quad \forall\ j \in \mathcal{J}$. Thus, the set of general conventional producers is denoted $\mathcal{O} = \mathcal{I} \cup \mathcal{J}$.

b) Renewable Producers: We define the set of renewable producers as $\mathcal{K}$. The decision variables are still their bids, with the same constraints as for the conventional producers. The main difference is in the parameters: their marginal cost c_k is 0, and their capacity v_k^{max} is a random variable. We then modify the objective function to account for these differences: $J_k^R(b_k, b_{-k}, v_k) = (c_k - \lambda(\boldsymbol{b}))\, v_k + \mathbb{E}_{\epsilon_k}[\Phi(\epsilon_k)]$. ϵ_k is the *imbalance volume* for producer k, representing over or under production of electricity, and $\Phi(\cdot)$ is the penalty function defined by the market operator. The real value of ϵ_k is only known after the market has cleared, hence we consider the expectation of the imbalance penalty. The capacity v_k^{max} and the imbalance volume ϵ_k can be taken directly from the data, or simulated according to a probability density function: we can write the observed capacity of producer k as $\hat{v}_k^{max} = v_k^{max} + \epsilon_k$, with $v_k^{max} \sim \mathcal{D}_k(.)$, $\mathcal{D}_k(.)$ obtained from the data, and $\epsilon_k \sim \mathcal{N}(0, \sigma_k)$, $\sigma_k > 0$ being the standard deviation. The full problem for producer k can then be written as:

$$\min_{b_k}\ J_k^R(b_k, b_{-k}, v_k),$$
$$\text{s.t. } b_k \in \mathcal{B}_k.$$

Let $\mathcal{N} = \mathcal{I} \cup \mathcal{J} \cup \mathcal{K}$ be the set of producers.

c) ISO: The ISO is the single follower in our game, reacting rationally to the producers' bids. Its decision variables are the volumes $\boldsymbol{v}$, and the parameters for the problem are the capacities of each producer, the bids and the demand. The volumes are constrained to be always less or equal to the capacities:

$$v_n \in \mathcal{V}_n = \{x \in \mathbb{R}_+ : 0 \leq x \leq v_n^{max}\},\ \forall n \in \mathcal{N}.$$

We also introduce $\boldsymbol{v} = (v_n)_{\mathcal{N}}$ and its associated feasibility set $\mathcal{V}$. The ISO's objective function is:

$$J^{ISO}(\boldsymbol{v}) = \sum_{i \in \mathcal{I}} b_i v_i + \sum_{k \in \mathcal{K}} b_k v_k + \sum_{j \in \mathcal{J}} c_j v_j,$$

which consists in the total cost to run the market. The goal for the ISO is to minimize this cost, while satisfying the demand and respecting the capacities. The complete problem can be written as:

$$\min_{\boldsymbol{v}}\ J^{ISO}(\boldsymbol{v}), \tag{4a}$$
$$\text{s.t.:}\quad v_n \in \mathcal{V}_n\ \ \forall n \in \mathcal{N}, \tag{4b}$$
$$\sum_{n \in \mathcal{N}} v_n \geq d\ \ (\lambda). \tag{4c}$$

The clearing price λ is obtained as the dual variable of the supply-demand balance constraint in Eq. (4c).

C. Market Formulation as a Stackelberg Game

We can now introduce the Stackelberg game using the compact formulations introduced in the previous section:

$$\textbf{L Conv.:}\ \min_{b_o \in \mathcal{B}_o}\ J_o^C(b_o, b_{-o}, v_o), \quad \forall o \in \mathcal{O},$$
$$\textbf{L RES:}\ \min_{b_k \in \mathcal{B}_k}\ J_k^R(b_k, b_{-k}, v_k), \quad \forall k \in \mathcal{K},$$
$$\textbf{F ISO:}\ \min_{\boldsymbol{v} \in \mathcal{V}}\ J^{ISO}(\boldsymbol{v}),$$
$$\text{s.t.:} \sum_{n \in \mathcal{N}} v_n \geq d,\ \ (\lambda)$$

where **L** stand for the leaders (producers) and **F** for the follower (ISO).

Proposition 1 (Uniqueness of the ISO Solution)**.** *If the bids are all different, the solution of the ISO problem is unique.*

Proof. The proof can be found in Appendix B. □

D. KKT Reformulation

Proposition 2 (Slater's Constraint Qualification)**.** *Slater's conditions for constraint qualification hold.*

Proof. The proof is straightforward and relies on the existence of large enough capacities for each producer, such that there

exists a solution $\boldsymbol{v}^*$ checking $\sum_{n\in\mathcal{N}} v_n^* > d$, with $0 < v_n^* < v_n^{max}$, $\forall\, n \in \mathcal{N}$. □

Under Prop. 2, the ISO's optimization problem (4) can be replaced by its KKTs, leading to a Generalized Nash Game (GNG). We assume the bids are affine functions of the volumes, i.e., $b = \alpha v + c$, with α, c different for each producer. Inserting the affine bids in Eq. (4a), and introducing the notation $\mathcal{M} = \mathcal{I} \cup \mathcal{K}$ for the learning producers, we obtain the following Lagrangian function for the ISO's problem:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{v}, \lambda) = \; & [\textstyle\sum_{m\in\mathcal{M}}(\alpha_m v_m + c_m)v_m + \sum_{j\in\mathcal{J}} c_j v_j] \\ - \; & \lambda\Big(\textstyle\sum_{n\in\mathcal{N}} v_n - d\Big). \end{aligned} \tag{6}$$

III. Benchmark Algorithms

Using Eq. (6) we obtain a single level problem for each producer, so that we can exploit classical optimization algorithms and use the results as benchmarks for the RL approach. Note that the producers may have different valuations of the clearing price λ.

A. Best-Response Approach

The first algorithm we implemented is Gauss-Seidel Best Response (GS-BR) [10]. To apply GS-BR, we relax the balancing constraint, by introducing a penalty in the producers' objective functions weighted by the constant $M \geq 0$:

$$\min_{\alpha_i, v_i, \lambda_i} (c_i - \lambda_i)v_i + M\Big(\sum_{n\in\mathcal{N}} v_n - d\Big)^2, \tag{7a}$$

$$\text{s.t.:} \quad c_i \leq \alpha_i v_i + c_i \leq b_i^{max}, \tag{7b}$$

$$v_i \in \mathcal{V}_i, \tag{7c}$$

$$\sum_{m\in\mathcal{M}} 2v_m\alpha_m + \sum_{n\in\mathcal{N}} c_n - |\mathcal{N}|\lambda_i = 0. \tag{7d}$$

The value of M is tuned from running multiple simulations.

B. ADMM Approach

In order to apply the ADMM algorithm we reformulate the problem following [11]: since we need a separable problem, we introduce some *tracking variables*, so that we can get rid of the shared coupling between the producers. In particular, we introduce $\tilde{d}_i = d - \sum_{n,n\neq i} v_n$, as a measure of the supply and demand balance, and $\pi_i = \sum_{m,m\neq i} 2v_m\alpha_m$, for the coupling constraint. The new problem is the following:

$$\begin{aligned} \min_{\alpha_i, v_i, \lambda_i} \; & (c_i - \lambda_i)v_i + (v_i - \tilde{d}_i)^2, \\ \text{s.t.:} \quad & c_i \leq \alpha_i v_i + c_i \leq b_i^{max}, \\ & v_i \in \mathcal{V}_i, \\ & 2v_i\alpha_i + \pi_i + \sum_{n\in\mathcal{N}} c_n - |\mathcal{N}|\lambda_i = 0, \end{aligned}$$

which gives the augmented Lagrangian:

$$\begin{aligned} \mathcal{L}_i(\alpha_i, v_i, \lambda_i) = \; & (c_i - \lambda_i)v_i + (v_i - \tilde{d}_i)^2 + \\ & \beta_i(2v_i\alpha_i + \pi_i + \sum_{n\in\mathcal{N}} c_n - |\mathcal{N}|\lambda_i) + \\ & \frac{\rho}{2}||2v_i\alpha_i + \pi_i + \sum_{n\in\mathcal{N}} c_n - |\mathcal{N}|\lambda_i||^2, \end{aligned}$$

where β is a Lagrangian multiplier and ρ is the penalty constant, chosen by hand to obtain a stable problem. We iteratively update the variables and the multiplier following the rules from [12].

IV. Deep RL Algorithm

The fundamental idea of RL is to have a model experiencing an environment previously unknown and learning by interacting with it. This is achieved by defining algorithms that aim to maximize a reward, while exploring different strategies or policies, until the optimal is closely approached. RL problems can be formalized relying on *Markov Decision Processes*, which involve a state space $\mathcal{S}$, an action space $\mathcal{A}$ and a reward function $r : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$. We let $s_t \in \mathcal{S}$ be the agent's state and $a_t \in \mathcal{A}$ be its action at time period t. The transition dynamics in this setting satisfies the Markov property, i.e.:

$$p(s_{t+1}|s_t, a_t) = p(s_{t+1}|s_1, a_1, \ldots, s_t, a_t).$$

The agent decides what action to take based on its *policy* $\pi(s_t) = a_t$. We let $R := \sum_{t=0}^{t=\infty} \gamma^t r_{t+1}$ be the discounted return over a trajectory defined by a sequence $s_1, a_1, r_1, s_2, a_2, r_2, \ldots$, with $\gamma \in [0; 1]$ the discount factor. We can now define the *Q-value function*: $Q_\pi(s_t, a_t) := \mathbb{E}[R|s_t, a_t, \pi]$. It represents the expected return given a state, an action and a policy. The aim of an agent is to obtain the optimal policy, the one maximizing the Q-value function:

$$Q^*(s_t, a_t) := \max_\pi Q_\pi(s_t, a_t)$$

The goal of an RL algorithm is then to approximate the optimal policy through learning. There are many different techniques to do so, usually divided by whether they can handle continuous or discrete settings. The simpler algorithms, such as *Q-Learning* [13] or *SARSA* [14], usually can be studied theoretically, while the more general and complicated ones, introducing deep neural networks, add a further level of complexity on the theoretical side.

A. `DDPG` Algorithm

We consider the `DDPG` (Deep Deterministic Policy Gradient) [15] algorithm, that belongs to the class of *actor-critic* architectures [16]: these types of algorithms simultaneously learn the policy and the Q-value function, making it possible to apply them to a much broader spectrum of problems. The Deep Deterministic Policy Gradient (DDPG) algorithm employs an actor $\pi_\theta(\cdot)$, where θ represents the network's parameters, to specify the current policy, and a critic $Q_\omega(\cdot)$, parameterized by ω, to represent the Q-value function. In this type of algorithm, copies of the actor and critic networks are also employed during training for stability issues, called *target networks*, we indicate them with $\pi_{\theta_\sharp}, Q_{\omega_\sharp}$. The objective function of the actor is:

$$\mathcal{L}_\pi(s) = \mathbb{E}_\mathcal{B}[Q_\omega(s, \pi_\theta(s))],$$

where $\mathcal{B}(\cdot)$ is the replay buffer, which is a collection of past instances from which a batch of data is sampled and used to train the network. Through gradient ascent this function is then maximized. The critic loss function is based on the Bellman's

equation (MSBE Loss: *Mean Squared Bellman's Equation*) which takes the following form:

$$\mathcal{L}_Q(\sigma) = \mathbb{E}_{\mathcal{D}}\Big[\big(Q_\omega(s,a) - (r + \gamma Q_{\omega_\sharp}(s', \pi_{\theta_\sharp}(s')))\big)^2\Big],$$

where now instead of just sampling s from $\mathcal{B}(\cdot)$, we sample the tuple $\sigma = (s, a, r, s')$, with s' and r being the state and reward resulting from taking action a when observing state s.

B. Stackelberg DDPG and Proof of Convergence

We now briefly introduce the first modified version of the DDPG algorithm, as proposed in [17]. The idea is to identify a Stackelberg game between the actor and the critic, in order to exploit the added information when computing the gradients to update the neural networks. Notice that the game between the actor and the critic is a different game than the one of the producers, as it is depicted in Fig. 1. Let $x_a \in \mathbb{R}, x_c \in \mathbb{R}$ be the decision variables of the actor-leader and critic-follower respectively, then we can define the cost function $f_a(x_a, x_c)$ and $f_c(x_a, x_c)$. We can write the best response of the follower as $x_c^*(x_a) := \operatorname{argmin}_y f_c(x_a, y)$, and using this we obtain the full gradient of the cost function for the leader (using the implicit function theorem):

$$\nabla f_a(x) = \nabla_a f_a(x) - \left((\nabla_c^2 f_2(x))^{-1} \nabla_{ca} f_c(x)\right)^\top \nabla_c f_a(x), \quad (10)$$

where $x = (x_a, x_c^*(x_a))$. Using Eq. (10) as the update rule for the DDPG algorithm, it is possible to prove the convergence towards an equilibrium $x^* = (x_a^*, x_c^*)$ in the single agent case (we refer to the original work [17] for further details). The proof of [17] does not extend straightforwardly to a multi-agent setting, due to the coupling between agents. We propose two ways to extend the work to multi-agent settings: first, it is possible to uncouple the agents passing to each of them a state independent of the other agents' actions (such as the demand); the second approach is to change the architecture in such a way that the coupling is no longer a problem. The first approach limits the information that the producers can access, thus we focus on the latter which is more general.

Consider the set of players defined by one actor and multiple critics $(a, c_1, \dots, c_n)$, with decision variables $x_a \in \mathbb{R}, x_{c_i} \in \mathbb{R}, \ \forall \ i = 1, \dots, n$. We define the objective functions $(F_a, f_{c_1}, \dots, f_{c_n})$ for the actor and the critics respectively, with $F_a, f_{c_i} \in C^q(\mathbb{R}, \mathbb{R}), \ q \geq 2, \ i \in [\![1, n]\!]$. The strategy update of the actor and the critics is controlled by a parameter γ called learning rate, that has to satisfy: $\gamma_a = o(\gamma_{c_i}), \ \forall \ i = 1, \dots, n$. Let $x^* := (x_a^*, \boldsymbol{x}_c^*)$ be an equilibrium of the actor-critics game.

Theorem 1. *There exists a neighborhood $\mathcal{U}$ of x^* s.t. for any $x_0 \in \mathcal{U}$, x_t converges almost surely to x^* as $t \to +\infty$.*

Proof. The proof is a direct generalization of [17]. □

V. Simulation Settings and Experiments

We now introduce the settings of our experiments. In the following sections, we refer to the modified DDPG algorithm as mDDPG, to our version as SAMC and to the case of uncoupled producers as IND, also BR and ADMM represent GS-BR and ADMM algorithm respectively.

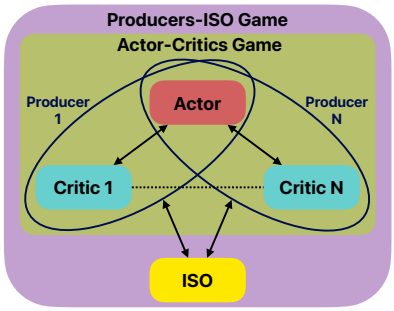


Fig. 1. Schematic representation of Stackelberg DDPG algorithm

A. Data

To run the simulation and train the agents we use data from the French electricity market, for the period between 2020 and 2023 [18].

B. Results

The benchmark and the RL algorithm operate on different time frames: the ADMM and best-response approaches compute an equilibrium at each time step, while the RL algorithm converges towards an equilibrium in a finite number of time steps. This means that at each step, *the RL simulation may not be at equilibrium*. Another key difference is that in both classical reformulations, each producer computes its own price λ_i, which may not to be the same across all the agents. Furthermore, in the RL simulation we have two learning agents (producer 0 and 1) and three static ones, as for the benchmark: only two agents can change their bids; the others are fixed. In the discussion following, we will only compare the data obtained from these two agents. We will compare the methods in two ways: first we aim to compare the profits of the different producers, then we will insert the data from the RL simulation into the constraints obtained by the KKT reformulation. Regarding the profits, we compare the data obtained from the last 20000 steps of the RL simulation (10 epochs, each consisting of 2000 steps) with a simulation of equivalent length using the classical algorithms. We choose this length empirically such that it allows spanning the demand data multiple times, so that we can compute meaningful averaged quantities. We report in Tab. I and Tab. II the profits of the two learning agents and the cost to run the market respectively. In RL, each producer has no access to any information regarding the other producers, in contrast with the BR and ADMM approach, where each producer needs access to the decision variables of the rest of the producers. From Tab. I we can see that all the methods generate higher profits than no learning, with the ADMM algorithm giving rise to the highest average profits. On the other hand, ADMM and BR are also associated with more volatile results. Between the RL algorithms, mDDPG and SAMC generate the two best results for producer 0, and mDDPG also achieves the highest profit for producer 1. Looking at the total cost for the market in Tab. II, we observe that ADMM generates

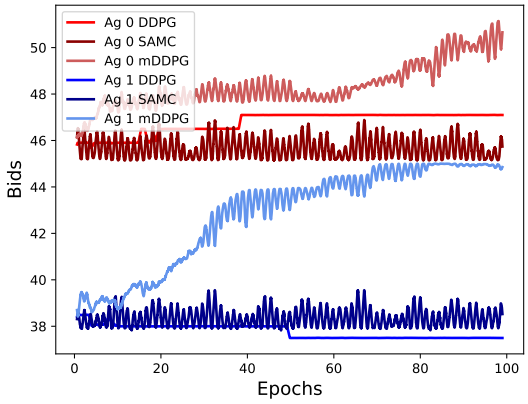


Fig. 2. Moving average of the strategies of the two learning producers in three different cases: the standard algorithm DDPG, the modified version mDDPG and our modified architecture SAMC. One epoch consists of

the highest average cost, while between the RL algorithms mDDPG has the highest cost. It is interesting to notice that SAMC has the same costs as DDPG and IND, while achieving similar profits for producer 1 and much higher profits for producer 0.

TABLE I
AVERAGE PROFITS OF AGENTS 0 AND 1 FOR A 10 EPOCHS PERIOD

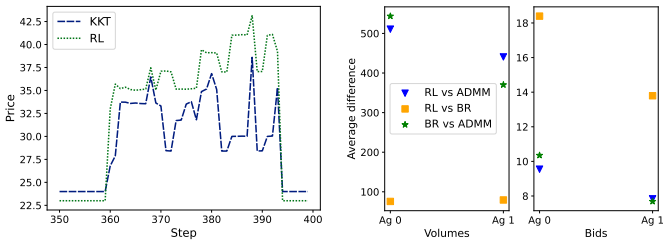
Algorithm	Profit Ag 0	Profit Ag 1
DDPG	$(0.5 \pm 0.1) \cdot 10^2$	$(28 \pm 1.2) \cdot 10^2$
mDDPG	$(6 \pm 6) \cdot 10^2$	$(32 \pm 15) \cdot 10^2$
SAMC	$(4 \pm 2) \cdot 10^2$	$(23 \pm 3) \cdot 10^2$
IND	$(0.6 \pm 0.2) \cdot 10^2$	$(30 \pm 3) \cdot 10^2$
No Learning	0	$1.36 \cdot 10^2$
BR	$(0.4 \pm 2) \cdot 10^2$	$(24 \pm 42) \cdot 10^2$
ADMM	$(30 \pm 30) \cdot 10^2$	$(60 \pm 36) \cdot 10^2$

TABLE II
AVERAGE COST TO RUN THE MARKET FOR A PERIOD OF 10 EPOCHS

Algorithm	Total Cost
DDPG	$29 \cdot 10^3$
mDDPG	$38 \cdot 10^3$
SAMC	$30 \cdot 10^3$
IND	$30 \cdot 10^3$
No Learning	$11 \cdot 10^3$
BR	$36 \cdot 10^3$
ADMM	$47 \cdot 10^3$

In Fig. 2 we highlight the stability differences between the different RL algorithms. The results for SAMC are particularly interesting, since it captures the dynamic nature of the environment similarly to mDDPG, while being more stable throughout all the simulations we ran.

In Fig. 3a, we compare the price obtained by the RL simulation with the one obtained by inserting the volume and bids from the simulation into Eq. (7d) (and assuming all λ_i are equal). The plot highest variability corresponds to epochs where the learning producers volume is activated.



(a) (b)

Fig. 3. In Fig. 3a we compare the price computed dynamically in the simulation (labeled RL) and the one obtained by inserting the results of the simulation in Eq. (7d) (labeled KKT), while in Fig. 3b we compare the bids and volumes between algorithms.

The data reflects that affine bids do not capture well the complex behaviour of the learning algorithm, as well as the fact that there is no guarantee of uniqueness of equilibrium. Lastly, in Fig. 3b, we compare the volumes and bid from the RL simulation, the best response algorithm and the ADMM algorithm. We averaged the absolute differences for the two learning agents over a number of steps such that the demand data would be passed through once, and for the RL algorithm, we took the data from the end part of the simulation. The different algorithms may compute different equilibria, since we have no guarantee of uniqueness of the market equilibrium. From the data we can see that the results of BR and RL are more similar for the volumes, while we observe differences in the bids. This clearly highlights that different market equilibria can be reached relying on GS-BR, ADMM, and variants of RL. In an extension, we may combine these algorithms to enable an automatic spanning of the set of mixed market equilibria.

VI. CONCLUSIONS AND FUTURE DEVELOPMENTS

In this work we presented a simple but still relevant model for the problem of strategic bidding in the electricity market with multiple producers, involving learning agents strategically bidding in the day-ahead market and static agents. To compute market equilibria, we rely on classical best-response and consensus variant algorithms (namely Gauss-Seidel Best Response and ADMM), while comparing them with a Deep RL based dynamic simulation which aim to learn a market equilibrium. On the algorithmic side, we extend results from the literature to prove the MARL algorithm convergence to a market equilibrium. This work is an important step towards the more systematic use of theoretically-backed Machine Learning algorithms in real world applications, such as electricity markets.

Next developments will involve studying the scalability of the algorithms by considering a larger number of learning agents and the automatic spanning of the set of equilibria. From the policy-side, we aim to apply MARL approaches to propose innovative designs for the ancillary markets, which are currently rather immature markets, involving a limited number of strategic – possibly bounded rational – stakeholders.

References

[1] S. P. Mathur, A. Arya, and M. Dubey, "Optimal bidding strategy for price takers and customers in a competitive electricity market," *Cogent Engineering*, vol. 4, no. 1, p. 1358545, 2017. [Online]. Available: https://doi.org/10.1080/23311916.2017.1358545

[2] A. Perera and P. Kamalaruban, "Applications of reinforcement learning in energy systems," *Renewable and Sustainable Energy Reviews*, vol. 137, p. 110618, Mar. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1364032120309023

[3] T. Yang, L. Zhao, W. Li, and A. Y. Zomaya, "Reinforcement learning in sustainable energy and electric systems: a survey," *Annual Reviews in Control*, vol. 49, pp. 145–163, 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1367578820300079

[4] Y. Ye, D. Qiu, J. Li, and G. Strbac, "Multi-period and multi-spatial equilibrium analysis in imperfect electricity markets: A novel multi-agent deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 130 515–130 529, 2019.

[5] D. Esmaeili Aliabadi, M. Kaya, and G. Sahin, "Competition, risk and learning in electricity markets: An agent-based simulation study," *Applied Energy*, vol. 195, pp. 1000–1011, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261917303677

[6] M. Rahimiyan and H. R. Mashhadi, "An adaptive q-learning algorithm developed for agent-based computational modeling of electricity market," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 547–556, 2010.

[7] R. Taisant, M. Datar, H. Le Cadre, and E. Altman, "Learning market equilibria using performative prediction: Balancing efficiency and privacy," in *European Control Conference*. IEEE, 2023, pp. 1–8.

[8] G. Belgioioso, W. Ananduta, S. Grammatico, and C. Ocampo-Martinez, "Energy management and peer-to-peer trading in future smart grids: A distributed game-theoretic approach," in *European Control Conference*. IEEE, 2020, pp. 1324–1329.

[9] T. Başar and G. Olsder, "Dynamic Non-Cooperative Game Theory," vol. 160, Jan. 1999.

[10] F. Facchinei, V. Piccialli, and M. Sciandrone, "Decomposition algorithms for generalized potential games," *Computational Optimization and Applications*, vol. 50, no. 2, pp. 237–262, Oct. 2011. [Online]. Available: http://link.springer.com/10.1007/s10589-010-9331-9

[11] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," *Automatica*, vol. 117, p. 108962, Jul. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109820301606

[12] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010. [Online]. Available: http://www.nowpublishers.com/article/Details/MAL-016

[13] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992. [Online]. Available: https://link.springer.com/article/10.1007/BF00992698

[14] G. Rummery and M. Niranjan, "On-Line Q-Learning Using Connectionist Systems," *Technical Report CUED/F-INFENG/TR 166*, Nov. 1994.

[15] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[16] V. Konda and J. Tsitsiklis, "Actor-Critic Algorithms," in *Advances in Neural Information Processing Systems*, vol. 12. MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1999/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[17] L. Zheng, T. Fiez, Z. Alumbaugh, B. Chasnov, and L. J. Ratliff, "Stackelberg Actor-Critic: Game-Theoretic Reinforcement Learning Algorithms," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, pp. 9217–9224, Jun. 2022. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20908

[18] "ENTSO-E Transparency Platform." [Online]. Available: https://transparency.entsoe.eu/dashboard/show

A. Uniqueness of ISO solution

The ISO problem has the following properties:

- the objective function is linear in the decision variables
- we assume the feasibility set is nonempty (this just amounts to assume;$\sum_{o\in\mathcal{O}} v_o^{max} + \sum_{k\in\mathcal{K}} v_k^{max} \geq d$
- the feasibility set is closed convex (which can be trivially proved using the definition of convex set).

Given these properties we can conclude that the problem has a *unique optimal value*, but may have *multiple optimal solutions*. We can easily see this in the case of multiple producers proposing the same bid: for the ISO it is equivalent requesting volume from all of them, obtaining different solutions with the same optimal cost.

B. Proof of Proposition 1

Let $\{b_i\}_{i\in[0,n]}$ be the agents' bids and $\{x_i\}_{i\in[0,n]}$ be the agents' volumes. Assume $b_i \neq b_j \ \forall i,j$. We can order the volumes by bid in increasing order, which means that $i < j \implies b_i < b_j$. Then, the optimal solution can be built as follows:

$$
\begin{aligned}
x_0 &= \min\{v_0^{max}, d\} \\
x_1 &= \min\{v_1^{max}, \min\{0, d - x_0\}\} \\
&\vdots \\
x_n &= \min\{v_n^{max}, \min\{0, d - \sum_{i=0}^{n-1} x_i\}\}.
\end{aligned}
$$

We indicate by m the index of the last volume which is different from 0, $m \in [\![0,n]\!]$, so that the total cost is $\sum_{i=0}^{m} b_i x_i$. Now, assume there is a different solution with lower or equal price, i.e., $\sum_{i=0}^{m'} b_i x'_i \leq \sum_{i=0}^{m} b_i x_i$. Note that $m' > m$ by construction. Let $\Delta x_i := x_i - x'_i$. We can write:

$$
\begin{aligned}
&\sum_{i=0}^{m} b_i(x'_i - x_i) + \sum_{j=m+1}^{m'} b_j x'_j \leq 0, \\
&\sum_{j=m+1}^{m'} b_j x'_j \leq \sum_{i=0}^{m} b_i(x_i - x'_i) = \sum_{i=0}^{m} b_i \Delta x_i.
\end{aligned}
$$

Note that we have $b_j > b_i, \ \forall \ (i \in [\![0,m]\!], j \in [\![m+1, m']\!])$. Define $\tilde{b}^0 = b_0$. We can write $b_l = \tilde{b}^0 + \tilde{b}^0_l$, where $\tilde{b}^0_0 = 0$. Replacing the bids with this new formulation we get:

$$
\sum_{j=m+1}^{m'} (\tilde{b}^0 + \tilde{b}^0_j) x'_j \leq \sum_{i=0}^{m} (\tilde{b}^0 + \tilde{b}^0_i) \Delta x_i,
$$

which can be rewritten as:

$$
\tilde{b}^0 \sum_{j=m+1}^{m'} x'_j + \sum_{j=m+1}^{m'} \tilde{b}^0_j x'_j \leq \tilde{b}^0 \sum_{i=0}^{m} \Delta x_i + \sum_{i=0}^{m} \tilde{b}^0_i \Delta x_i.
$$

Given that the total demand is fixed we obtain: $\sum_{j=m+1}^{m'} x'_j = \sum_{i=0}^{m} \Delta x_i$, and we already mentioned that $\tilde{b}^0_0 = 0$, so we finally obtain:

$$
\sum_{j=m+1}^{m'} \tilde{b}^0_j x'_j \leq \sum_{i=1}^{m} \tilde{b}^0_i \Delta x_i,
$$

where the sum on the right-hand side of the inequality starts from 1. At the next step, we can define $\tilde{b}^1 = \tilde{b}^0_1$, so that $\tilde{b}^0_l = \tilde{b}^1 + \tilde{b}^1_l$ and $\tilde{b}^1_1 = 0$. By repeating the process until step m, we will eliminate every term from the right-hand side sum, thus obtaining as last step $\tilde{b}^{m-1}_j = \tilde{b}^m + b^*_j$:

$$
\sum_{j=m+1}^{m'} b^*_j x'_j \leq 0,
$$

which is a contradiction.