Agent-Based Modeling in Electricity Market Using Deep Deterministic Policy Gradient Algorithm

Yanchang Liang, *Student Member, IEEE*, Chunlin Guo, Zhaohao Ding, *Member, IEEE*, and Huichun Hua

***Abstract*—Game theoretic methods and simulations based on reinforcement learning (RL) are often used to analyze electricity market equilibrium. However, the former is limited to a simple market environment with complete information, and difficult to visually reflect the tacit collusion; while the conventional RL algorithm is limited to low-dimensional discrete state and action spaces, and the convergence is unstable. To address the aforementioned problems, this paper adopts deep deterministic policy gradient (DDPG) algorithm to model the bidding strategies of generation companies (GenCos). Simulation experiments, including different settings of GenCo, load and network, demonstrate that the proposed method is more accurate than conventional RL algorithm, and can converge to the Nash equilibrium of complete information even in the incomplete information environment. Moreover, the proposed method can intuitively reflect the different tacit collusion level by quantitatively adjusting GenCos' patience parameter, which can be an effective means to analyze market strategies.**

***Index Terms*—Electricity market, Nash equilibrium, deep reinforcement learning (DRL), deep deterministic policy gradient (DDPG), game theory, tacit collusion.**

NOMENCLATURE

A. Indices and Sets

$\mathcal{D}$	Set of loads indexed by d.
$\mathcal{D}_i$	Set of loads at bus i, $\mathcal{D}_i \subseteq \mathcal{D}$.
$\mathcal{G}$	Set of GenCos $\{1, 2, \ldots, G\}$ indexed by g.
$\mathcal{G}_i$	Set of GenCos at bus i, $\mathcal{G}_i \subseteq \mathcal{G}$.
$\mathcal{I}$	Set of buses $\{1, 2, \ldots, I\}$ indexed by i.
t	Index for operation intervals.

B. Parameters

α_g^m	Intercept of marginal cost function of GenCo g.
β_g^m	Slope of marginal cost function of GenCo g.
$\boldsymbol{F}$	Vector of the maximum lines flow limits.
γ	Discount factor.
$\mathbf{PTDF}$	Matrix of power transfer distribution factor.
τ	Soft update rate.
θ^μ	Parameter consisting of the weights and biases of the network μ.
θ^Q	Parameter consisting of the weights and biases of the network Q.
f_d	Slope of demand curve of load d.
lr_a	Learning rate of actor network.
lr_c	Learning rate of critic network.
N	Size of a mini-batch.
$p_g^{\max}$	Maximum power generation of GenCo g.
$p_g^{\min}$	Minimum power generation of GenCo g.
T	Total number of operation intervals.

C. Variables

α_{gt}	Intercept of the strategic supply function of GenCo g at t, also called strategic variable.
α_{gt}^*	GenCo g's strategic variable in the NE of static game Γ_t.
$\bar{\lambda}_t$	Average nodal price at t.
$\boldsymbol{p}_t$	Power generation vector for all buses at t.
$\boldsymbol{q}_t$	Power demand vector for all buses at t.
Γ_t	Static game at t.
λ_{it}	Nodal price of bus i at t.
a_{gt}	Action variable of GenCo g at t.
$D_{dt}^{\max}$	Maximum load demand of load d at t.
D_t^{Σ}	Total maximum load demand at t.
n_{gt}	Action noise of GenCo g at t.
q_{dt}	Power demand of load d at t.
R_g	Cumulative payoff of GenCo g.
r_{gt}	Payoff of GenCo g at t.
s_{gt}	State variable of GenCo g at t.

D. Functions

$\mu(s)$	Policy function.
$\mu(s, a\|\theta^\mu)$	Policy function approximated by the neural network with parameter θ^μ.
$\rho_g^m(p_{gt})$	Marginal cost function of GenCo g.
$\rho_{dt}(q_{dt})$	Demand function of load d at t.
$\rho_{gt}(p_{gt})$	Strategic supply function of GenCo g at t.
$C_g^m(p_{gt})$	Cost function of GenCo g.
$J(\theta^\mu)$	Objective function of parameter θ^μ.
$L(\theta^Q)$	Loss function of parameter θ^Q.
$Q(s, a\|\theta^Q)$	Action-value function approximated by the neural network with parameter θ^Q.
$Q(s, a)$	Action-value function.

Manuscript received May 17, 2019; revised September 18, 2019, December 13, 2019, and March 28, 2020; accepted May 30, 2020. Date of publication June 2, 2020; date of current version November 4, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 51907063, in part by the Fundamental Research Funds for the Central Universities under Grant 2019MS054, and in part by the Support Program for the Excellent Talents in Beijing City under Grant X19048. Paper no. TPWRS-00684-2019. *(Corresponding author: Chunlin Guo.)*

The authors are with the State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, Beijing 102206, China (e-mail: liangyancang@gmail.com; gcl@ncepu.edu.cn; zhaohao.ding@ncepu.edu.cn; huahuichun@126.com).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2020.2999536

I. Introduction

NASH equilibrium (NE) is an effective tool for analyzing trading trends, assisting electricity market design and regulation [1]. In the deregulated electric wholesale markets, independent system operators (ISOs) use auction mechanism to clear market based on supply and demand bids submitted by market participants. During the clearing process, market participant, such as generation company (GenCo), independently makes strategic decisions without the information of other competitors, which is a static game of incomplete information.

To solve the NE of this static game, it is usually assumed that the information is complete and game theoretic method is typically adopted as solving method. The Supply Function Equilibrium (SFE) model [2] which represents GenCos' competitive behavior, and the computational optimization tools such as Mathematical Programs with Equilibrium Constraints (MPEC) and Equilibrium Problem with Equilibrium Constraints (EPEC) are widely used in the electricity market [3]. Most of these methods model the market as a single-stage game. However, in electricity markets that run daily or hourly, static game models may not adequately reflect the behaving characteristics of market participants. In empirical studies, GenCos' behavior is more similar to tacit collusion than to the NE of static game [4]. In tacit collusion, there is no explicit agreement between GenCos, and the rise in market prices is simply due to each GenCo's perception of market clearing clearance [5]. In fact, in the case of game-theoretic modeling we observe that in an infinitely repeated static game, one set of circumstances can support many different collusive outcomes, which is often referred in literature as Folk Theorem [6]. Although the Folk Theorem supports the existence of tacit collusion, but it cannot describe the level and characteristics of collusion. In addition, Folk Theorem is still limited to games of complete information, and cannot strictly reflect electricity market with incomplete information.

As an alternative to the game theoretic approach, market simulation based on multi-agent system (MAS) has received more attention because it is low-cost, repeatable, and can reflect market dynamics with incomplete information. Reinforcement Learning (RL) is a type of machine learning technique commonly used in MAS that enables an agent to learn in an interactive environment by trial and error using feedback from their own actions and experiences. The commonly used RL algorithms in the electricity market are Roth-Erev (RE) learning [7], Q-Learning [8] algorithms and their variants, which store the estimated value of all actions or state-action pairs in a table and interact with the environment to update the table. Tabular method makes them suitable only for low-dimensional, discrete state and action spaces, and do not easily converge to optimal behavior. For example, 10 experiments were repeated using the variant Roth-Erev (VRE) algorithm [9] in the same market environment in [10], but each time the results were different, and the standard deviation of the results had reached 30% of the average price. The AMES framework [11] based on the VRE algorithm was used to simulate market equilibrium in [12], but the result deviated from the real NE, and it was found that Q-Learning algorithm has stronger exploration ability than VRE. Q-Learning was used to evaluate the proposed market power mitigation rules for the California electricity market [13], but it was also mentioned that if there are too many decision variables, the Q-Learning model will suffer from the curse of dimensionality.

In order to address the shortcomings of the tabular method in traditional RL, Mnih *et al.* combined the deep neural network (DNN) with traditional Q-Learning algorithm to propose a deep Q network (DQN) model [14], [15], which is a pioneering work in the field of deep RL (DRL). Experiments show that the DQN-based agent exhibits a competitive level comparable to that of a human player in solving complex problems such as Atari 2600 games [15]. Recent years, DQN algorithm has been applied to model-free optimization and control in complex environments of power systems, such as electric vehicle charging scheduling [16], online building energy optimization [17], microgrid energy management [18], short-term voltage control [19], and adaptive power system emergency control [20]. Although the DQN algorithm can better deal with problems in high-dimensional continuous state space, its action space is still required to be discrete. In order to solve the problem in the continuous action space, Lillicrap *et al.* [21] used the idea of DQN extended Q-Learning to transform the deterministic policy gradient (DPG) [22], and proposed deep deterministic policy gradient (DDPG) algorithm. Recently, the DDPG algorithm was used to solve the joint bidding and pricing problem for a load service entity [23], and to model the strategic bidding of a market participant considering the physical non-convex operational characteristics [24]. However, our paper uses multiple DDPG-based agents to model the competition of GenCos and analyze market equilibrium.

This paper aims to address the limitations of previous methods on market equilibrium modeling. For instance, game theoretic method is generally limited to solving the NE of complete information static game. Although traditional RL algorithms can dynamically simulate repeated game of incomplete information, they are limited to low-dimensional discrete state/action space, and the convergence results are unstable. Considering all those aforementioned factors, the DDPG algorithm is used to model GenCo agents, which uses DNN to improve performance and avoid discretization of state/action space. The proposed method was used to simulate several market scenarios, including different setting of patience characteristics of GenCos, different numbers of GenCos and time-varying loads. The simulation results demonstrate the effectiveness of proposed method by comparing with prevalent game theoretic methods and traditional RL approaches.

To summarize, the main contributions of this paper are summarized as follows:

1) An electricity market simulation model based on DDPG algorithm is proposed. The employment of DNN enhances the performance of proposed model on processing high-dimensional continuous data which avoids the discretization of state/action space.
2) The accuracy and stability of agent-based modeling is

significantly improved. Experiments demonstrate that the proposed model can converge to the NE of complete information even in an incomplete information environment.

3) A method of analyzing market power has been proposed. The proposed model can accurately simulate different bidding levels by quantitatively adjusting the patience of the agent, which can be used to characterize the degree of competition in the market and analyze the potential market power.

The rest of this paper is constructed as follows. Section II describes GenCos' bidding procedure and ISO market clearing model. Section III describes two models of game theory to analyze market: static game and infinitely repeated game. Section IV presents a GenCo agent model based on DDPG algorithm, including its structure and learning scheme. Section V uses several different methods to analyze market equilibrium in a simple environment. The simulation results of proposed method in more complex market environments are described and discussed in Section VI. Section VII provides the conclusion and discusses the future work.

II. Electricity Market Structure

A. GenCos Bidding Procedure

In this paper we adopt the SFE model for GenCos' bidding procedure. The cost function of GenCo g is modeled as a quadratic function of its output power:

$$\begin{aligned} C_g^m(p_{gt}) &= \alpha_g^m p_{gt} + \frac{1}{2}\beta_g^m p_{gt}^2 \\ p_g^{\min} &\le p_{gt} \le p_g^{\max},\ \forall g \in \mathcal{G} \end{aligned} \tag{1}$$

where p_{gt} is the output power at time interval t, $p_g^{\min}$ and $p_g^{\max}$ are minimum/maximum offered power generation limits, respectively and $\mathcal{G}$ is the set of GenCos. The marginal cost of GenCo g is therefore a linear function of the output power:

$$\rho_g^m(p_{gt}) = \alpha_g^m + \beta_g^m p_{gt} \tag{2}$$

where α_g^m and β_g^m are the intercept and slope of marginal cost function, respectively.

At time interval t, each GenCo g will submit a supply offer to ISO. We use intercept-parameterization [25] to model the bidding strategy of GenCos, and the supply offer can be formulated as

$$\rho_{gt}(p_{gt}) = \alpha_{gt} + \beta_g^m p_{gt}, \alpha_{gt} \in \mathcal{A}_g \tag{3}$$

where α_{gt} is the intercept of supply function and is referred to as *strategic variable*, which is assigned by GenCo g in strategy space $\mathcal{A}_g$ and can be deviated from α_g^m to exert market power. The slope of supply function is kept equal to β_g^m.

It is worth noting that there are other alternatives to model GenCo's strategy, e.g., only the slope of supply function can be modified but the intercept is equal to α_g^m, or both the intercept and slope can be arbitrarily assigned.

B. ISO Market Clearing Model

At time interval t, the electricity consumer at load d is modeled with a linear demand curve:

$$\rho_{dt}(q_{dt}) = f_d \cdot (q_{dt} - D_{dt}^{\max}) \tag{4}$$

where f_d is the slope and does not change with time, q_{dt} is the demand quantities, $D_{dt}^{\max}$ is the maximum load demand at time interval t, and $\mathcal{D}$ is the set of loads. The total maximum load demand for all loads is

$$D_t^{\Sigma} = \sum_{d \in \mathcal{D}} D_{dt}^{\max} \tag{5}$$

Under the condition of satisfying node power balance, branch flow constraint and generator output constraint, ISO clears the market with objective of maximizing total social benefits. Market clearing at time interval t can be formulated with a DC power flow model as

$$\begin{aligned} \max_{p_{gt}, q_{dt}} & \sum_{d \in \mathcal{D}} \Big(- f_{dt} D_{dt}^{\max} q_{dt} + \frac{1}{2} f_{dt} q_{dt}^2 \Big) \\ & - \sum_{g \in \mathcal{G}} \Big(\alpha_{gt} p_{gt} + \frac{1}{2} \beta_g^m p_{gt}^2 \Big) \\ \text{s.t.} & \sum_{g \in \mathcal{G}} p_{gt} - \sum_{d \in \mathcal{D}} q_{dt} = 0 \\ & -\boldsymbol{F} \le \mathbf{PTDF}(\boldsymbol{p}_t - \boldsymbol{q}_t) \le \boldsymbol{F} \\ & p_g^{\min} \le p_{gt} \le p_g^{\max},\ \forall g \in \mathcal{G} \end{aligned} \tag{6}$$

where $\boldsymbol{p}_t$ and $\boldsymbol{q}_t$ are power generation and demand vector for all buses, which are linear combinations of p_{gt} and q_{gt}, respectively, $\mathbf{PTDF}$ is the matrix of power transfer distribution factor, and $\boldsymbol{F}$ is the vector of maximum lines flow limits.

At this time interval, the payoff of each GenCo g is

$$r_{gt} = \lambda_{it} p_{gt} - (\alpha_g^m p_{gt} + \frac{1}{2}\beta_g^m p_{gt}^2),\ g \in \mathcal{G}_i \tag{7}$$

where λ_{it} is the nodal price at bus i, which can be calculated from the Lagrange multipliers corresponding to constraints in (6), and $g \in \mathcal{G}_i$ indicates that GenCo g is located at bus i.

III. Game Theoretic Method

From the perspective of game theory, there are two types of models, static game and infinitely repeated game, that have been widely adopted in the existing literatures for analyzing electricity market. The static game model focuses on how to solve the NE, while the infinitely repeated game model focuses on the existence of tacit collusion in the market.

A. Static Games

The market opration, as described in Section II, can be seen as a static game Γ_t with a set of players (GenCos) $\mathcal{G}$, strategy space $\mathcal{A}_1, \ldots, \mathcal{A}_G$ and a vector of all GenCos' payoffs $(r_{1t}, \ldots, r_{Gt})$.

In game Γ_t, each GenCo must choose the strategic parameter α_{gt} to maximize payoff by considering the ISO market clearing problem, which is a MPEC model, which can be formulated as a

bilevel optimization problem. Considering that the ISO market clearing problem (6) is a convex quadratic programming problem, its corresponding Karush-Kuhn-Tucker (KKT) conditions are equivalent to the global optimal of the original problem. Therefore, the MPEC faced by each GenCo g can be formulated as:

$$\begin{aligned} &\max_{\alpha_{gt}} r_{gt} = \lambda_{it} p_{gt} - (\alpha_g^m p_{gt} + \frac{1}{2}\beta_g^m p_{gt}^2) \\ &\text{s.t. } \alpha_{gt} \in [0, 3\alpha_g^m] \\ &\text{the KKT conditions in (6)} \end{aligned} \tag{8}$$

Many methods can be used to solve MPEC, such as sequential quadratic programming [26], interior point method [27], branch and bound method [28], and particle swarm optimization [29]. This paper uses the sequence quadratic programming method to solve MPEC.

Simultaneous solution of all GenCos' MPECs forms an EPEC, which can be solved using diagonalization solution method: the MPEC problem faced by each GenCo is solved alternately (other GenCos' α_{gt} are fixed) until an approximation equilibrium point is found. The solution process of EPEC assumes that the information is complete, because each GenCo knows other GenCos' α_{gt} when selecting its own α_{gt}. Therefore, the solution of EPEC is the NE of the game Γ_t with complete information, which is recorded as $(\alpha_{1t}^*, \ldots, \alpha_{Gt}^*)$.

B. Infinitely Repeated Games

Repeated auctions in the electricity market can be modeled as an infinite sequence of static games $\Gamma_1, \Gamma_2, \Gamma_3, \ldots$ with discount factor γ shared by GenCos. For each GenCo g, the present value of payoff sequence $r_{g1}, r_{g2}, r_{g3}, \ldots$ is

$$r_{g1} + \gamma r_{g2} + \gamma^2 r_{g3} + \cdots = \sum_{t=1}^{\infty} \gamma^{t-1} r_{gt} \tag{9}$$

where, discount factor $\gamma \in [0, 1]$ reflects the time value of money [30]. The future payoff will be more emphasized if γ is closer to 1, which means the GenCo is more patient in the game.

If the static game Γ_t for each time interval is the same ($\Gamma = \Gamma_1 = \Gamma_2 = \Gamma_3 = \cdots$), the game sequence $\Gamma_1, \Gamma_2, \Gamma_3, \ldots$ is called an infinitely repeated game $\Gamma(\infty, \gamma)$, in which Γ is also called the stage game.

The Folk Theorem [6] suggests that, in an infinitely repeated game, when players are sufficiently patient ($\gamma \to 1$), they can earn higher payoff than in a single-stage game. In fact, any feasible cooperation of the player is the NE of infinitely repeated games with $\gamma \to 1$ [31].

Gibbons [30] analyzed the infinitely repeated Prisoner's Dilemma with the assumption that the players used grim-trigger strategy. Initially, a player using grim-trigger will cooperate (stay silent), but as soon as the opponent defects (betray), the player using grim-trigger will defect for the remainder of the repeated game. If both players adopt this grim-trigger strategy then the outcome of the infinitely repeated game will be cooperate with each other in every stage. Gibbons concludes that it is a NE for all the players to play grim-trigger strategy if and only if $\gamma \geq \underline{\gamma}$, where

$$\underline{\gamma} = \max_g \frac{r_g^d - r_g^c}{r_g^d - r_g^*} \tag{10}$$

where r_g^c is player g's payoff when both players stay silent, r_g^d is the payoff from the player g's unilateral betrayal, and r_g^* is player g's payoff when both players choose to betray.

However, in the electricity market, GenCo's action space is continuous, unlike the Prisoner's Dilemma, which has only two actions (stay silent and betray). There are also many different levels of tacit collusion in the market. So it is not easy to analytically calculate the critical value $\underline{\gamma}$. Therefore, we only consider the special case of $\gamma = 0$ when calculating the NE using the game theoretic method. $\gamma = 0$ means that the GenCos only care about the payoff of the static game at the current stage, which is also true when the static games at each stage are different. Therefore, the NE of the static game sequence $\Gamma_1, \Gamma_2, \Gamma_3, \ldots$ with $\gamma = 0$ is $(\alpha_{1,1}^*, \ldots, \alpha_{G1}^*), (\alpha_{1,2}^*, \ldots, \alpha_{G2}^*), (\alpha_{1,3}^*, \ldots, \alpha_{G3}^*), \cdots$.

IV. GenCo Agent Model

Different from the requirement of complete information in game theoretic method, the agent-based simulation method models the market as a partially observable Markov decision process (POMDP) and solves it using RL algorithm. The POMDP is described as follows: At each time interval t, each GenCo agent g receives an observable state s_{gt}, takes an action a_{gt} and receives a scalar payoff r_{gt}. Each agent g aims to maximize its own cumulative payoff $R_g = \sum_{t=1}^{T} \gamma^{t-1} r_{gt}$, where T is the total number of time intervals.

We interpret the above variables in combination with market mechanisms: 1) State s_{gt}: We take the nodal prices of previous time interval and the total load demand of current time interval as state variables:

$$s_{gt} = (\lambda_{1,t-1}, \lambda_{2,t-1}, \ldots, \lambda_{I,t-1}, D_t^{\Sigma}) \tag{10}$$

2) Action a_{gt}: This action a_{gt} is generated by the GenCo agent g, which is essentially the strategic variable α_{gt} in this paper, but their range may be different and needs to be further scaled. 3) Payoff r_{gt}: It is often referred to as *reward* in RL field. In this paper, we assume that the GenCos are rational market participants who only consider their own payoffs, so we make payoffs as rewards.

A. DDPG Algorithm

DDPG is an actor-critic, model-free algorithm based on the deterministic policy gradient that can operate in continuous state and action space [21]. The actor-critic algorithm consists of a policy function and an action-value function: the policy function acts as an actor, generating actions and interacting with the environment; the action-value function acts as a critic, which evaluates the performance of the actor and guides the follow-up of the actor.

DDPG algorithm uses DNNs to establish two approximation functions of the actor-critic algorithm: the actor network can be

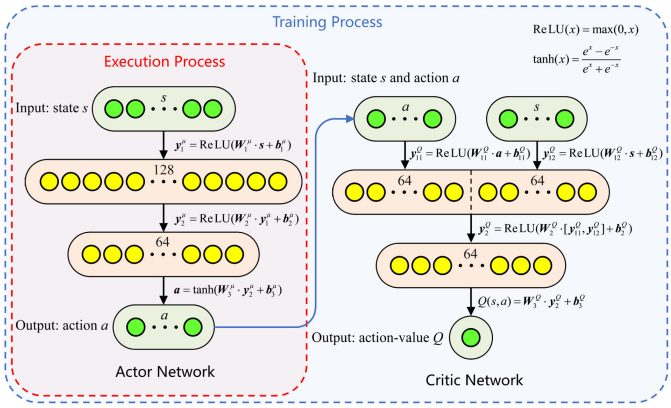


Fig. 1. Network structure of DDPG algorithm.

described as a policy function $\mu(s|\theta^\mu)$ with parameter θ^μ (abbreviated as network μ); the critic network can be described as an action-value function $Q(s,a|\theta^Q)$ with parameter θ^Q (abbreviated as network Q). The actor and critic network architecture used in this paper is shown in the Fig. 1. In order to facilitate training, those two networks are also created with a copy: actor target network μ' with parameter $\theta^{\mu'}$, critic target network Q' with parameter $\theta^{Q'}$.

The learning method of DDPG-based agent make use of the recursive relationship known as Bellman equation:

$$Q^\mu(s_t,a_t)=\mathbb{E}\left[r_t+\gamma Q^\mu(s_{t+1},\mu(s_{t+1}))\right] \tag{11}$$

The loss of the network Q is determined as follows:

$$L(\theta^Q)=\mathbb{E}_{\mu'}\left[\left(Q(s_t,a_t|\theta^Q)-y_t\right)^2\right] \tag{12}$$

where

$$y_t=r_t+\gamma Q(s_{t+1},\mu(s_{t+1})|\theta^Q) \tag{13}$$

The network μ is updated by applying the chain rule to (11) with respect to its parameters:

$$\begin{aligned}\nabla_{\theta^\mu}J&\approx\mathbb{E}_{\mu'}\left[\nabla_{\theta^\mu}Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t|\theta^\mu)}\right]\\&=\mathbb{E}_{\mu'}\left[\nabla_a Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t)}\nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s=s_t}\right]\end{aligned} \tag{14}$$

Silver *et al.* [22] proved that this is the policy gradient, which is the gradient of the policy's performance.

The training process of DDPG-based agent is as follows. Firstly, the network μ generates $\mu(s_t)$ under state s_t, and it is added with noise n_t to get action $a_t=\mu(s_t)+n_t$. After executing a_t, the agent reaches a new state s_{t+1} and gets a payoff r_t. The agent then stores the tuple (s_t,a_t,r_t,s_{t+1}) in the experience replay buffer. After that, the agent chooses N tuples in the buffer to make up a mini-batch (s_j,a_j,r_j,s_{j+1}).

With the mini-batch, the network μ' outputs action $\mu'(s_{j+1})$ to the network Q'. With the mini-batch and $\mu'(s_{j+1})$, the network Q' can calculate y_j based on (13) and input it to the network Q to calculate loss $L(\theta^Q)$ based on (12). In order to minimize the loss function, the automatic differentiation technique [32] is used to calculate $\nabla_{\theta^Q}L$, and then the parameters of the network Q are updated by gradient descent with a small learning rate lr_c:

$$\theta^Q\leftarrow\theta^Q-lr_c\nabla_{\theta^Q}L \tag{15}$$

With the mini-batch (s_j,a_j,r_j,s_{j+1}), the network μ can calculate $\mu(s_j)$ and input it to the network Q. In this process, the action a's gradient $\nabla_a Q(s,a|\theta^Q)|_{s=s_j,a=\mu(s_j)}$ and the parameter θ^μ's gradient $\nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s=s_j}$ can be calculated using the automatic differentiation technique. With those two gradients, the policy gradient in (14) can be approximated as follows:

$$\nabla_{\theta^\mu}J\approx\frac{1}{N}\sum_j\left[\nabla_a Q(s,a|\theta^Q)|_{s=s_j,a=\mu(s_j)}\nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s=s_j}\right] \tag{16}$$

In order to improve the performance of the policy, the parameters of network μ are updated by gradient ascent with a small learning rate lr_a:

$$\theta^\mu\leftarrow\theta^\mu+lr_a\nabla_{\theta^\mu}J \tag{17}$$

Finally, agent softly updates the target networks with a small update rate τ:

$$\begin{aligned}\theta^{Q'}&\leftarrow\tau\theta^Q+(1-\tau)\theta^{Q'}\\\theta^{\mu'}&\leftarrow\tau\theta^\mu+(1-\tau)\theta^{\mu'}\end{aligned} \tag{18}$$

B. Agent Setup

We use multiple DDPG-based agents to simulate the electricity market, in which each agent independently learns its own policy, treating other agents as part of the environment. Our model is very similar to the architecture of independent Q-learning (IQL) [33], but the only difference is that the RL algorithm used by each agent is not Q-learning but DDPG. Under the IQL architecture, the environment becomes nonstationary from the point of view of each agent, as it contains other agents who are themselves learning, ruling out any convergence guarantees. Although there are some methods to improve the convergence of MAS, such as centralized training [34], opponent modeling [35], and communication [36], these methods often need to obtain the actions or policies of other agents when training one agent, which does not meet market settings. We are more concerned about the market strategy of each GenCo without knowing the behavior of other GenCos, so we still use the architecture similar to IQL. In addition, substantial empirical evidence has shown that this type of architecture often works well in practice [37], and our experimental results will also prove this.

Algorithm 1 shows how to use multiple DDPG-based agents to simulate electricity market. We added the subscript g to the parameters from different agents. In this paper, we assume that strategic variable α_{gt} can be freely assigned in the range 0–3 times of α_g^m, i.e., $\mathcal{A}_g=[0,3\alpha_g^m]$. However, as shown in Fig. 1, the output of actor network a ranges from -1 to 1 (the range of $\tanh(x)$), so it needs to be scaled to the feasible range of strategic variable:

$$\alpha_{gt}=(a_{gt}+1)\cdot\frac{3}{2}\alpha_g^m \tag{19}$$

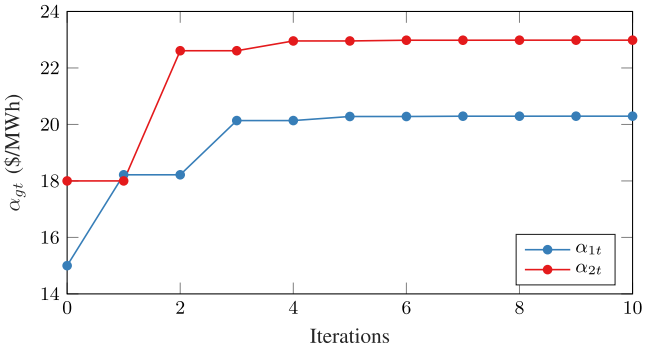


Fig. 2. The iterative curve for solving NE by game theoretic method in 3-bus system.

The default hyperparameter settings for DDPG algorithm are as follows: $T = 10000$, $lr_a = 10^{-3}$, $lr_c = 10^{-3}$, $\tau = 10^{-2}$, $N = 64$, $\gamma = 0$. The capacity of replay buffer $T_{cap} = 1000$. The training of agent begins after replay buffer is filled. We use Gaussian noise as the exploration mechanism of agent: $n_{gt} \sim \mathcal{N}(0, \sigma_t^2)$. The initial value of standard deviation σ_t is 1, but gradually decreases to 0.02 (set the lower limit to prevent too small) after the training begins:

$$\sigma_t = \begin{cases} 1, & 0 < t \le T_{cap} \\ \max(0.999^{t-T_{cap}}, 0.02), & T_{cap} < t \le T_{tra} \end{cases} \quad (20)$$

where, T_{tra} is the time to stop training, and we set $T_{tra} = 9000$. When $t > T_{tra}$, we stop training and stop adding noise to observe the original output of network (i.e., enter the test phase).

V. Market Equilibrium Under Different Methods

In this section, we numerically compare the ability of different methods to reflect market equilibrium: the game theoretic method is used to solve the NE of the static game of complete information; the multi-agent based on different RL algorithms is used to simulate the repeated game of incomplete information. These case studies are carried out on a 3-bus system [38]: the GenCo and load parameters are shown in Table I; the maximum load demand $D_{dt}^{\max}$ is assumed not to change with time; the flow limit of branch 1-2 $F_{12} = 25\text{MW}$, and the other branches are unconstrained.

A. Game Theoretic Method

Since the GenCo and load parameters in this case study are constant, the electricity auction can be seen as a infinitely repeated game $\Gamma(T, \gamma)$. We use the method described in Section III to solve the NE at $\gamma = 0$. For each stage game Γ_t, the iterative curve for solving NE is shown in Fig. 2, and the convergence result is $(\alpha_{1t}^* = 20.29, \alpha_{2t}^* = 22.98)$. Therefore, the NE of $\Gamma(\infty, 0)$ is $(\alpha_{1t}^* = 20.29, \alpha_{2t}^* = 22.98), \forall t = 1, 2, 3, \cdots$.

B. Agent-Based Modeling

In this section, the 3-bus system is simulated with different RL algorithms. The state/action space can be continuous in DDPG algorithm, but is required to be discrete in Q-Learning algorithm,

Algorithm 1: Agent-Based Modeling in Electricity Market Using DDPG Algorithm.

for agent $g = 1$ **to** G **do**
  Randomly initialize critic network $Q_g(s, a|\theta_g^Q)$ and actor $\mu_g(s|\theta_g^\mu)$ with weights θ_g^Q and θ_g^μ
  Initialize target network Q'_g and μ'_g with weights $\theta_g^{Q'} \leftarrow \theta_g^Q, \theta_g^{\mu'} \leftarrow \theta_g^\mu$
  Initialize replay buffer B_g with a capacity of T_{cap}
  Receive initial state s_{g1}
end for
for $t = 1$ **to** T **do**
  if $t \le T_{tra}$ **then**
    for agent $g = 1$ **to** G **do**
      Get noise n_{gt} from (20)
      Get the action with the network μ_g and the noise:

$$a_{gt} = \min(\max(\mu_g(s_{gt}|\theta_g^\mu) + n_{gt}, -1), 1)$$

      Get the intercept α_{gt} from (19)
    end for
    Market clearing by ISO using (6)
    for agent $g = 1$ **to** G **do**
      Get payoff r_{gt} and state $s_{g,t+1}$ from (7) and (10)
      if $t \le T_{cap}$ **then**
        Store $(s_{gt}, a_{gt}, r_{gt}, s_{g,t+1})$ in B_g
      else
        Randomly replace a tuple in B_g with $(s_{gt}, a_{gt}, r_{gt}, s_{g,t+1})$
        Randomly choose N tuples from B_g to form a mini-batch $(s_{gj}, a_{gj}, r_{gj}, s_{g,j+1}), \forall j = 1, 2, \ldots, N$
        Set $y_{gi} = r_{gi} + \gamma Q'[s_{i+1}, \mu'(s_{g,t+1}|\theta_g^\mu)]$
        Calculate the loss function

$$L(\theta_g^Q) = \frac{1}{N} \sum_j \left[Q(s_{gj}, a_{gj}|\theta_g^Q) - y_{gj}\right]^2$$

        Update the parameters of network Q_g:

$$\theta_g^Q \leftarrow \theta_g^Q - lr_c \nabla_{\theta_g^Q} L$$

        Calculate the sampled policy gradient using (16)
        Update the parameters of network μ_g:

$$\theta_g^\mu \leftarrow \theta_g^\mu + lr_a \nabla_{\theta_g^\mu} J$$

      end if
    end for
    Update target network parameters for each agent g:

$$\theta_g^{Q'} \leftarrow \tau\theta_g^Q + (1-\tau)\theta_g^{Q'} \quad \theta_g^{\mu'} \leftarrow \tau\theta_g^\mu + (1-\tau)\theta_g^{\mu'}$$

  else
    for agent $g = 1$ **to** G **do**
      Get the action with μ_g: $a_{gt} = \mu_g(s_{gt}|\theta_g^\mu)$
      Get the intercept α_{gt} from (19)
    end for
    Market clearing by ISO using (6)
  end if
end for

TABLE I
GENCO AND LOAD PARAMETERS OF 3-BUS SYSTEM

Bus	GenCo				Load		
i	g	α_g^m ($/MWh)	β_g^m ($/MWh^2)	$p_g^{\max}$ (MW)	d	f_d ($/MWh^2)	$D_d^{\max}$ (MW)
1	1	15	0.01	500	1	-0.08	500
2	—				2	-0.06	666.67
3	2	18	0.008	500	—		

TABLE II
RELATIVE ERROR RANGE AND VARIANCE IN TEN REPEATED EXPERIMENTAL RESULTS BASED ON DIFFERENT ALGORITHMS

	GenCo 1		GenCo 2	
Algorithm	Error Range	Variance	Error Range	Variance
Q-Learning	3.49∼18.27%	1.0124	1.81∼25.31%	2.1550
DDPG	-1.58∼1.56%	0.2121	-1.11∼3.09%	0.2846

so we need to make different settings for each algorithm. As described in Section IV on DDPG-based agent settings, the state variable α_{gt} contains the nodal prices and $D_{dt}^{\max}$, and the strategic parameter $\alpha_{gt} \in [0, 3\alpha_g^m]$. But in this case $D_{dt}^{\max}$ does not change with t, so D_t^{Σ} is also a constant and can be omitted: $s_{gt} = (\lambda_{1,t-1}, \lambda_{i,t-1}, \ldots, \lambda_{I,t-1})$.

For Q-Learning algorithm, state space need to be reduced in dimension and discretized. The average nodal price is used to replace all nodal price: $s_{gt} = \bar{\lambda}_{t-1} = \frac{1}{I}\sum_{i\in\mathcal{I}} \lambda_{i,t-1}$. It is further discretized: $s_{gt} \in \{0, k_s, 2k_s, 3k_s, \ldots\}$, where k_s is the step size. The strategy space also needs to be discretized: $\alpha_{gt} \in \{0, k_\alpha\alpha_g^m, 2k_\alpha\alpha_g^m, 3k_\alpha\alpha_g^m, \ldots, 3\alpha_g^m\}$, where $k_\alpha\alpha_g^m$ is the step size. The effect of step coefficients k_α and k_s on the experiment will be discussed in Section V-D, but here we temporarily set $k_\alpha = 0.1, k_s = 4$. In addition, in order to facilitate comparison with game theoretic method, we set the discount factor (γ) of Q-Learning and DDPG algorithms to 0 in this section.

We use MAS based on Q-Learning and DDPG algorithms to simulate the 3-bus system, respectively. The curves of strategic parameter α_{gt} are shown in Fig. 3. The corresponding NE $(\alpha_{1t}^*, \alpha_{2t}^*)$ of each time interval is also shown in the figure as a reference. It can be seen that the strategic parameter α_{gt} of DDPG converges to α_{gt}^* while Q-Learning does not. Notice that each agent has not been informed of any information about other agents, which reflects the ability of DDPG to converge to the NE even in an incomplete information environment.

To prevent contingency, we repeated the experiment with 10 random seeds for each algorithm. The convergence results of all strategic parameters $(\alpha_{1T}, \alpha_{2T})$ are shown in Fig. 4. The relative error range and variance of the results are shown in Table II. It can be seen that DDPG achieves 81% lower error range and 84% lower variance over Q-Learning, respectively, which proves that DDPG has higher accuracy and stability.

The code[1] for all RL algorithms is written in a python environment. For DDPG algorithm, an open source deep learning platform, Pytorch, is used to build DNNs and implement automatic differentiation. All simulations run on an Intel Core™ i5-8500 CPU, 3.00 GHz, 24.0 GB RAM. The time required to perform a simulation using Q-Learning and DDPG algorithms is 23.09 seconds and 69.98 seconds, respectively. The use of DNNs makes DDPG algorithm slower than traditional RL algorithm, but still within acceptable limits.

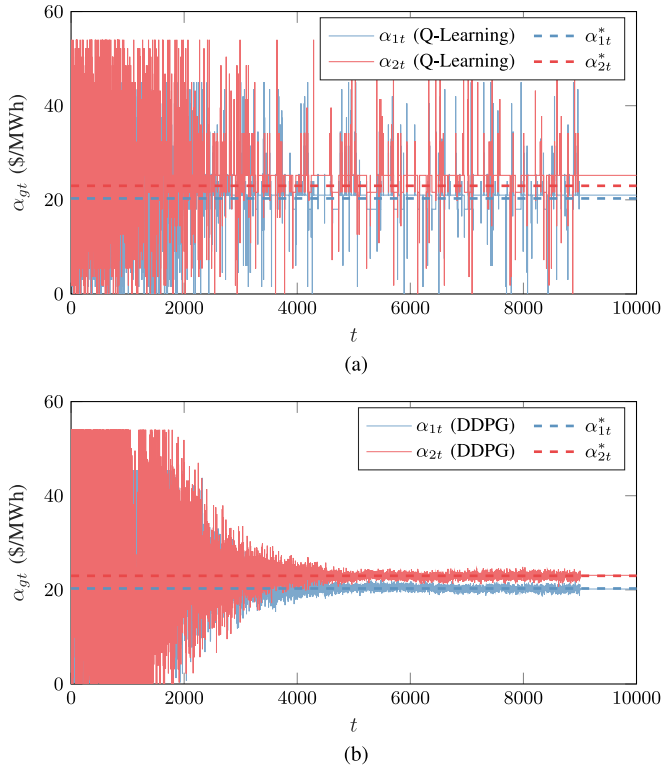


Fig. 3. Strategic variable curve under Q-Learning and DDPG algorithms.

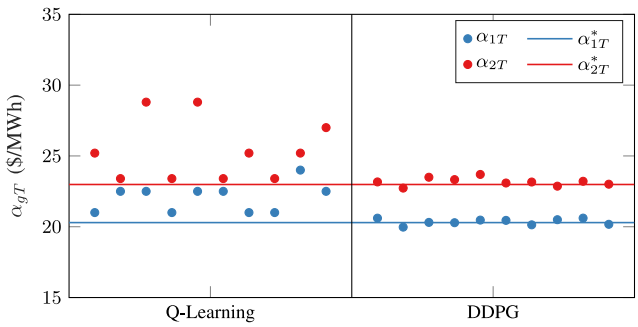


Fig. 4. The final convergence value of strategic variable (repeat 10 experiments with different random seeds for each algorithm).

C. Discount Factor and Tacit Collusion

The experiments in previous section reflect that Q-Learning algorithm tends to deviate from NE. A closer look at its convergence results in Fig. 4 reveals that they are more like tacit collusion: the average α_{1T} of Q-Learning is higher than α_{1T}^* 8.66%, and the average α_{2T} is higher than α_{2T}^* 10.43%. According to the discussion in Section III, only γ close to 1 will lead to tacit

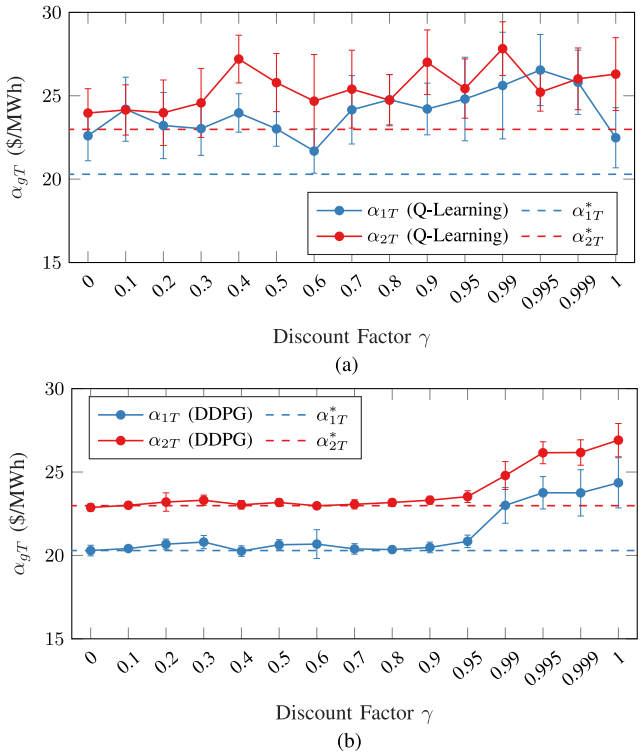


Fig. 5. Average convergence results of strategic variables with different discount factor. (Error bars are the 95% confidence intervals across 10 experiments with different random seeds.) (a) Q-Learning algorithm; (b) DDPG algorithm.

collusion; but the fact is that Q-Learning algorithm with $\gamma = 0$ has produced experimental phenomena of tacit collusion, which is inconsistent with the theory.

In fact, Athina *et al.* [39] have found similar experimental results when simulating the electricity market using SA-Q-Learning algorithm (a modified version of Q-Learning). They found that even though the discount factor was absent in their model, the experimental results were still tacit collusion. To explain this phenomenon, Athina *et al.* state that "in games of complete information, the discount factor affects the outcome of the repeated game. The analysis presented in this paper—which refers to a game of incomplete information—is indifferent about the value of the discount factor". In short, they believe that the reason for the difference from game theoretic method is that the simulation environment is incomplete information. Finally, they concluded that "discount factor is not necessary for the emergence of cooperation" in repeated games of incomplete information. However, we found different experimental results in the following experiments.

We tested Q-Learning and DDPG algorithms with different γ on the 3-bus system. For each γ, we repeated the experiment with 10 different seeds, and the average convergence result of the strategic parameters $(\alpha_{1T}, \alpha_{2T})$ are shown in Fig. 5. In addition, the corresponding stage game Γ_T's NE $(\alpha^*_{1T}, \alpha^*_{2T})$ is also shown in the figures as a reference.

As shown in Fig. 5(a), regardless of the value of γ, the strategic parameter α_{gT} of Q-Learning is higher than α^*_{gT}, but how much α_{gT} is higher than α^*_{gT} is uncertain. This reflects that

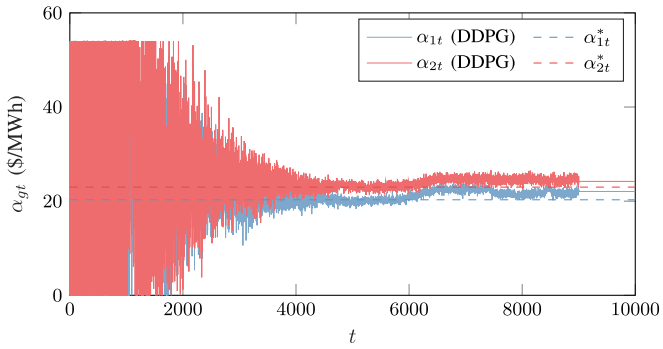


Fig. 6. Curve of strategic variable with discount factor $\gamma = 0.999$.

γ has almost no effect on the convergence of Q-Learning, i.e., regardless of the value of γ, Q-Learning will only converge to tacit collusion.

However, the experimental results of DDPG are different. It can be seen from Fig. 5(b) that the strategic parameters α_{gT} of DDPG can converge to α^*_{gT} stably when γ is substantially less than 0.95 ($\underline{\gamma} \approx 0.95$), but as the γ approaches 1, the strategic parameter α_{gT} is tacitly increased to achieve higher payoffs. Therefore, the experimental results of DDPG are consistent with the game theoretic method, and it can intuitively reflect the characteristics of tacit collusion, such as how much γ can cause tacit collusion, and what is the level of tacit collusion (distribution of strategic parameters) under different γ.

A typical convergence curve when $\gamma = 0.999$ is shown in Fig. 6. It can be seen that the convergence curve of $\gamma = 0.999$ is unstable compared to Fig. 4(c) of $\gamma = 0$. Fig. 5(b) also shows that the confidence interval with $\gamma \to 1$ is large. This instability can be explained by two reasons: 1) as Folk Theorem proves, there is no fixed NE when $\gamma \to 1$, but many different levels of collusion are all NE; 2) a larger γ makes the agent pay more attention to the future payoff, which makes the action-value instability increase and more difficult to estimate. It is worth noting that in this case with many NEs, each run of DDPG algorithm may only converge to one NE. By repeating the experiment with different random seeds, a representative NE distribution range can be obtained (e.g. Fig. 5(b)).

The experiment also reflects the tacit collusion of Q-Learning is caused by its suboptimal performance. For example, when $\gamma = 0$, rational agents should only consider the payoff of the current stage game and choose the optimal strategy to converge to the NE of the stage game. However, Q-Learning tends to converge to a suboptimal strategy because of its poor ability to perceive the environment, resulting in an irregular and γ-independent collusion. Thus given $\gamma = 0$, the accuracy of MAS to converge to the NE of the corresponding stage game clearly characterizes its performance.

D. Sensitivity Analysis

We performed a sensitivity analysis of DDPG algorithm based on the 3-bus system. As mentioned before, the default network structure is shown in Fig. 1, and the default parameters are: batch size $N = 64$, learning rate $lr_a = lr_c = 10^{-3}$, soft update rate

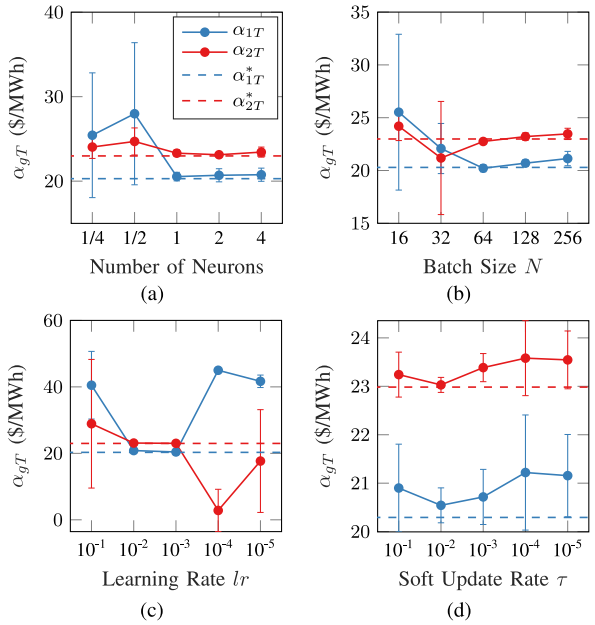


Fig. 7. Convergence of strategic variables under different hyperparameters. (Error bars are the 95% confidence intervals across 10 experiments with different random seeds.) (a) Number of neurons; (b) Batch size N; (c) Learning rate lr; (d) Soft update rate τ.

$\tau = 10^{-2}$, and discount factor $\gamma = 0$. lr_a and lr_c are generally not much different, so we set them to the same value as lr. We redo the experiment by changing the number of neurons in each layer of the default network to 1/4, 1/2, 1, 2, and 4 times and keeping the other parameters unchanged. The convergence result is shown in Fig. 7(a). Similarly, the experimental results of changing N, lr, and τ, respectively, are shown in Fig. 7(b)-(d).

As shown in Fig. 7, reducing the number of neurons per layer or batch size N from the default settings may have a greater impact on the performance of DDPG algorithm, but it is not obvious to increase them. The performance of DDPG is sensitive to the learning rate lr, and the suitable range is $10^{-3} \sim 10^{-2}$. The change of the soft update rate τ has little effect on the algorithm, and the value is preferably 10^{-2}.

In addition, we analyze the influence of step coefficients k_α and k_s on Q-Learning algorithm. In order to compare the performance of Q-Learning with different k_α and k_s, let it compete with a fixed opponent and observe its payoff. This fixed opponent is a DDPG-based agent with default parameters. These two agents compete on the 3-bus system, but the system is adjusted as follows to make their environment completely symmetrical: the cost parameters of the two GenCos are set to be the same ($\alpha_1^m = \alpha_2^m = 15, \beta_1^m = \beta_2^m = 0.01$), and the branch flow constraints are ignored so that all nodal prices are always the same. We repeated the experiment with 10 random seeds for each (k_α, k_s) and the average profit is shown in Fig. 8.

As can be seen from Fig. 8, the payoff with smaller k_α and k_s is lower and the confidence interval is wider, but some payoffs with larger steps are also low. This is because there are too many states or actions in small steps, which are difficult to fully explore and evaluate, but larger step sizes may increase the deviation from the optimal solution. Therefore, we try to choose a step size that has a small value and can lead to high profits, such as $k_\alpha = 0.1, k_s = 4$. Another rule that can be derived from the figure is that in any case the payoff of DDPG-based agent is higher than that of Q-Learning, which proves that DDPG algorithm has better performance.

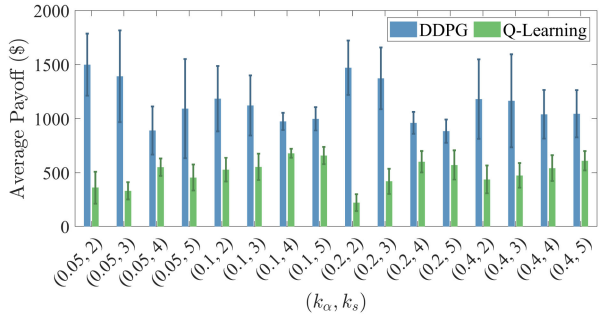


Fig. 8. Average payoff of Q-Learning against DDPG in different discrete spaces. (Error bars are the 95% confidence intervals across 10 experiments with different random seeds).

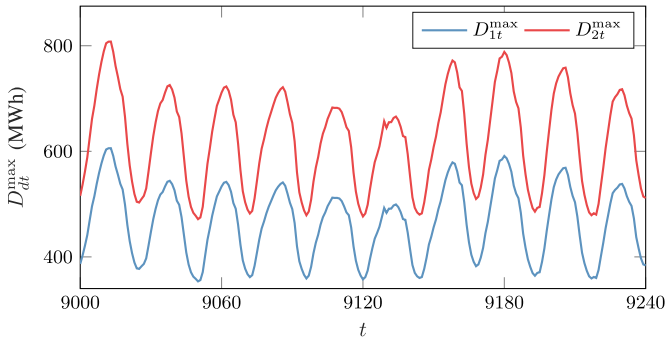


Fig. 9. Load curve in the 3-bus system.

VI. Simulation Cases in Complex Environments

To verify the effectiveness of proposed method in more complex market environments, we tested the case of time-varying loads, different levels of network congestion and an increase in the quantity of GenCos.

A. Performance With Time-Varying Load

To illustrate the capability of proposed method in a more complex scenario, the DDPG-based agents is simulated with time-varying load to test whether NE can be achieved for each time interval. The simulation case is still based on the 3-bus system in Table I. As shown in Fig. 9, the perviously fixed demand is replaced by a time-varing load curve obtained from PJM [40]. We simulated the market with $\gamma = 0$, $\gamma = 0.5$ and $\gamma = 0.999$, respectively. The curves of strategic variable α_{gt} in a certain period of time are shown in Fig. 10.

In addition, the market auction in this case can be regarded as a static game sequence $\Gamma_1, \Gamma_2, \Gamma_3, \cdots$. The load that varies with t makes each static game Γ_t no longer the same, so we need to find the NE $(\alpha^*_{1t}, \ldots, \alpha^*_{6t})$ for each Γ_t separately. The variation curve of α^*_{gt} is also shown in Fig. 10 as a reference.

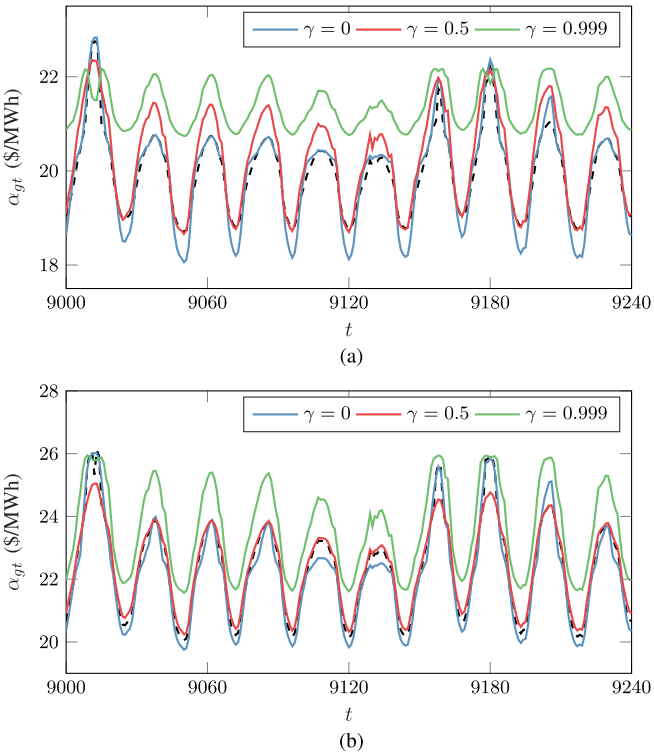


Fig. 10. Strategic variable curves with different γ under time-varying load. (The black dotted line is the strategic variable α^*_{gt} in the NE found by the game theoretic method.) (a) GenCo 1; (b) GenCo 2.

As can be seen from Fig. 10, the strategic variable α_{gt} can approximately converge to α^*_{gt} when $\gamma = 0$ and $\gamma = 0.5$, which reflects the ability of DDPG to converge to NE even under time-varying load. However, when $\gamma = 0.999$, the strategic parameter α_{gt} is significantly raised. This shows that when $\gamma \to 1$, the tacit collusion is also likely to be the NE of static game sequence $\Gamma_1, \Gamma_2, \Gamma_3, \cdots$ with different Γ_t, which extends the applicable conditions of Folk Theorem (the applicable condition of Folk Theorem is $\Gamma_1 = \Gamma_2 = \Gamma_3 = \cdots$).

B. Performance With Network Congestion

We test the impact of different levels of network congestion on the proposed model. The experiment is still based on a 3-bus system, where the power limit of branch 1-2 is recorded as F_{12}, and the other branches are unconstrained. The convergence results of DDPG with different F_{12} are shown in Fig. 11. It can be seen that the convergence results of DDPG under different network congestion levels are still consistent with the game theoretic method. In addition, with the increase of F_{12}, GenCos' strategic variables are generally reduced, and the market is more competitive.

C. Performance With More GenCos

In this section, we test the performance of DDPG algorithm in IEEE 30-bus system with 6 GenCos. GenCo and load parameters [28] are shown in Table III and Table IV, network parameters are shown in [41]. It is assumed that the flow limits of branches

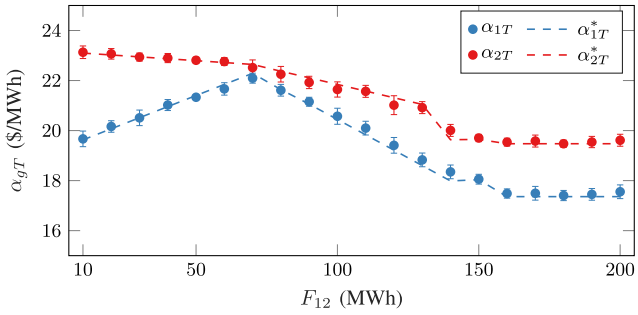


Fig. 11. Average convergence results of strategic variables with different network congestion levels. (Error bars are the 95% confidence intervals across 10 experiments with different random seeds).

TABLE III
GENCO PARAMETERS OF IEEE 30-BUS SYSTEM

g	Bus	α_g^m ($/MWh)	β_g^m ($/MWh^2)	$p_g^{\min}$ (MW)	$p_g^{\max}$ (MW)
1	1	18	0.25	5	100
2	2	20	0.20	5	80
3	13	25	0.20	5	50
4	22	22	0.20	5	80
5	23	22	0.20	5	50
6	27	16	0.25	5	120

TABLE IV
LOAD PARAMETERS OF IEEE 30-BUS SYSTEM

d	Bus	f_d ($/MWh^2)	$D_d^{\max}$ (MWh)	d	Bus	f_d ($/MWh^2)	$D_d^{\max}$ (MWh)
1	2	−5.0	24.0	11	17	−3.5	25.7
2	3	−5.5	23.6	12	18	−3.5	27.1
3	4	−4.5	26.7	13	19	−3.5	25.7
4	7	−5.0	27.0	14	20	−3.5	25.7
5	8	−5.0	30.0	15	21	−6.0	26.7
6	10	−3.0	31.7	16	23	−5.0	24.0
7	12	−5.5	27.3	17	24	−6.0	25.0
8	14	−4.0	31.3	18	26	−4.5	22.2
9	15	−4.5	22.2	19	29	−3.5	27.1
10	16	−5.0	30.0	20	30	−4.5	27.8

6–8, 12–17, and 10–17 are 10, 8, and 10 MW, respectively, and the other branches are unconstrained [28].

The NE in static game of complete information is still used as a reference. The convergence curve for solving NE using the game theoretic method is shown in Fig. 12 and the convergence result is $(\alpha^*_{1t} = 21.39, \alpha^*_{2t} = 23.81, \alpha^*_{3t} = 34.32, \alpha^*_{4t} = 27.24, \alpha^*_{5t}$ is divergent, $\alpha^*_{6t} = 24.85), \forall t = 1, 2, 3, \ldots,$ which is consistent with the result obtained by the branch and bound method in [28].

We tested the DDPG-based MAS with different γ on IEEE 30-bus system. For each γ, we experimented 10 times with different random seeds, and the average value of the convergence result α_{gT} is shown in Fig. 13. The convergence curve when $\gamma = 0.5$ and $\gamma = 0.999$ is shown in Fig. 14.

As can be seen from Fig. 13-14, most of the agents still satisfy the following rules: When γ is small, such as $\gamma < 0.8$, they will

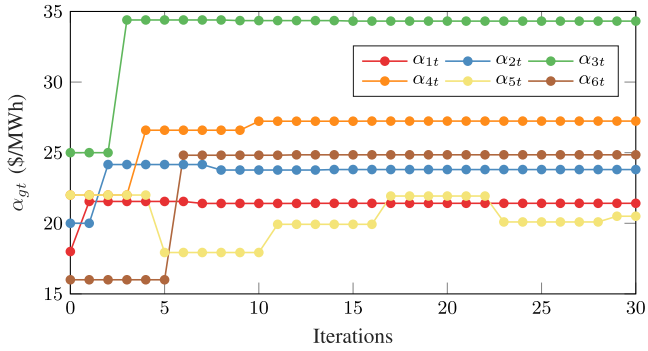


Fig. 12. The iterative curve for solving NE by game theoretic method in 30-bus system.

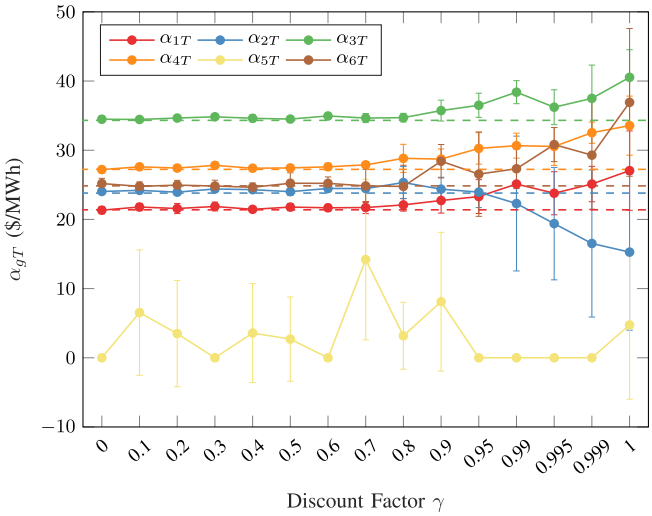


Fig. 13. Average convergence results of strategic variables with different discount factor. (Error bars are the 95% confidence intervals across 10 experiments with different random seeds. The dotted line is the strategic variable α^*_{gT} in the NE found by the game theoretic method).

converge to α^*_{gT}; as γ becomes closer to 1, the behavior of tacitly raising prices becomes more and more obvious. However, there are exceptions: as γ increases, α_{2T} decreases and diverges; α_{5T} is small and divergent in any case. In order to more intuitively show why these parameters are small and divergent, we keep other GenCos' α_{gT} unchanged, constantly adjust α_{2T} (or α_{5T}) and calculate the payoff r_{2T} (or r_{5T}). The results are shown in Fig. 15. It can be seen that when $\gamma = 0.5$, r_{2T} has only one maximum value; But when $\gamma = 0.999$, r_{2T} can reach the maximum value within a certain range; In addition, regardless of $\gamma = 0.5$ or 0.999, r_{5T} can reach the maximum value within a certain range. The market power of a GenCo is affected by many factors such as cost parameters, branch flow constraints, and other GenCos' bidding parameters. When a GenCo's market power is too small to affect the nodal price, it tends to reduce the bidding parameter to get more market-clearing quantities to increase profits. The experiment also shows that GenCo 1, 3, 4 and 5 have strong market power at different levels of competition.

It can be seen that the proposed method can reflect the trading strategy of different GenCo under different levels of

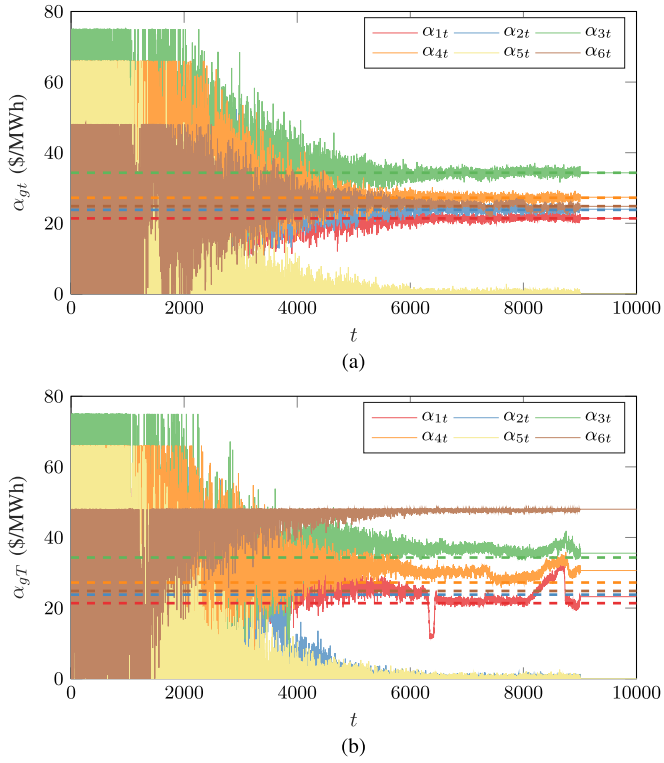


Fig. 14. Strategic variable curve with different γ. (a) $\gamma = 0.5$; (b) $\gamma = 0.999$.

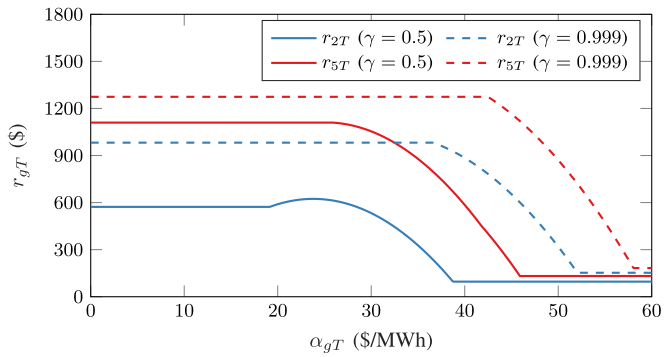


Fig. 15. GenCo's payoff curve with its own strategic variables.

competition. In addition, $\underline{\gamma} \approx 0.8$ in the current study case and $\underline{\gamma} \approx 0.95$ in the 3-bus system, which means that the 3-bus system is more competitive. The ability of proposed method to approximate the calculation of $\underline{\gamma}$ and the ability to reflect the level of tacit collusion (the distribution interval of strategic variable) is also meaningful for the analysis of strategic behaviors, such as determining which market design is more competitive.

We did the following experiments to verify the stability of proposed model as the number of GenCos increased. In the IEEE 30-bus system, we divide each GenCo into c small GenCos on average. The cost parameters of each small GenCo (i.e., α_g^m, β_g^m) remain unchanged, but the minimum and maximum power generation $(p_g^{\min}, p_g^{\max})$ become $1/c$ times the original. We perform experiments on $c = 1, 2, 3$, and 4 (i.e., the total number of GenCos is 6, 12, 18, and 24), respectively, and the results are shown in Fig. 16.

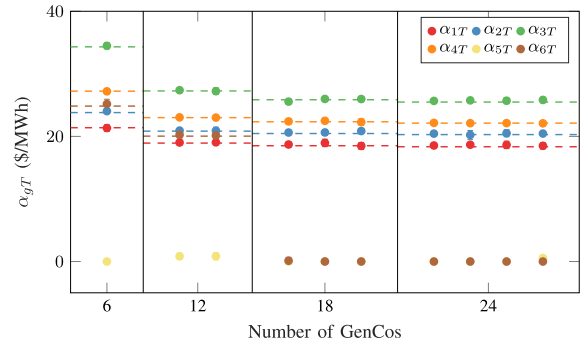


Fig. 16. Average convergence results of strategic variables with different numbers of GenCos. (Error bars are the 95% confidence intervals across 10 experiments with different random seeds. The dotted line is the strategic variable α^*_{gT} in the NE found by the game theoretic method).

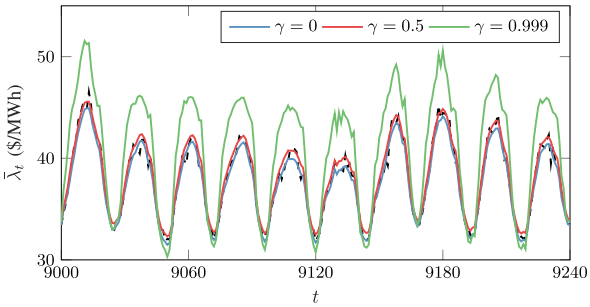


Fig. 17. Average nodal price curves with different γ under time-varying load. (The black dotted line is the average nodal price λ^*_{gt} in the NE found by the game theoretic method).

It can be seen that even if the number of GenCos increases, the experimental results of DDPG can still converge to the NE obtained by game theoretic method. In addition, as the number of GenCos increases, the price of the corresponding NE gradually decreases, and the competitiveness of the market is improved. Especially when the number was increased from 6 to 12, the strategic variables of GenCos decreased greatly, but when it increased from 18 to 24, the strategic variables decreased slightly.

We next simulate the market with time-varying loads on the 30-bus system with 6 GenCos. The hourly load data ($D^{\max}_{dt}$) is obtained from PJM [40] and scaled by an appropriate ratio. Considering the large number of agents, we use the average nodal price $\bar{\lambda}_t$ to reflect the market clearing results. For each time interval t, we also use the game theoretic method to calculate the NE of corresponding stage game Γ_t, in which the average nodal price is recorded as $\bar{\lambda}^*_t$. The curves of $\bar{\lambda}_t$ with different γ and $\bar{\lambda}^*_t$ are as shown in Fig. 17. The results still reflect the ability of the proposed method to converge to the NE under time-varying loads.

VII. Conclusion

In this paper, a market simulation model based on the DDPG algorithm is proposed and tested on a 3-bus system and IEEE 30-bus system. The experimental results show that the proposed model can converge to the NE of complete information even in the incomplete information environment. The convergence result is NE of the corresponding stage game when the discount factor is small. In contrast, the result become tacit collusion when the discount factor is close enough to 1. The proposed model can reflect the different levels of competition and find the critical value that triggers tacit collusion by quantitatively adjusting the discount factor, which can be an effective means to analyze strategic behavior of market participants.

An open problem with current multi-agent DRL methods is the lack of theoretical understanding of their convergence properties. Nevertheless, our experimental results show that the proposed method has good convergence in general. If only NE needs to be solved, the game theoretic method with complete information is still preferred. Our proposed method can be regarded as a supplement to the game theoretic method, which is used to analyze the GenCos' strategies under incomplete information.

Future work mainly includes the following three directions. The first is to apply the proposed method to compare the level of competition under different market models. The second is to enhance the functionality of the GenCo agent, such as participating in multi-level power markets, implementing load forecasting, self-scheduling and risk management. The third is to extend the model to other types of market participants, such as retailers and consumers.

References

[1] T. Dai and W. Qiao, "Finding equilibria in the pool-based electricity market with strategic wind power producers and network constraints," *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 389–399, Jan. 2017.

[2] Z. Younes and M. Ilic, "Generation strategies for gaming transmission constraints. Will the deregulated electric power market be an oligopoly?" in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 1998, vol. 3, pp. 112–121.

[3] X. Zhang, *Restructured Electric Power Systems: Analysis of Electricity Markets With Equilibrium Models*. Hoboken, NJ, USA: Wiley, 2010.

[4] A. Sweeting, "Market power in the england and wales wholesale electricity," *Econ. J.*, vol. 117, no. 520, pp. 654–685, 2010.

[5] J. Joseph E. Harrington, "A theory of tacit collusion," Department of Economics, The Johns Hopkins University, Baltimore, MD, USA, Tech. Rep. 588, 2012.

[6] J. W. Friedman, "A non-cooperative equilibrium for supergames," *Rev. Econ. Stud.*, vol. 38, no. 1, pp. 1–12, 1971.

[7] A. E. Roth and I. Erev, "Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term," *Game. Econ. Behav.*, vol. 8, no. 1, pp. 164–212, 1995.

[8] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3-4, pp. 279–292, 1992.

[9] J. Nicolaisen, V. Petrov, and L. Tesfatsion, "Market power and efficiency in a computational electricity market with discriminatory double-auction pricing," *IEEE Trans. Evol. Comput.*, vol. 5, no. 5, pp. 504–523, Oct. 2001.

[10] W. Yeping, J. Zhaoxia, C. Haoyong, H. Ngan, and Z. Yao, "A method for designing agent based electricity market simulation experiments," *Automat. Electric Power Syst.*, vol. 33, no. 17, pp. 56–61, 2009.

[11] A. Somani and L. Tesfatsion, "An agent-based test bed study of wholesale power market performance measures," *IEEE Comput. Intell. Mag.*, vol. 3, no. 4, pp. 56–72, Nov. 2008.

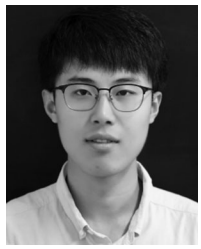
[12] H. Chen, Y. Yang, Y. Zhang, Y. Wang, Z. Jing, and Q. Chen, "Realization of decision-making module in agent-based simulation of power markets," *Autom. Electr. Power Syst.*, vol. 32, no. 20, pp. 22–26, 2008.

[13] N. Yu, C. Liu, and J. Price, "Evaluation of market rules using a multi-agent system method," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 470–479, Feb. 2010.

[14] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[15] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[16] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.
[17] E. Mocanu *et al.*, "On-line building energy optimization using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3698–3708, Jul. 2019.
[18] Y. Ji, J. Wang, J. Xu, X. Fang, and H. Zhang, "Real-time energy management of a microgrid using deep reinforcement learning," *Energies*, vol. 12, no. 12, 2019, Art. no. 2291.
[19] J. Zhang, C. Lu, J. Si, J. Song, and Y. Su, "Deep reinforcement learning for short-term voltage control by dynamic load shedding in china southem power grid," in *Proc. Int. Jt. Conf. Neural Netw.*, Jul. 2018, pp. 1–8.
[20] Q. Huang, R. Huang, W. Hao, J. Tan, R. Fan, and Z. Huang, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1171–1182, Mar. 2020.
[21] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[22] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 387–395.
[23] H. Xu, H. Sun, D. Nikovski, S. Kitamura, K. Mori, and H. Hashimoto, "Deep reinforcement learning for joint bidding and pricing of load serving entity," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6366–6375, Nov. 2019.
[24] Y. Ye, D. Qiu, M. Sun, D. Papadaskalopoulos, and G. Strbac, "Deep reinforcement learning for strategic bidding in electricity markets," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1343–1355, Mar. 2020.
[25] A. Banal-Estanol and A. R. Micola, "Behavioural simulations in spot electricity markets," *Eur. J. Oper. Res.*, vol. 214, no. 1, pp. 147–159, 2011.
[26] R. Fletcher and S. Leyffer, *Numerical Experience with Solving MPECS and NLPs*. Dundee, U.K.: Univ. Dundee Press, 2002.
[27] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.
[28] J. Yang and Z. Yan, "Branch and bound approach to the solution of the linear supply function equilibrium model in presence of network constraints," *Proc. Chin. Soc. Elect. Eng.*, vol. 30, no. 13, pp. 94–100, 2010.
[29] Y. Ma, C. Jiang, Z. Hou, and C. Wang, "The formulation of the optimal strategies for the electricity producers based on the particle swarm optimization algorithm," *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1663–1671, Nov. 2006.
[30] R. Gibbons, *A Primer in Game Theory*. Hemel Hempstead, U.K.: Harvester Wheatsheaf, 1992.
[31] J. Levin, "Bargaining and repeated games," 2002. [Online]. Available: http://web.stanford.edu/∼jdlevin/Econ%20203/RepeatedGames.pdf
[32] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, no. 153, pp. 1–43, 2018.
[33] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learn. (ICML)*, 1993, pp. 330–337.
[34] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Proces. Syst.*, 2017, pp. 6379–6390.
[35] R. Raileanu, E. Denton, A. Szlam, and R. Fergus, "Modeling others using oneself in multi-agent reinforcement learning," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, vol. 80, pp. 4257–4266.
[36] A. Singh, T. Jain, and S. Sukhbaatar, "Learning when to communicate at scale in multiagent cooperative and competitive tasks," in *Proc. 7th Int. Conf. Learn. Representations*, 2019.
[37] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Independent reinforcement learners in cooperative markov games: A survey regarding coordination problems," *Knowl. Eng. Rev.*, vol. 27, no. 1, pp. 1–31, 2012.
[38] W. Xian, L. Yuzeng, and Z. Shaohua, "Oligopolistic equilibrium analysis for electricity markets: a nonlinear complementarity approach," *IEEE Trans. Power Syst.*, vol. 19, no. 3, pp. 1348–1355, Aug. 2004.
[39] A. C. Tellidou and A. G. Bakirtzis, "Agent-based analysis of capacity withholding and tacit collusion in electricity markets," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1735–1742, Nov. 2007.
[40] "PJM market data," [Online]. Available: https://www.pjm.com/
[41] E. Bompard, W. Lu, and R. Napoli, "Network constraint impacts on the competitive electricity markets under supply-side strategic bidding," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 160–170, Feb. 2006.



Yanchang Liang (Student Member, IEEE) received the B.S. degree from College of Electrical Engineering, North China Electric Power University, Baoding, China, in 2018. He is currently working toward the M.S. degree at North China Electric Power University, Beijing, China.

His current research interests include control, optimization, reinforcement learning, with applications to power systems and electric transportation systems.

Chunlin Guo received the B.S. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1998 and 2003.

He is currently a Professor with the North China Electric Power University, Beijing, China. His research interests include electric vehicle charging technology, analysis and control of FACTS, and power quality control.

Zhaohao Ding (Member, IEEE) received the B.S. degree in electrical engineering and the B.A. degree in finance both from Shandong University, Jinan, China, in 2010, and the Ph.D. degree in electrical engineering from the University of Texas at Arlington, Arlington, TX, USA, in 2015.

He is currently an Associate Professor with North China Electric Power University, Beijing, China. His research interests include power system planning and operation, power market, and electric transportation system.

Huichun Hua received the M.S. degree in computational mathematics from Northwestern Polytechnical University, Xi'an, China, in 2004, and the Ph.D. degree in electrical engineering from the North China Electric Power University, Beijing, China, in 2014.

He is currently an Assistant Professor with the North China Electric Power University, Baoding, China. His research interests include power market and power quality analysis.