Agent-Based Analysis of Monopoly Power in Electricity Markets

A. C. Tellidou, and A. G. Bakirtzis, Senior Member, IEEE

Abstract—In this paper agent-based simulation is employed to study the energy market performance and, particularly, the exercise of monopoly power. The energy market is formulated as a stochastic game, where each stage game corresponds to an hourly energy auction. Each hourly energy auction is cleared using Locational Marginal Pricing. Generators are modeled as adaptive agents capable of learning through the interaction with their environment, following a Reinforcement Learning algorithm. The SA-Q-learning algorithm, a modified version of the popular Q-Learning, is used. Test results on a two-node power system with two competing generator-agents, demonstrate the exercise of monopoly power.

Index Terms—Electricity Markets, Capacity Withholding, Monopoly Power, Reinforcement Learning, Stochastic Games.

I. INTRODUCTION

MARKET power is one of the central topics in economics and, consequently, in electricity market design. Market power is “the ability to alter profitably prices away from competitive levels” [1], [2]. “Monopoly power is the market power of sellers, who want higher prices, as opposed to buyers, who want lower prices”, as stated in [1]. “Market power is static and exercised through a unilateral decision (i.e. a single firm which takes rival choices as given) either to withdraw capacity or to raise price” according to [3]. Hence, there are two different paths for a generator to exercise monopoly power; he can either withhold output or he can ask a higher price than his marginal cost.

As mentioned in [3], even a firm with a small market share can affect market price, as long as it is “pivotal either in a particular location or in reference to a system reliability standard”. This argument is also supported by Rothkopf in [4], where he states that a pivotal seller “by any reasonable definition has extreme market power” and he also adds “this is true whatever its market share”.

According to [5], market power policy in electricity sector is a priority concern mostly for regulators, but also for market participants. The peculiar properties of the commodity called electricity make difficult the application of even the most sophisticated theoretical models. Hence, one of the most challenging tasks in power market monitoring is to identify who exercises market power and how. The answers to these questions have important implications to power market design; therefore, many researchers have tried to approach this demanding issue in a satisfying way.

In [6] Cramton deals with market design of electricity markets, focusing on FERC’s (Federal Energy Regulatory Council) Standard Market Design. He outlines some basic principles, but mainly he discusses the important issue of market power. In [7] a method is described, which attempts to give answer to the aforementioned questions. The authors introduce a practical approach to examine market power based on revenue sensitivities; they identify the suppliers who are able to increase their revenues by either raising prices or withholding capacity. In [3], the market power definition and the abuse of dominant position are discussed under the terms of the UK Pool and the proposed New Electricity Trading Arrangements (NETA). In [4], the exercise of market power in electricity auctions under different circumstances is discussed and a remedy for the high prices is proposed; complications as well as variations of this proposal are also studied. Oren has also studied the inefficiencies in electricity markets that result from the capacity withholding, in case of a congested network, using a game theoretic approach [8].

An alternative to the game theory, used to study the behavior of market participants, is agent-based simulation (ABS) [9], [10], [11] and [12]. In [10], the market participants develop their profit-maximizing strategy based on a naive reinforcement learning algorithm, while in [11] and [12], Q-learning algorithm is used. Q-learning [13] is among the most popular reinforcement learning algorithms, owing to its simplicity. [11] suggests a bidding strategy for suppliers in electricity market; different values in the parameters of the algorithm are used to demonstrate different participants profiles. In [12] an agent-based study is presented, where the electricity auction is simulated as a repeated game. The analysis concludes that agents following a Q-learning strategy can learn to behave as game theory suggests.

In this paper, the power market operation is formulated as a stochastic game, with each stage being an hourly auction; Q-learning algorithm is used to model the bidding strategy of generators, but the state of their environment is fully defined by their rivals’ actions. As far as it concerns the policy, an approach based on the Metropolis criterion of Simulated Annealing is used. Our purpose is to study players’ behavior in a network constrained market. In section II the spot electricity market structure is presented. In section III the stochastic games are introduced and their analogies to the electricity market are discussed. In section IV a description of
the SA-Q-learning algorithm is presented and the generator’s behavior is modeled. Finally, section V reports the results of our test case.

II. ELECTRICITY MARKET STRUCTURE

In this paper the behavior of generators in a spot market with hourly trading interval is studied. Generators submit Energy Offers to the Independent System Operator (ISO), declaring the power they are willing to sell at or above a certain price. Each generator, \( g \in \mathcal{G} \), with net capacity \( P_g^{\text{max}} \) and marginal cost, \( \text{mc}_g \), offers a certain amount of power \( P_g \) in MW, \( 0 \leq P_g \leq P_g^{\text{max}} \), at a constant price \( b_g \) in €/MWh, which cannot be higher than the market price cap, \( pc \), or lower than the generator’s marginal cost, \( \text{mc}_g \leq b_g \leq pc \).

A. The ISO market clearing problem

The ISO collects the energy offers \((P_g,b_g)\) of all generators, \( g \in \mathcal{G} \), and based on the most recent forecast of the nodal demands, \( d \), computes the quantities, \( P_g, \forall g \in \mathcal{G} \), and nodal prices, \( \text{LMP}_k, \forall k \in \mathcal{K} \), that clear the market, by solving the following optimal power flow (OPF) problem:

\[
\text{Min} \sum_{g \in \mathcal{G}} b_g \cdot P_g
\]

Subject to:

\[
\mathbf{B} \cdot \mathbf{\theta} = \mathbf{H}_{KG} \mathbf{p} - \mathbf{H}_{KF} \mathbf{d}
\]

\[
\theta_{\text{ref}} = 0
\]

\[
\frac{l}{x_{km}} (\theta_k - \theta_m) \leq F_{km}^{\text{max}} \quad \text{for all lines km}
\]

\[
0 \leq P_g \leq P_g^{\text{max}} \quad \text{for all generators } g \in \mathcal{G}
\]

where:

- \( \mathbf{p} \) generator active power output vector
- \( d \) consumer active power demand vector
- \( \mathbf{\theta} \) bus voltage phase angle vector
- \( \theta_{\text{ref}} \) reference bus voltage phase angle
- \( x_{km} \) reactance of line \( km \)
- \( F_{km}^{\text{max}} \) transmission capacity limit of line \( km \)
- \( \mathbf{B} \) network admittance matrix
- \( \mathbf{H}_{KG} \) bus to generator incidence matrix (size \( K \cdot G \))
- \( \mathbf{H}_{KF} \) bus to consumer incidence matrix (size \( K \cdot F \))

Constraints (2) represent the system DC power flow equations, (3) defines the slack bus voltage phase angle since \( \text{det}(\mathbf{B}) = 0 \)\(^1\), (4) represent the transmission line power flow limits, and (5) represent the unit active power output limits. The Lagrange multipliers of the nodal active power balance constraints (2) are the nodal prices (LMPs).

B. The Generator profit maximization problem

The Generator’s objective is to maximize his profits in the spot market, by selecting the parameters of his energy offer, \((P_g,b_g)\).

\[
\text{Max Profit}_g = (\text{LMP}_{g(k)} - \text{mc}_g) \cdot P_g
\]

Subject to the following constraints:

\[
0 \leq P_g \leq P_g^{\text{max}}
\]

\[
\text{mc}_g \leq b_g \leq pc
\]

as well as the ISO market clearing problem solution (1)-(5), needed to define \( P_g \) and \( \text{LMP}_{g(k)} \) used in (6).

However, the generator does not have the information on the transmission network, the consumer demand and the competitor Energy Offers that the ISO has when solving the market-clearing problem. Section IV describes a reinforcement learning process by which the generator can “learn” through the repetition of the hourly energy auction to select the profit maximizing parameters of his Energy Offer, \((P_g,b_g)\), based only on publicly available information (LMPs and previous Energy Offers).

III. STOCHASTIC GAMES

A. Static Games

Let us assume the following \( n \)-player simple game:

1. Players 1 through \( n \) simultaneously choose actions \( a_i \in A_i \) through \( a_n \in A_n \), respectively.
2. They receive their payoffs \( u_i(a_1,...,a_n) \) through \( u_n(a_1,...,a_n) \).

\( \Gamma = \{ A_1,...,A_n : u_1,...,u_n \} \) denotes the above static game [14].

According to the above, the market operation, as described in section II, can be seen as a stage game, where \( \mathcal{G} \) is the set of players; then each generator can be regarded as player and his action space consists of all the possible selections of his Energy Offer, \((P_g,b_g)\).

B. Stochastic Games

An \((n\text{-player})\) or \((n\text{-agent})\) stochastic game (SG) is a tuple \( (\mathcal{N},S,\mathcal{A},\bar{u},T) \). \( \mathcal{N} \) is the set of agents indexed \( 1,...,n \); \( S \) is the set of \( n\text{-agent} \) stage games; \( \bar{A} = A_1,...,A_n \) has already been defined (we assume the agent has the same strategy space

\(^1\) the slack bus is not omitted in (2)
in all games; this is a notational convenience, but not a substantive restriction), as well as \( \tilde{a} = u_1, \ldots, u_n \) with \( u_i : S \times \tilde{A} \rightarrow \mathbb{R} \); finally, \( T : S \times \tilde{A} \rightarrow \Pi(S) \) is a stochastic transition function, specifying the probability of the next game to be played based on the game just played and the actions taken in it [9].

Hence, the SG includes as many static games, \( \Gamma = \{ A_1, \ldots, A_n : u_1, \ldots, u_n \} \), as its stages. In general, games may have various properties, either known or not, either observable or not, etc. We will focus on games where, for each stage \( t \), the outcomes of the \( t-1 \) stage are observed before the \( t^{th} \) stage begins.

IV. GENERATOR’S BEHAVIOR UNDER MODIFIED Q-LEARNING

A. Reinforcement Learning

In this setting, there are two possible things to learn, according to [9]. “First, the agent can learn the opponent’s strategy, so that the agent can then devise a best response. Alternatively, the agent can learn a strategy of his own that does well against the opponents, without explicitly learning the opponent’s strategy. The first is sometimes called model-based learning and the second model-free learning”. The second is the case in Reinforcement Learning (RL).

Many learning theories developed as a result of man’s effort to analyze the behavior of animals and artificial systems. RL is one of them and focuses on the effect of rewards (positive payoffs) and punishments (negative payoffs) on subjects’ choices in their attempt to achieve a goal [15].

RL theory’s basic elements are:

- the learner or the decision-maker, who is called the agent, and
- everything it interacts with, which is called the environment.

A standard assumption is that the agent is able to observe those aspects of the state of the world that are relevant to deciding what to do. In the environment the agent seeks to achieve a goal despite the prevailing uncertainty. The agent’s actions are permitted to affect the future state of the environment, thereby affecting the options and opportunities available to the agent at later times. Correct choice requires taking into account indirect, delayed consequences of actions, and thus may require foresight or planning. At the same time, the effects of actions cannot be fully predicted; thus the agent must monitor its environment frequently and react appropriately [16]. As it becomes clear, the basic concept behind RL is trial and error search, since the agent explores its environment and learns from his mistakes.

B. Q-Learning Algorithm

Q-Learning algorithm, which proposed by Watkins [13], is among RL most popular algorithms. The main advantages of the algorithm lay with its simplicity; it is easily understood and implemented. Moreover, it can be used on-line and it is model free – it avoids building an explicit model of the environment or for the opponent’s strategy, when it comes to games – so it seems rather attractive approach to decision problems.

The agent \( g \) in Q-Learning keeps in memory a function \( Q_g(s, a_g) \) that represents the expected payoff he believes he will obtain by taking an action \( a_g \) while being in state \( s_g \); the function of the expected payoff is represented by a two-dimensional lookup table indexed by states and actions, whose elements are defined as Q-values.

The agent’s experience, concerning his interaction with the environment, consists of a sequence of distinct stages. Let \( S_g = \{ s_{g,1}, s_{g,2}, \ldots, s_{g,n} \} \) be the set of \( S_g \) possible states of the environment and \( A_g = \{ a_{g,1}, a_{g,2}, \ldots, a_{g,n} \} \) be the set of \( A_g \) admissible actions the agent \( g \) can take. In the \( t^{th} \) stage, the agent:

1. Observes its current state \( s_{gt}^{(t)} \in S_g \).
2. Selects and performs an action \( a_{gt}^{(t)} \in A_g \) using a policy.
3. Observes the subsequent state \( s_{gt+1}^{(t+1)} \in S_g \).
4. Receives an immediate payoff \( r_{gt}^{(t)}(s_{gt}^{(t)}, a_{gt}^{(t)}) \).
5. Updates his Q-values according to:

\[
Q_g^{(t)}(s_{gt}^{(t)}, a_{gt}^{(t)}) = \begin{cases} 
(1-\alpha_g^{(t)}(s_{gt}^{(t)}, a_{gt}^{(t)}))Q_g^{(t-1)}(s_{gt}^{(t)}, a_{gt}^{(t)}) + \alpha_g^{(t)}(s_{gt}^{(t)}, a_{gt}^{(t)})Q_g^{(t-1)}(s_{gt}^{(t)}, a_{gt}^{(t)}) & \text{if } s_{gt} = s_{gt}^{(t)} \text{ and } a_{gt} = a_{gt}^{(t)} \text{,} \\
Q_g^{(t-1)}(s_{gt}^{(t)}, a_{gt}^{(t)}) & \text{otherwise.} 
\end{cases}
\]  

(9)

According to equation (9), only Q-values corresponding to current state and last action chosen are updated. \( \alpha_g^{(t)} \) is a learning rate in the range \((0,1)\), that reflects the degree to which estimated Q-values are updated by new data and can be different in each stage.

C. SA-Q-Learning Algorithm

The SA-Q-Learning algorithm was proposed by Guo, Liu and Malec [17]. They applied the Metropolis criterion [18], used in the Simulated Annealing (SA) algorithm [19], in order to determine the action-selection strategy of Q-learning.

The SA-Q-learning can be described by the steps 1-5 mentioned in the previous subsection, with the second step being replaced by the following actions:

a. Selects an action \( a_{gt}^{(t)} \in A_g \) randomly.

b. Selects an action \( a_{gt}^{(t)} \in A_g \) following a greedy policy:

\[
a_{g,\rho} = \arg \max_{a_g} Q_g^{(t-1)}(s_{gt}^{(t)}, a_g) \]

c. Generates a random number \( \xi \in (0,1) \).

d. Selects and performs action \( a_{gt}^{(t)} \in A_g \) as follows:
a_g^{(t)} = \begin{cases} 
  a_{g,p} & \text{if } \xi \geq \exp \left[ \frac{Q_g^{(t-1)}(s_g,a_{g,r}) - Q_g^{(t-1)}(s_g,a_{g,p})}{T_g^{(t)}} \right] \\
  a_{g,r} & \text{otherwise} 
\end{cases} \quad (10)

\text{e. Calculates } T_g^{(t+1)} \text{ by the temperature-dropping criterion.}

Although the temperature-dropping criterion can be in general arbitrary, in this paper the geometric scaling factor criterion is used, as in [17]. Let \( T_g^{(t)} \) be the Temperature in the \( t \)th stage and \( \lambda \in (0, 1) \) a constant, usually close to 1, in order to guarantee a slow decay of the temperature in the algorithm. Then in the \( t+1 \) stage the temperature will be:

\[ T_g^{(t+1)} = \lambda T_g^{(t)}, \quad t = 0, 1, 2, \ldots \] \quad (11)

D. Modeling Generator’s Behavior

The application of the SA-Q-learning algorithm requires the definition of the possible states, the admissible actions and the returned payoff.

\textit{States.} The state set of generator-agent \( g, S_g \), includes all the possible combinations of the joint actions of the other participants. The current state \( s_g^{(t)} \) is defined as the joint actions selected by the participants in the previous, \((t-1)^{th}\), stage.

\textit{Action.} The generator-agent action is the selection of the offer quantity and price, \((p_g, h_g)\).

\textit{Payoff:} The payoff received by each agent during an auction round is equal to the profit, in €, the agent makes by participating in the spot market, defined in (6).

The learning rate in this paper is designed to be state-action dependent as in [11] and [20]. The learning rate \( \alpha_g^{(t)}(s_g, a_g) \) is inversely proportional to the visited number \( \beta_g^{(t)}(s_g, a_g) \) of action \( a_g \) from state \( s_g \) up to the present trading stage, as follows:

\[ \alpha_g^{(t)}(s_g, a_g) = \frac{1}{\beta_g^{(t)}(s_g, a_g)} \quad (12) \]

E. Agent Based Energy Market Simulation

The spot energy market simulation consists of the repetition, for a large number of stages, \( t = 0, 1, 2, \ldots, t^{\max} \), of the following steps:

\text{Step 1:} All generator-agents observe their current state \( s_g^{(t)} \), select an action \( a_g^{(t)} \equiv (p_g^{(t)}, h_g^{(t)}) \) according to the policy defined by the SA-Q Learning, and submit their Energy Offers defined by the selected action to the ISO.

\text{Step 2:} The ISO processes the Energy Offers submitted by all generator-agents, along with transmission system and nodal demand information, and computes the quantities and prices that clear the market by solving the OPF problem (1)-(5).

\text{Step 3:} The ISO posts the public information on nodal prices, \( LMP_k, \forall k \in K \), and the Energy Offers submitted by all generator-agents, \( a_g^{(t)} \equiv (p_g^{(t)}, h_g^{(t)}) \), \( \forall g \in G \), and informs every generator-agent \( g \in G \) about the quantity, \( p_g \), of his Energy Offer accepted in the spot market.

\text{Step 4:} All generator-agents use the information they receive from the ISO (Step 3) to compute profits according to (6) and update their Q-Tables according to (9).

\text{Step 5:} The stage count, \( t \), the Temperature and the learning rate are updated. The whole process, Step 1 through Step 5, is repeated if the stage count, \( t \), is less than the maximum number of stages, \( t^{\max} \).

V. TEST CASES

A simple, two-node system, shown in Fig. 1 is used in our test cases. The transmission capacity limit is 100 MW. The generator data are shown in Table 1. Locational Marginal Pricing is used for market settlement, as already discussed. The market price cap is 40 €/MWh.

Two cases are examined:

In Case A each generator offers its full capacity at marginal cost, so that competitive prices result. This case is used as reference to test the exercise of market power by the generators.

In Case B each generator participates in a repeated energy auction trying to maximize its profits, by reinforcement learning, as described in Section IV.

\textbf{Parameter Selection.} All generators are considered to be players in our market and their behavior is modeled through the SA-Q-learning algorithm, as described before. The parameters of the algorithm that need to be defined are the initial temperature \( T^{(0)} \) and the constant \( \lambda \) of the temperature-dropping criterion. All generators have the same parameters \( T^{(0)} = 100,000 \) and \( \lambda = 0.99 \).

In the simulations presented each agent has an action space consisting of 21 different levels for the offer quantity, the step is 25 MW, while the step for the offer price is 2 €/MWh.

![Two-Node Test System](image-url)
A. Reference Case: Two Generator-Agents

Both generators offer their full capacity at marginal cost, so that competitive prices result. The market clearing results under competitive prices are presented in Table II. Owing to the 100 MW transmission limit, the cheaper generator, Gen-1, is dispatched only up to 200 MW. The remaining 100 MW of the Node-2 demand are supplied by the more expensive local generator, Gen-2. There is locational price difference, owing to congestion, and the LMP at each node is equal to the marginal cost of the local generator. Hence, neither generator makes profit from the energy market, while the ISO collects €1000 of congestion rent.

B. Monopoly Case: Two Generator-Agents

When both generator-agents act strategically, trying to maximize profits through reinforcement learning, the resulting market conditions are shown in Tables III and IV. The simulation runs for 2500 iterations. As shown in Table III the first agent identifies his potential market power and exercises it by withholding output. He offers only 200 MW, in order to leave the transmission line un-congested and be paid at the LMP of Node-2. Hence, he manages to raise the LMP of Node-1 to a level certainly greater than 30 €/MWh and increase his profits, from 0 € to a level greater than (30 – 20) * 200 = 2000 €. As stated in the Introduction: market power is “the ability to alter profitably prices away from competitive levels” [1], [2], and this is what Gen-1 does, by withholding capacity.

According to [8], this is consistent with Coase Theorem (1960), which supports the argument that “in the absence of transaction costs and with public knowledge of transmission capacity, bargaining among buyers and sellers will capture all congestion rents”. In our case, despite the fact that Gen-1 does not know the transmission capacity, he “learns” to withhold capacity through repetition.

Since there is no congestion, both producers are paid at the same system-wide Market Clearing Price, MCP. The resulting MCP is equal to the market price cap, as shown in Table IV owing to the fact that agent Gen-2 realizes his monopoly power of the local, Node-2, demand and the absence of demand elasticity at Node-2. Hence, he chooses to exercise his monopoly power and offer his quantity to the higher price he can; consequently, his offer price is equal to the market price cap, as shown in Table III, and his profits have been increased from 0 € to 1000 €.

As can be easily shown after some simple calculations, none of the generator-agents can do anything better, given the choice of the other. Therefore, since no agent can profitably deviate from this point, by altering unilaterally his strategy, this point is a Nash Equilibrium.

C. Simulation Environment-Execution Time

The agent based simulation has been developed in JBuilder 2005 environment using Java (J2SE 5.0). A commercial package, GAMS 2.5 (CPLEX solver), is used for the solution of the ISO market clearing problem. All simulations run on an AMD Athlon™ Processor 3200+, 2.01 GHz, 1.75 GB RAM. The execution time for case B was 21.65 min.

VI. CONCLUSIONS

An analysis of capacity withholding in a simulated electricity market was presented in this paper. The electricity market was formulated as a stochastic game, where each hourly auction is represented by a stage of the game. For the needs of the analysis agent-based simulation was employed, where each generator was modeled as an adaptive agent, following a SA-Q-learning bidding behavior. Test cases on a simple two-node test system, with two competing generators participating in a repeated energy auction can learn to exercise their monopoly power following both the ways described in Section I Introduction. They can learn to develop capacity withholding strategies (Gen-1), even when their behavior is simulated through the simple model presented in Section IV. They can also recognize their locational market power (Gen-2) and raise the market price based only on publicly available information (LMPs and Energy Offers).

VII. REFERENCES


