

Comparison of Two Learning Algorithms in Modelling the Generator's Learning Abilities

Zhifeng Qiu

ELECTA/ESAT
Katholic University of Leuven
Leuven, Belgium
zhifeng.qiu@esat.kuleuven.be

Eefje Peeters

Flemish Institute for Technological
Research
Mol, Belgium
eefje.peeters@vito.be

Geert Deconinck

ELECTA/ESAT
Katholic University of Leuven
Leuven, Belgium
geert.deconinck@esat.kuleuven.be

Abstract—This paper discusses the generator's optimal bidding problem. Reinforcement learning is employed to model the generator's learning ability. Through the repeated learning, the generator can develop optimal bidding in the point view of long term. Simulation result shows the generator equipped with learning ability can definitely perform better than the one without learning ability. The learning ability increases the profit of this 'smart' generator, who exercises more market power than the 'normal' generator. It's the main advantage that the generator with learning gets. We compare two learning algorithms, and conclude that SA-Q agent can always converge to the optimal action, but VRE can't. VRE has serious stochastic characteristic, which lead agent converge to one action randomly. In this work, VRE is quite sensitive to the parameters of system. However SA-Q has no such problem, which can lead agent converge to the optimal action.

Keywords- *Optimal Bidding; Power Market; Reinforcement Learning; Game Theory.*

I. INTRODUCTION

SINCE the early 1990's, electricity markets in many parts of world have moved away from vertically integrated monopolies, towards liberalized structures in an effort to increase competition. In such market, all competitive power generators (supply) and buyers (demand) are required to submit blocks of energy amounts and corresponding prices they are willing to receive from or pay to the pool. The prices and quantities submitted by the market participants are binding obligations, as they require financial commitments to the market. Once all the supply and demand bids have been submitted and the bidding period ends, an independent system operator (ISO) uses the special market clearing mechanism to work out the amount that each participant will supply or consume. At the meanwhile, market-clearing price (MCP) is cleared.

Market power is defined as the ability of a seller to maintain prices above competitive levels for a significant period of time. If the generator can successfully increase its profits by strategic bidding or by any means other than lowering its costs, it is said to have market power. It is normal to assume that the generator making the bid has certain degree

of freedom in deviating their bid function from their actual cost function. Such deviation gives the generator the opportunity to maximize their profits.

In this paper, a network-constrained pool market is set as the market framework. The cost function of generator is piecewise linear curve. The set of allowable supply function is a two degree-of-freedom parameterization involving the price and the quantity of block. That means both two components of market-power strategy, quantity withholding and financial withholding, are investigated in this paper. It is different from either Cournot model or Bertrand model. In Cournot model, each player decides the quantity to be produced and leaves the price to the market. In Bertrand's model, each player decides the price and leaves quantity to the market. In this paper, both price and quantity is decided by player to perform optimal strategic bidding.

Broadly speaking, there are three ways for developing optimal bidding strategies [1]. The first one relies on estimations of the market clear price (MCP) in the next trading period, the second utilizes estimations of bidding behavior of the rival participants, and the third is game theory based. In this paper, we focus on the last one. Gaming is market participants engaging in uncompetitive behavior that takes advantage of certain market rules and system conditions by deviating from normal bidding, scheduling and operating patterns [2].

In the system we are researching on, we assume the generator just know its own private information. The only public information is MCP. Thus it presents a more realistic model of the electricity market, meanwhile incorporating real power system physical constraints. The long term maximization of individual profit is obtained by equipping generator with learning capability. Specifically, the learning method investigated in this paper is reinforcement learning. We made a comparison of two learning algorithms, simulated annealing-Q-Learning algorithm and Roth-Erev reinforcement learning algorithm, in the power market context.

The paper is organized as follows. In Section 2 we introduce the market clearing mechanism which solves an optimal power flow problem. Section 3 describes the noncooperative game and two kinds Q learning method. In

Section 4 we set up numerical and discuss the results obtained. Eventually, Section 5 concludes the paper.

II. MARKET FRAMEWORK AND CLEARING MECHANISM

A. Market framework

The electricity market framework is modeled as a Multi-Agent System. In the system, there are some agents that represent generators and loads. The independent system operator (ISO) first collects these information and then use the specific clearing mechanism to clear the market. The cleared results include the dispatched power quantity and the market clearing price (MCP) for each entity in the system. The generators and loads continuously update their ask and bid curves in each auction period based on the obtained profit.

B. Market clearing mechanism

Here, the physical constraints of power system are considered. The Optimal Power Flow (OPF) is used as the market clearing mechanism. The objective of OPF is to find a steady state operation point which minimizes generation cost, while maintaining an acceptable system performance in terms of limits on generators' active and reactive powers, line flow limits, maximum output of various compensating devices etc. After computing, OPF gives the optimal power generated and purchased at each bus and the nodal prices which are important conception in the electricity market. Nodal prices [3] are of specific interest because they reflect the marginal cost of delivering one more unit at each bus (node). These prices are also called locational prices and are found to be the optimal prices, maximizing social welfare and taking transmission constraints into account. In an uncongested grid all locational price are equal, whereas in case of congestion locational price difference occurs. The MATPOWER [4] is employed in the simulation.

In the system we are researching on, nodal price (locational price) is taken as the market cleared price. Each generator uses this price to calculate its own profit. The generator's objective is to maximize its profits in the market, by selecting the parameters of its energy offer (quantity-price curve). The profit of a generator is the amount of revenue received from selling the power, minus the cost of supplying the power.

The generator just knows its own price-quantity curve information, who does not know the information on the all power network, the consumer demand and the competitor offers.

In the next session we will describe this game process in which the generator can "learn" through the repetition of the energy auction to select the profit maximizing energy bid, based only on those information mentioned here.

III. GAME THEORY AND REINFORCEMENT LEARNING

A. Introduction

Game theory is particularly concerned with finding equilibrium points, defined as the set of strategies [4]. One set is for each decision maker, from which nobody finds

convenient to deviate. In this way, models based on games theory find an intermediate placing (oligopoly) between perfect competition and monopoly. The most typical situation in an electricity market is that a limited number of big producers (plus eventually a small number of price takers) satisfy a market constituted by a large number of customers whose demand is essentially rigid because of the scare possibility to modulate consumption in function of price. Thus producers may behave strategically and this market is an oligopoly [5].

Reinforcement learning is a technique to find the optimal action of each individual to reach the equilibrium points of system in a game. Q-learning is one method of reinforcement learning technique that works by learning an action-value function that gives the expected utility of taking a given action in a given state and following a fixed policy thereafter. One of the strengths of Q-learning is that it is able to compare the expected utility of the available actions without requiring a model of the environment [6]. In the Q learning, there are two important elements, namely Q value and control policy.

▪ Q value

The problem model consists of an agent, states S and a number of actions per state A . By performing an action a , where $a \in A$, the agent can move from state to state. Each state provides the agent a reward (a real or natural number) or punishment (a negative reward). The goal of the agent is to maximize its total reward. It does this by learning which action is optimal for each state.

The algorithm therefore has a function which calculates the quality of a state-action combination: $Q: S \times A \rightarrow \mathbb{R}$.

Before learning has started, Q returns a fixed value, chosen by the designer. Then, each time the agent is given a reward (the state has changed) new values are calculated for each combination of a state s from S , and action a from A . The core of the algorithm is a simple value iteration update. It assumes the old value and makes a correction based on the new information. The formula (4) shows this update.

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha(s_t, a_t)}_{\text{learning rate}} \times \underbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_{a'} Q(s_{t+1}, a_{t+1})}_{\text{max future value}} \right)}_{\text{expected discounted reward}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \quad (4)$$

Where r_t is the reward given at time t , α ($0 < \alpha \leq 1$) the learning rates, may be the same value for all pairs. The discount factor γ is such that $0 < \gamma \leq 1$.

▪ Control policy

The aim of control policy is to find the proper balance between exploration and exploitation in Q-learning [6]. Exploitation limits the agent to know more about the environment. It will lead agent to locally optimal policies, possibly differing from a globally optimal one. In contrast,

exploration makes the agent obtain more knowledge about the environment, which is beneficial for agent to reach the optimal policies. However, excessive exploration will drastically decrease the performance of the learning algorithm especially after some learning processes.

A simple strategy proposed to deal with this problem is the ϵ -greedy (with $0 \leq \epsilon < 1$), with larger ϵ corresponding to larger probability of exploration. The value of ϵ has obviously a great impact on the algorithm.

In the following parts, two revised Q-learning algorithms will be introduced and compared in the simulation parts.

B. Simulated Annealing-Q-Learning Algorithm (SA-Q)

Simulated Annealing-Q-Learning Algorithm is proposed by Guo [7]. Compared with the normal Q-learning, SA-Q learning algorithm employs the Metropolis criterion of simulated annealing algorithm to balance exploration and exploitation, which can accelerate the agent to converge to the optimum avoiding the excessive exploration during the learning process.

Algorithm: SA-Q-learning algorithm [7].

1. Initiate arbitrarily all $Q(s, a)$ values;
 2. Repeat (for each episode):
 - (a) Choose a random (initial) action a ;
 - (b) Repeat (for each step in the episode):
 - i. Select an action a_r in $A(s)$ arbitrarily;
 - ii. Select an action a_p in $A(s)$ according to the policy;
 - iii. $a \leftarrow a_p$
 - iv. Generate random value $\xi \in (0, 1)$
 - v. If $\xi < \exp(\frac{Q(s, a_r) - Q(s, a_p)}{Temperature})$ (5), then $a \leftarrow a_r$
 - vi. Execute the action a , receive immediate reward r , then observe the new state s'
 - vii. $Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$ (6)
 - viii. $s \leftarrow s'$
- Until s is one of the goal states
- (c) Recalculate *Temperature* by the temperature-dropping criterion,

$$Temperature = \text{Lambda} * Temperature \quad (7)$$

Until the desired number of episodes has been investigated.

C. Roth-Erev learning algorithm (VRE learning algorithm)

Roth-Erev reinforcement learning algorithm was proposed by Erev and Roth [8]. Unlike the SA-Q algorithm, where the action choice only depends on the absolutely accumulated Q value, in VRE algorithm the action choice is determined by the choice probability. The choice probability is simultaneously determined by both the Q value of the chosen action and that of the other non-chosen actions in the action domain for one agent.

For this algorithm, the algorithm performance, convergence for example, is affected largely by the setting value of parameters. In the studies of Roth and Erev, there was a calibration process for the learning parameters in the algorithm. The purpose of calibration is to obtain a “best fit” set of learning parameters that best mimics human learning behavior observed in a wide range of games. This problem will be detailed discussed in the following simulation part.

Algorithm: VRE learning algorithm [8]

1. Initiate all $Q(s, a)$ and choice propensity p values;
2. Repeat (for each episode):
 - (a) Choose a random (initial) action a ;
 - (b) Repeat (for each step in the episode):
 - i. Select an action a according to the propensity p ;
 - ii. Execute the action a , receive immediate reward r , then observe the new state s'
 - iii. Update Q value for each action in the action.

$$Q(s, a) \leftarrow (1 - r_i)Q(s, a) + \text{Reward}_{im} \quad (8)$$

where,

$$\text{Reward}_{im} = \begin{cases} (1 - e_i) \cdot \text{immediate_Profit}, & \text{for the chosen action} \\ \frac{e_i \cdot Q(s, a)}{M_i - 1}, & \text{for the non-chosen action} \end{cases} \quad (9)$$

where, r_i is the recency parameter. e_i is the experimentation parameter.

- iv. Update propensity p values for each action.

$$p_i = \frac{\exp(Q_i / C_i)}{\sum_{j=1}^m \exp(Q_j / C_i)} \quad (10),$$

where, C_i is the cooling parameter that affects the degree to which agent i makes use of propensity values in determining its choice probabilities.

- v. $s \leftarrow s'$

Until s is one of the goal states

Until the desired number of episodes has been investigated.

IV. NUMERICAL SIMULATION

In the simulation, generator offer the piecewise linear price-quantity curve to the ISO. The curve is illustrated in Fig. 1. A generator could offer, for each generating unit, one energy block consisting of the maximum capacity and price equal to the marginal cost of producing this amount.

A 3-bus with 2 generators system revised from IEEE 6-bus system is used in the test cases. Fig.2 shows the sketch of the system. The all physical parameters of system include those of all generators, bus, branch, area, generation cost. Gen1 and Gen2 are two generators, who serve one constant load in the node_3 shown in Fig.2. Locational marginal price is used for market settlement. The quantity-price curve of each generator is 3-block piecewise linear curve. The 4 break points of 3-block for quantity are [0, 12, 36, 60] MW. The market price cap for each generator is same, namely [0, 120, 144, 184] Euro. In the

simulation, the generator bids for both quantity and price. The marginal price for each generator is [0, 20, 44, 84] Euro.

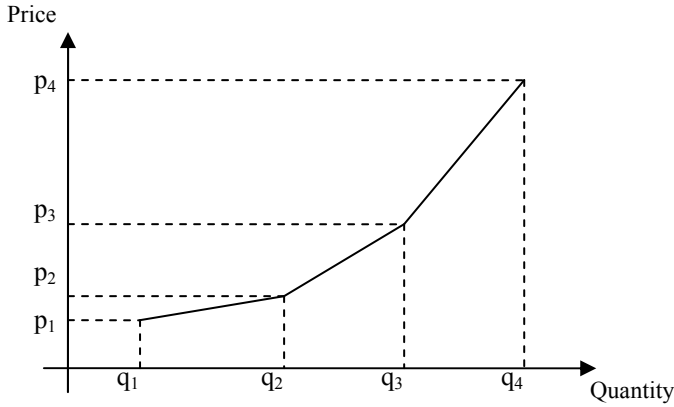


Fig.1 Piecewise linear supply curve of generator

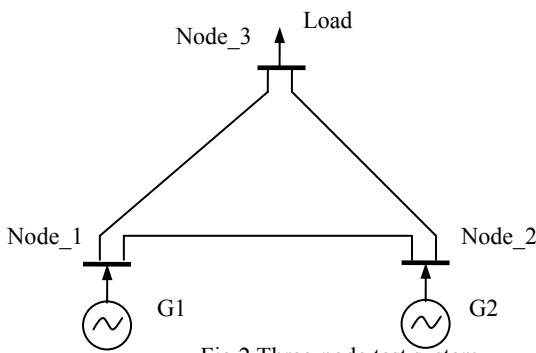


Fig.2 Three-node test system

There is no congestion in the system, low level load (=40KW). So the relation between Gen-1 and Gen-2 is competitive.

In the simulation, we make such scenario: Gen 2 has no learning ability; Gen 1 is a player with learning ability in this system. So we want to test how advantage for one with learning ability compared with the non-learning one.

A. Using SA-Q learning algorithm

Parameter setting is like this, Initial Temperature=100; Lambda=0.99; initial Q = 300. The learningrate is designed to be action dependent, as in [9]. The learning rate is inversely proportional to the visited number of action up to the present auction iteration, as follows: LearningRate=1/VisitingNum(ChosenAction). The simulation day is 100.

In this case, Gen 2 is set to always choose the certain action, here, the 10th action; in action 10, the price of Gen-2 is the lowest of all actions that the Gen-2 has. So, it is doomed that Gen-2 has better competitive ability. We can see this from the simulation result. Gen 1 is a player in this market. From Fig.3.c, for Gen 1, with the auction stage going on, some actions' Q value diminished gradually. This means the opportunity to choose such action become less and less.

Comparably, other actions' Q values are strengthened due to the accumulated Q value. The opportunity to choose such action becomes bigger and bigger. The action 9 is proven to be the optimal bidding action for the player Gen 1. From Fig.3.b, we can see that Gen 1 converge to this action after almost 40 iterations. It should be mentioned here that in this system there is more than one optimal bidding action. Action 18, 27 are also optimal actions. As long as the Gen 2 converges to one of these three actions, the system reaches the equilibrium. When this player converges, the profit of whole system is highest, which can be saw from Fig.3.a. Q-learning algorithm is a kind of stochastic algorithm, that's why after convergence, the system whole profit still have two falling points in the Fig.3.a.. The cause of this problem is that algorithm has very small chance to deviate the action from the found optimal action.

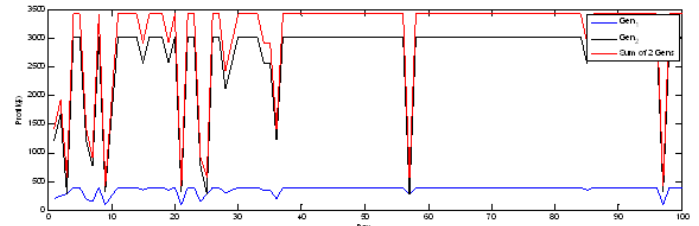


Fig.3.a. Profit of Gen1, Gen2 and profit sum of two Gens

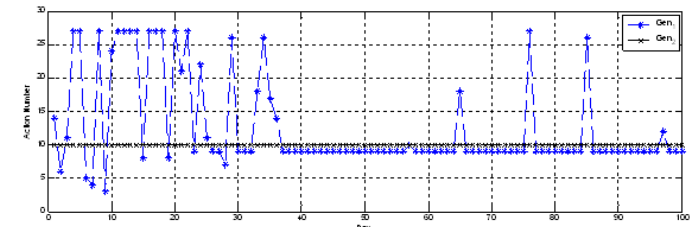


Fig.3.b. Action choice of two Gens

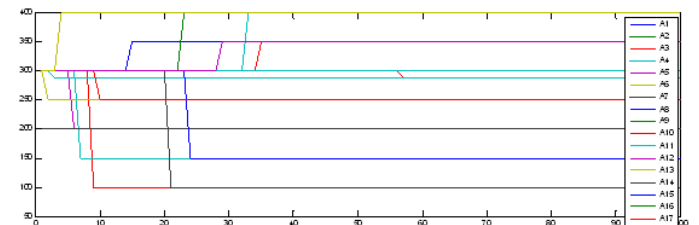


Fig.3.c. Q value of Gen1

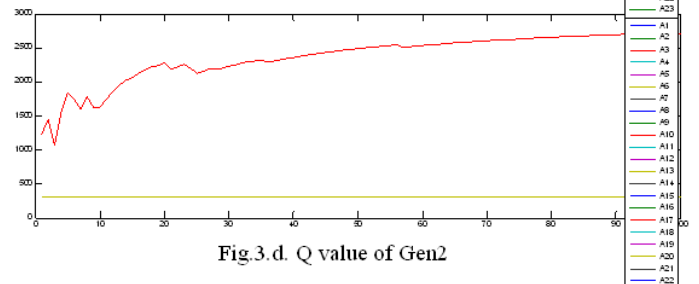


Fig.3.d. Q value of Gen2

B. Using VRE learning algorithm

In this algorithm, we divide the learning process into two stages. One is learning stage, another is learned stage. Parameter setting is like this, in the learning stage, $C=40$; LearningRate=0.8; $e_rate=0.03$; $Q_ini=200$; In the learned stage, $C=20$; LearningRate=0.99; $e_rate=0.2$;

In this case, like the previous section, only Gen 1 is a player in the market. We can see in Fig. 4.a that after 70 iterations, Gen 1 converges to action 8. But different from the previous section, the converged action is not the best response to the system. That means in the VRE learning algorithm the agent sometimes can't converge to the optimal action. This is the most unfavorable drawback for VRE learning algorithm. In this kind of reinforcement learning algorithm, the choice of action depends on the calculated choose propensity of each action. But the propensity value depends on two factors, which can be seen from equation (10), one is the Q value of selected action itself, the other is the Q value sum of all actions in the action domains. So the choice of action is determined by a relative quantity, which, compared with SA-Q, is prone to make the agent premature fixation on suboptimal action or even non-optimal action. Also from the Fig. 4.b., for the Gen 2, in the exploring stage that is before 70 iterations, the action searching shows the serious fluctuation. That means in VRE learning algorithm, the agent should spend more time on the action exploration, instead of exploitation. The longer exploration time means the lower learning efficiency, which is not favorable in good reinforcement learning.

In these points, VRE learning algorithm is not as good as the SA-Q.

The simulation result found in this work has considerable discrepancy with reference [10], which says, "The VRE learning algorithm is well-defined for any action domain consisting of finitely many elements, regardless of the precise nature of these elements". In our case, VRE is quite sensitive to the parameters of system. But this maybe caused by the different system setting up.

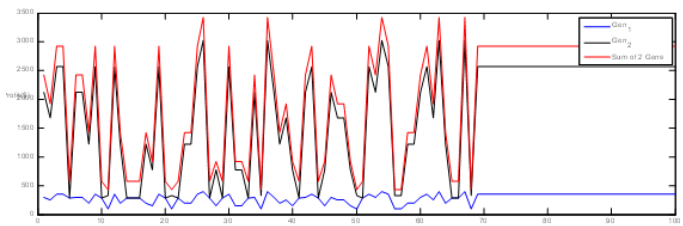


Fig.4.a. Profit of Gen1, Gen2 and profit sum of two Gens

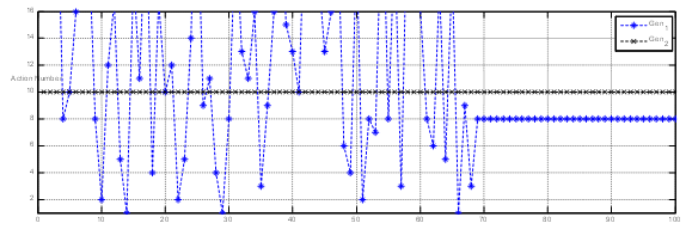


Fig.4.b. Action choice of two Gens

Fig. 4.c and Fig. 4.d shows the evolution of each action's Q

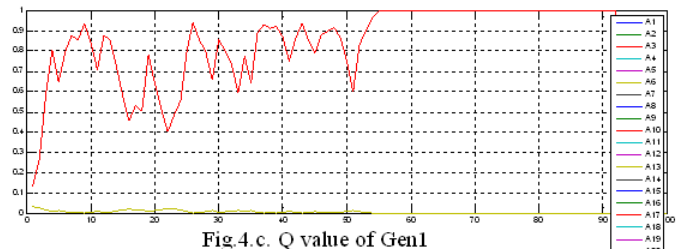


Fig.4.c. Q value of Gen1

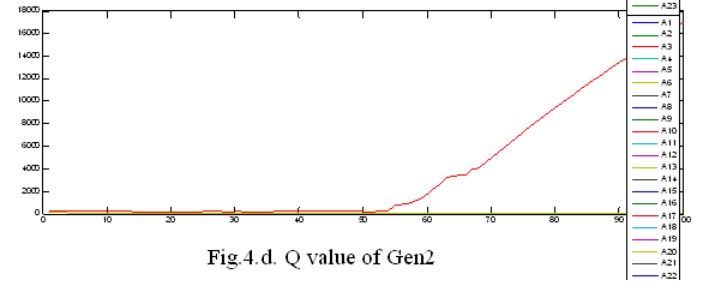


Fig.4.d. Q value of Gen2

value for two Gens.

V. CONCLUSIONS

In this work, we discuss the generator's optimal bidding problem. To solve this, we model the generator's reinforcement learning ability. Through the repeated learning, the generator can develop optimal bidding in the point view of long term.

Simulation result shows the generator equipped with learning ability can definitely increases the profit of this 'smart' generator. It's the main advantage that the generator with learning gets. We compare two learning algorithms, and conclude that SA-Q agent can always converge to the optimal action, but VRE can't. VRE has serious stochastic characteristic, which lead agent converge to one action randomly. This conclusion has considerable discrepancy with reference [10], which maybe is caused by different simulation setting up. In this work, VRE is quite sensitive to the parameters of system. However SA-Q has no such problem, which can lead agent converge to the optimal action.

The future work is to investigate the learning ability of multi-agent system. So far, we just test one learning agent in the system. For the multi-agent system, it is a challenge topic. The difficulty lies in, first, the environment, which consists of other agents who are similarly adapting, is no longer stationary. Second, the nonstationarity of the environment is not generated by an arbitrary stochastic process, but rather by other agents, who might be presumed rational or at least regular in some important way. In the future work, it is important to model the other generator's bidding behavior.

REFERENCES

[1] A.K.David, Fushuan Wen, "Strategic Bidding in Competitive Electricity Markets: a Literature Survey," Power Engineering Society Summer Meeting, 2000. IEEE, Vol. 4, pp. 2168-2173, July, 2000, Seattle, WA, USA

- [2] R. D. Luce and H. Raiffa, *Games and Decisions*. Mineola, NY: Dover, 1989.
- [3] D. Ray, F. Alvarado. Use of an engineering model for economic analysis in the electricity utility industry. presented at the Advanced Workshop on Regulation and Public Utility Economics, May 25-27, 1988
- [4] R. D. Zimmerman and Carlos E. Murillo-Sánchez, D. Gan, "MATPOWER: A MATLAB Power System Simulation Package (Version 3.2)," Cornell University, New York 2007. available on <http://www.pserc.cornell.edu/>
- [5] Vasileios P. Gountis, Anastasios G. Bakirtzis, Bidding Strategies for Electricity Producers in a Competitive Electricity Marketplace, IEEE TRANSACTIONS ON POWER SYSTEMS, VOL. 19, NO. 1, FEBRUARY 2004
- [6] Dutta, Prajit K. (1999), Strategies and games: theory and practice, [MIT Press](http://www.mitpress.com/), ISBN 978-0-262-04169-0.
- [7] Leslie Pack Kaelbling, Michael L. Littman, Andrew W. Moore, Reinforcement Learning: A Survey, *Journal of Artificial Intelligence Research*, 4(1996), 237-285
- [8] M. Guo, Y. Liu, and J. Malec, "A new Q-learning algorithm based on the metropolis criterion," *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, vol. 34, no. 5, pp. 2140–2143, Oct. 2004.
- [9] Roth, Alvin E., and Erev, Ido. Learning in extensive form games: Experimental data and simple dynamic models in the intermediate term, *Games and Econ. Behavior* 8, 164-212. 1995.
- [10] J. Nie and S. Haykin, "A dynamic channel assignment policy through Q-learning," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1443–1455, Nov. 1999.
- [11] Junjie Sun, Leigh Tesfatsion, Dynamic Testing of Wholesale Power Market Designs: An Open-Source Agent-Based Framework, Working Paper No. 06025, available on <http://www.econ.iastate.edu/tesfatsi/DynTestAMES.JSLT.pdf>