Nash Equilibria and Reinforcement Learning for Active Decision Maker Modelling in Power Markets

Thilo Krause, Göran Andersson, Damien Ernst[†]
Power Systems Laboratory
Swiss Federal Institute of Technology Zürich
[†]Visiting researcher from University of Liège
{krause,goran.andersson}@eeh.ee.ethz.ch, dernst@ulg.ac.be

Elena V. Beck, Rachid Cherkaoui, Alain Germond
Laboratoire des Réseaux d'énergie Electrique
Swiss Federal Institute of Technology Lausanne
{elena.vdovina,rachid.cherkaoui,alain.germond}@epfl.ch

***Abstract*— In this paper, we study the behavior of power suppliers who submit their bids to the market place in order to maximize their payoffs. The market clearing mechanism is based on the locational marginal price.**

To study the interaction of the power suppliers, we rely on two different approaches and compare the results obtained. One approach consists of computing the Nash equilibria of the market, and the other models each player's behavior by using reinforcement learning algorithms.

Simulations are carried out on a five node power system.

***Index terms*— Electricity markets modeling, spot markets, Nash equilibria, reinforcement learning, matrix games.**

I. Introduction

The main driver behind electricity markets restructuring was the willingness to achieve highly competitive markets with prices close to short-run marginal costs. The basic economic principle is that with perfect competitive market conditions, market participants maximize their own profit in a decentralized way and bid in the market a price equal to their marginal cost. In this way, competitive and efficient market results are attained. In such a case, the prices at the nodes of the network reflect the marginal cost of production and the marginal value for the consumers [1].

However, in most deregulated markets around the world, perfect competitive conditions do not hold true. The presence of few power producers who bid strategically, of a highly inelastic electricity demand, and of a transmission network that could be insufficient to accommodate the flows derived from a merit order dispatch are only some of the reasons that move the market results from the expected optimum.

Two families of approaches have been used to study the characteristics of imperfectly competitive electricity markets. One family computes the market equilibrium points [2], [3], [4]. The other family models the behavior of each agent (i.e., market participant) by a set of rules, builds a market dynamics and analyzes its characteristics through simulations [5], [6], [7].

In this paper we model the behavior of the active market participants in such a way that they are able to use their past experience to improve their behavior. To do so, we use an algorithm known as Q-learning that belongs to the class of reinforcement learning algorithms [8]. After having modeled the active market participants, we simulate the market and discuss the similarities that exist between the policy learned by the agents and the pure Nash equilibria of the market.

In Section II we introduce matrix games, define the notion of Nash equilibrium and describe the Q-learning algorithm. Section III describes the market structure considered, explains how the process of bidding to the power market for the different generators may be formalized as a matrix game and discusses whether reinforcement learning is better adapted than Nash equilibria to analyze this type of market. In Section IV we define our benchmark electricity market and analyze simulation results. Finally, Section V concludes.

II. Matrix games, Nash equilibrium and reinforcement learning

The theory of games is explicitly designed for reasoning about multi-agent systems [9]. There exists a vast variety of games, such as cooperative and non-cooperative games, static and dynamic games, 2-agent or n-agent games, etc. In this paper we consider n-agent matrix games defined through the following elements:

- a set of n agents $\{1, \cdots, n\}$
- $A^1, \cdots, A^n$ a collection of finite sets of *actions* available to each agent (A_i is the set of actions for agent i),
- $r^i : A_1 \times \cdots \times A_n \to R$ for $i \in \{1, \cdots, n\}$ is each agent's reward function, giving the reward gained by agent i for each set of action choices the group of agents could make.
- $\pi^i : A^i \to \Delta A^i$ the strategy for each agent i, where ΔA^i is the space of probability distributions over agent i actions. We may distinguish between pure strategies and mixed strategies. π^i is a pure strategy if there exists an $a^i \in A^i$ such that $\pi^i(a^i) = 1$, and is a mixed strategy otherwise. Equivalently, an agent plays a pure strategy if he plays with probability one an action and plays a mixed strategy otherwise. We denote by Π^i the set of strategies available to agent i.
- $r^i(\pi^1, \cdots, \pi^n)$ denotes the expected reward of agent i when the different strategies $\pi^1, \cdots, \pi^n$ are played, that is $r^i(\pi^1, \cdots, \pi^n) = \sum_{(a^1, \cdots, a^n) \in A^1 \cdots, A^n} \pi^n(a^1) * \cdots * \pi^1(a^n) * r^i(a^1, \cdots, a^n)$

A. Nash equilibrium

One important notion associated with a game is the notion of Nash equilibrium point. A Nash equilibrium is a joint strategy where each agent strategy is a best response to the strategies of the others. It is formally defined as follows:
The tuple of n strategies $(\pi^1_*, \cdots, \pi^n_*)$ is a Nash equilibrium if for all $i \in \{1, \cdots, n\}$ we have

$$r^i(\pi^1_*, \cdots, \pi^n_*) \geq r^i(\pi^1_*, , \pi^{i-1}_*, \pi^i, \pi^{i+1}_*, \cdots, \pi^n_*) \qquad (1)$$

for all $\pi^i \in \Pi^i$.

It can be shown that for every game there exists at least one Nash equilibrium. The appropriate method for computing Nash equilibria for a game depends on a number of factors. Certainly, the most important factor involves whether we want to simply find one equilibrium (*a sample equilibrium*) or find all equilibria. The problem of finding one equilibrium is a relatively well-studied problem, and there exists a number of different methods for numerically computing a sample equilibrium (see for example the Lemke-Howson algorithm [10] for a 2-agent game, and its extension by to an n-agent game by Rosenmüller [11]). While there exist methods for the computation of all equilibria, they take prohibitively much time for games beyond a rather small size.

We have observed the presence of pure Nash equilibria for the different game problems studied in this paper (see Section IV) - Nash equilibria for which the corresponding n strategies are pure strategies - but did not try to determine whether other equilibria existed. Since the action spaces A^i are finite in our examples, these Nash equilibria were computed by considering all the n-tuples $(a^1, \cdots, a^n) \in A^1 \times \cdots \times A^n$ and determining through Eqn (1) those which indeed correspond to equilibria.

B. Reinforcement learning

Reinforcement learning is the problem of an agent learning from experience. In the context of reinforcement learning, we suppose that the matrix game is played several times, and that each time the game is played the different agents observe their rewards and use these observations to adjust their strategy in order to maximize their next reward. We propose to use here for the problem of learning in matrix games the well-known Q-learning algorithm [12], which was initially designed for learning through interaction with a Markov Decision Process (MDP). There are several papers which discuss extensions of Q-learning algorithm to various types of games and study under which conditions the behavior of the players converge to a Nash equilibrium [8], [13].

When an agent i is modeled by a Q-learning algorithm, it keeps in memory a function $Q^i : A^i \rightarrow R$ such that $Q^i(a^i)$ represents the expected reward it believes it will obtain by playing action a^i. It then plays with a great probability the action it believes is going to lead to the highest reward, observes the reward it obtains and uses this observation to update its estimate of Q^i. Suppose that the tth time the game is played, the joint actions $(a^1_t, \cdots, a^n_t)$ represent the actions the different agents have taken. After the game is played and the different rewards have been observed, agent i updates its Q^i-function according to the following expression:

$$Q^i(a^i_t) \leftarrow Q^i(a^i_t) + \alpha^i_t(r^i(a^1_t, \cdots, a^n_t) - Q^i(a^i_t)) \qquad (2)$$

where $\alpha^i_t \in [0, 1]$ is the degree of correction. If $\alpha^i_t = 1$, the agent supposes that the expected reward it will get by taking action a^i in the next game is equal to the reward it just observed. If $\alpha^i_t = 0$, it means the agent does not use its last observation to update the value of its Q^i-function.

If all the agents use a time-invariant policy[1], then it can be shown that $Q^i(a^i)$ $\forall a^i \in A$ $\forall i \in \{1, \cdots, n\}$ indeed converges towards the expected reward obtained by agent i while playing action a^i, if α^i_t satisfies the conditions

$$\sum_{t=1}^{\infty} \alpha^i_t = \infty \quad \sum_{t=1}^{\infty} \alpha^{i^2}_t < \infty \qquad (3)$$

and if action a^i is played an infinite number of times.

Similarly, it can be shown that if all the agents except agent i use a time-invariant strategy, if α^i_t satisfies conditions (3) and if action a^i is played an infinite number of times, then $Q^i(a^i)$ converges towards the expected reward obtained by agent i while playing action a^i.

We will suppose in this paper that the agents select their actions according to the so-called ϵ-Greedy policy. When an agent i uses an ϵ-Greedy policy to choose its action, it selects with probability $1 - \epsilon$ the action which maximizes its believed expected reward $(\underset{a^i \in A^i}{\arg\max} Q^i(a^i))$, and chooses with probability ϵ an action at random in A^i. The main reason for an agent to adopt a policy that selects from time to time an action that it believes does not lead to the highest expected reward, is to guarantee that all actions have been tried a sufficient number of times to be able to correctly assess their expected reward.

Even if the value of ϵ is chosen to be constant for each of the agents, they will constantly update their Q^i-functions and their policies become time-variant. Therefore, nothing can be firmly said about the convergence of these reinforcement learning algorithms. However, as we have observed in our simulations (see Section IV), the learned Q^i-functions sometimes remained almost unchanged after a certain learning time, and their corresponding *greedy actions* - the actions that maximize their Q^i-functions - corresponded to a pure Nash equilibrium or said otherwise, after playing several games, the joint pure strategies $(\underset{a^1 \in A^1}{\arg\max} Q^1(a^1), \cdots, \underset{a^n \in A^n}{\arg\max} Q^n(a^n))$ corresponded to a pure Nash equilibrium.

On Figure 1 we have drawn a tabular version of the algorithm that simulates reinforcement learning driven agents interacting with a matrix game. The number of games after which the simulation should be stopped (step 7 of the algorithm) depends on the use desired of the algorithm. For example, one may be interested in studying the dynamics of the system for a predefined number of games, or to simulate it until the different agents have learned a rational behavior.

[1] Agent i uses a time-invariant policy if its probability of selecting action $a^i \in A^i$ is constant for all t.

Fig. 1. Simulation of reinforcement learning agents interacting with a matrix game

```
1] Set $t = 0$.
2] Initialize $Q^i(a^i) = 0 \; \forall i \in \{1, \cdots, n\}$ and $\forall a^i \in A^i$.
3] $t \leftarrow t + 1$.
4] Select for each agent $i$ an action $a^i_t$ by using an $\epsilon$-Greedy policy.
5] Play the game with the joint actions $(a^1_t, \cdots, a^n_t)$.
5] Observe for each agent $i$ the reward $r^i(a^1_t, \cdots, a^n_t)$ it has obtained.
6] Update for each agent $i$ its $Q^i$-function according to
   $Q^i(a^i_t) \leftarrow Q^i(a^i_t) + \alpha^i_t(r^i(a^1_t, \cdots, a^n_t) - Q^i(a^i_t))$
7] If a sufficient number of games have been played, then stop. Otherwise, return to step 3.
```

III. Market structure and corresponding matrix game

A. Market structure

We assume that the energy can only be traded through a spot market (no bilateral agreements, etc.) where the suppliers submit to the ISO how much they are willing to produce for a certain price. We suppose that we are dealing with a power system in which we have $nbGen$ generators (G_1, $\cdots$, G_{nbGen}) having constant marginal costs (M_{G_1}, $\cdots$, $M_{G_{nbGen}}$), $nbNodes$ nodes (1, $\cdots$, $nbNodes$) and inelastic and constant loads. We suppose that each supplier (generator) G_i always bids its full generation capacity $P^{max}_{G_i}$ at a constant price per MW produced. This procedure conforms to the so-called "block bids"[14], where the only simplification we make is that generators always bid their full capacity $P^{max}_{G_i}$. We also assume that the generators are not allowed to bid higher that a price cap and denote by b_{G_i} (\$/MW) the bid that generator G_i submits to the ISO.

The ISO collects all bids and is then in charge of clearing the market by minimizing the sum of the production costs while satisfying network constraints. To realize this objective, the ISO solves the following linear programming problem:[2]

Determine

$$(P_{G_1}, \cdots, P_{G_{nbGen}}, \theta_1, \cdots, \theta_{nbNodes}) \in R^{nbGen+nbNodes}$$

that minimizes

$$\sum_{G_i} b_{G_i} P_{G_i} \tag{4}$$

subject to the constraints

$$P_{load}(k) = P_{produced}(k) + \sum_{j=l}^{nbNodes} y_{kl}(\theta_l - \theta_k)$$
$$P_{G_i} \leq P^{max}_{G_i}$$
$$|y_{kl}(\theta_k - \theta_l)| \leq P^{max}_{kl}$$

Here P_{G_i} denotes the power injected by generator G_i, θ_k the voltage angle at node k, P^{max}_{kl} the maximum flow allowed in the line connecting node k to node l, y_{kl} the admittance of the line connection node k to node l, and $P_{load}(k)$ ($P_{produced}(k)$) the power consumed (injected) at node k.

By solving this linear programming problem, the ISO can determine the power each generator G_i should be dispatched (P_{G_i}), and through the knowledge of the Lagrangian multipliers associated with this optimization problem, the nodal prices at each node k of the system.[3] We denote by n_{G_i} the nodal price at the node at which generator G_k is connected. After the market is cleared, each generator G_i is dispatched P_{G_i} and is paid n_{G_i} by MW produced.

B. The matrix game

Generator G_i submits its bid in order to maximize its reward, which is equal to the money it is paid for producing electricity P_{G_i} MW ($n_{G_i}P_{G_i}$) minus the money it has to pay for producing this quantity of power ($MC_{G_i}P_{G_i}$). The power dispatched to a generator G_i and the nodal prices are a function of the different bids submitted by the generators. Therefore, the reward of each generator is also a function of the joint bids submitted to the ISO. If we assume that each generator G_i can only choose a bid which belongs to the finite set B_{G_i}, we face a matrix game wherein:

- we have $nbGen$ agents G_1, $\cdots$, G_{nbGen}
- the finite action set for agent G_i is B_{G_i}
- the reward function for agent G_i is given by

$$n_{G_i}(b_{G_1}, \cdots, b_{G_n})p_{G_i}(b_{G_1}, \cdots, b_{G_n}) - P_{G_i}(b_{G_1}, \cdots, b_{G_n})MC_{G_i}$$

C. Nash equilibria versus reinforcement learning to study electricity markets

We have seen in the previous subsection that a spot market may be seen as a matrix game in which the agents are the power producers. One may wonder whether reinforcement learning is better suited to analyze such electricity markets than Nash equilibria. In the following, we discuss several points that attempt to answer this question:

The notion of information. In an electricity market, the different agents in principle know neither the set of actions the other agents have at their disposal nor the different reward functions r^i. In this context, it is difficult to assume that the players are indeed going to adhere to an Nash equilibrium since they do not have the elements to compute it. On the other hand, reinforcement learning does not use any information to compute the agents' behavior that is not indeed available to the agents. At each stage of the market, an agent i corrects its Q^i-function only by using its old estimate and the reward it has observed.

Multiplicity of Nash equilibria. Even if we admit that the different agents will indeed in the real world adhere to a Nash

[2] The problem is a linear optimization problem because we assume a DC representation of the transmission network. Furthermore the ISO does not consider any security criteria such as $N - 1$ criteria.

[3] The nodal price at node k may be seen as the price for extracting one additional MW at this node.

equilibrium, there may exist different Nash equilibria. Even if there exists a large literature on equilibrium refinements, which defines criteria for selecting from among multiple equilibria (such as perfect equilibria, proper equilibria, sequential equilibria, etc.), the choice of the right equilibrium may prove to be ill-defined.

Lack of studies of Q-learning for market modeling. If the Q-learning algorithm possesses a firm foundation in the theory of Markov Decision Processes (MDP), results concerning its properties in a multi-agent environment are much poorer. Moreover, to our knowledge, no experimental results have shown that by using this type of algorithm to model the behavior of the objective-oriented agents of a market we may reproduce the reality.

Computational burdens. While it may be computationally expensive to identify Nash equilibria, especially when one deals with a large number of players, computational burdens associated with Q-learning algorithms may seem much lighter, since for a fixed number of games played they grow only linearly with the number of players. However, it is difficult to determine how many games have to be played before obtaining, if ever obtained, a behavior for the agents that may be call "rational". In this respect, one can perfectly imagine that for electricity markets the number of games having to be played before obtaining some rational behavior tends to grow exponentially with the number of power producers which could make the reinforcement learning approach computationally much less efficient than the Nash equilibria approach.

IV. Case Studies

A. Test market description and simulation conditions

We have carried out simulations on the power system sketched on Figure 2, whose topology is similar to the Pennsylvania-New-Jersey-Maryland (PJM) five node power system [15].

The market is cleared according to the procedure detailed in the previous section, and the price cap for this market is set equal to 50 $/MW.

This system has four loads and three generators. The loads are assumed to be inelastic and constant, and every generator G_i is assumed to have a maximum production capacity of $P_{G_i}^{max}$, a constant marginal cost MC_{G_i} and a finite bid set B_{G_i}. The values of these production limits and these marginal costs as well as the description of these bid sets are given in Table I. Note that the lowest bid of each generator is equal to its marginal cost, while its highest possible bid equal to the price cap.

The line connecting nodes 2 and 5 can only transfer 100 MW, and as a result may be subjected to congestion. For the other lines of the system, we suppose that there exist no power dispatches that may lead to flows greater than their transfer capacity.

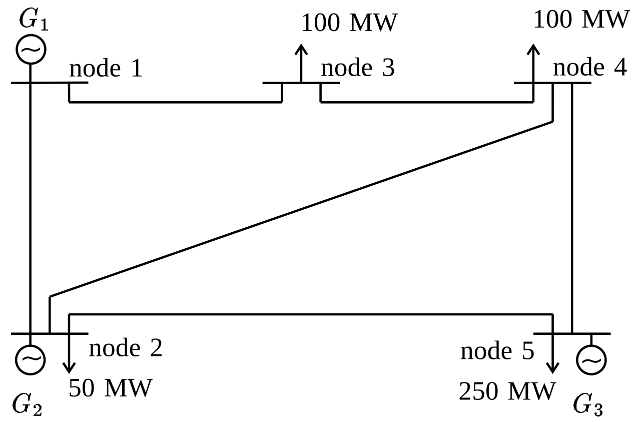

Fig. 2. Power system description

	$P_{G_i}^{max}$ [MW]	MC_{G_i} [$/MW]	B_{G_i} [$/MW]
G_1	300	20	{20, 30, 40, 50}
G_2	300	20	{20, 30, 40, 50}
G_3	250	30	{30, 40, 50}

TABLE I
Generation data and bid sets

We consider in our simulations two different cases. In the first case, we suppose that only generators G_1 and G_3 behave as active agents,[4] while G_2 always bids at its marginal cost.[5] In the second case, all three generators are considered as being active agents. For each case we simulate the market dynamics when the active agents are modeled through reinforcement learning algorithms (see Figure 1), and compute the different pure Nash equilibria. When using reinforcement learning algorithms, the update of the different Q^i-functions of the agents depends on the value of the parameters α_t^i. These parameters have been chosen in our simulations equal to 0.1 $\forall i, t$. Furthermore, the value of ϵ, the parameter that determines the degree of randomness in the action selection process, has been chosen to be 0.1 for all agents. This means that the agents select the action that maximizes their Q^i-function with a probability of 0.9 and with a probability of 0.1 an action at random.

B. Two generators behaving as active agents

In this case G_2 always bids at its marginal cost of 20 $/MW while the other two generators G_1 and G_3 are active agents.

In Figure 3 we have represented the evolution of the Q-function for G_3. Each curve drawn on this figure represents the evolution of the expected reward G_3 believes it will obtain by submitting a certain bid to the market. As one may observe, G_3 learns after less than 100 market clearings that to obtain the highest rewards it should bid at the price cap (50 $/MW). Obviously, generator G_3 'realizes' its advantaged position in the network. Indeed, it is connected to a node where the power consumption is equal to 250 MW and, due to the limited transfer capacity of the line connecting nodes 2 and

[4] By active agent, we mean an agent that selects its actions in order to maximize its rewards.

[5] or equivalently that all the generators are active agents and that B_{G_2} is now equal to $\{20\}$

5, all this power cannot come from G_1 and G_2. Therefore, in order to cover the load it needs to be dispatched, and uses this market power to sell its energy at the highest price. The spikes observed in the evolution of the different curves drawn in Figure 3 result from the ϵ-greedy strategies used by the different agents of the system. Even if all the active generators are able to assess quite clearly which bid is going to lead to the highest reward, they select a bid totally at random one out of every ten times and may therefore deviate from their optimal strategies. This may modify the power dispatches and the nodal prices and "perturb" therefore the previous estimates of the different Q-functions. The generator G_1 learns that its best strategy is to bid at the price cap (50 $/MW) (see Figure 4). However, now the learning is slower, since only after approximately 800 clearings of the market does 50 $/MW become the bid that maximizes its Q-function.

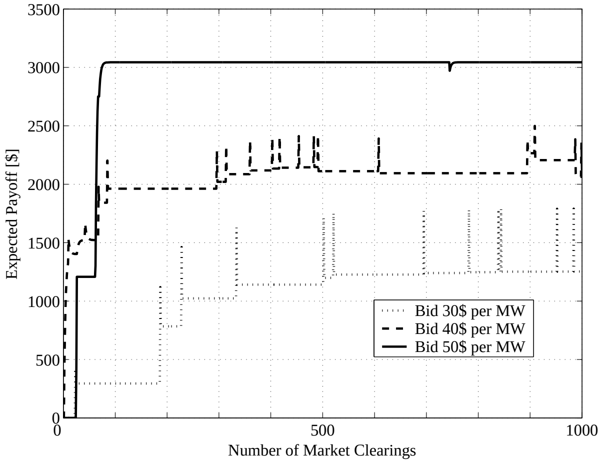

Fig. 3. Evolution of the Q-function for G_3 (2 active agents)

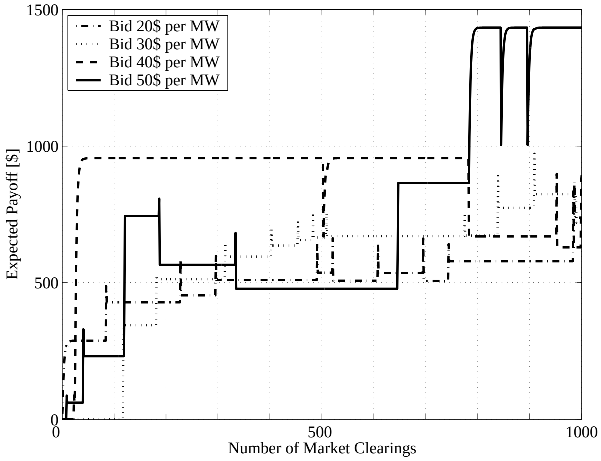

Fig. 4. Evolution of the Q-function for G_1 (2 active agents)

Table II gathers the bids that would have been submitted by G_1, G_2 and G_3 just after 1000 clearings of the market if G_1 and G_3 (the active agents) would indeed submit their greedy bids. On the same table one may also view the corresponding power dispatches, nodal prices and rewards. Although with such power dispatches the line connecting nodes 2 and 5 is congested, we have the same nodal prices, as the next MW will either be produced by G_1 or G_3, both having submitted bids equal to 50 $/MW.

	b_{G_i} [$/MW]	P_{G_i} [MW]	n_{G_i} [$/MW]	Reward [$]
G_1	50	48	50	1440
G_2	20	300	50	9000
G_3	50	152	50	3040

TABLE II

MARKET INPUT AND OUTPUT WHEN AFTER 1000 OF MARKET CLEARINGS THE GENERATORS SELECT THEIR GREEDY BIDS.

To determine the pure Nash equilibria of the market, we clear the market for all the bids $(b_{G_1}, b_{G_2}, b_{G_3}) \in B_{G_1} \times \{20\} \times B_{G_3}$ in order to compute the reward functions for generators G_1 and G_3, and then determine by explicit search the bids (b_{G_1}, b_{G_3}) that satisfy expression (1). The reward functions for G_1 and G_2 are provided in Table III. To explain the meaning of the different elements of this table, let us say that the cell containing the two elements 477 3040 gathers the value of the rewards obtained by G_1 (477) and G_3 (3040) when G_1 bids 30 $/MW and G_3 50 $/MW. By analyzing this table, one can observe that there exists a unique pure Nash equilibrium $((b_{G_1}, b_{G_3}) = (50, 50))$. In this case, the greedy strategies learned by G_1 and G_3 "converge" to a Nash equilibrium.

	30 $/MW		40 $/MW		50 $/MW	
20 $/MW	144	0	288	1401	431	2802
30 $/MW	478	0	478	1522	477	3040
40 $/MW	0	0	956	1522	956	3040
50 $/MW	0	0	0	2000	1440*	3040*

TABLE III

REWARD FUNCTIONS WHEN G_1 AND G_2 ARE THE ONLY ACTIVE AGENTS

C. *Three generators behaving as active agents*

In the previous subsection, only generators G_1 and G_2 were modeled as active agents. In this case G_3 also bids actively to the spot market.

Under such conditions, we have observed that the evolution of the Q-function for G_3 was quite similar to the evolution observed in the previous case analyzed (see Figure 3), and so G_3 learns that it has market power and that it should bid at the price cap (50 $/MW) to maximize its reward. However, the evolution of the Q-function for G_2 now exhibits a completely different behavior. If when only G_2 and G_3 were active agents, we observed (see Figure 4) that the Q-function learned by G_2 was clearly indicating that the bid 50 $/MW was the greedy action (the greedy action being the action that maximizes the Q-function), it is no longer the case here (see Figure 5). Indeed the greedy action always changes. Furthermore, the evolution of the Q-function seems now to be driven by a chaotic process.

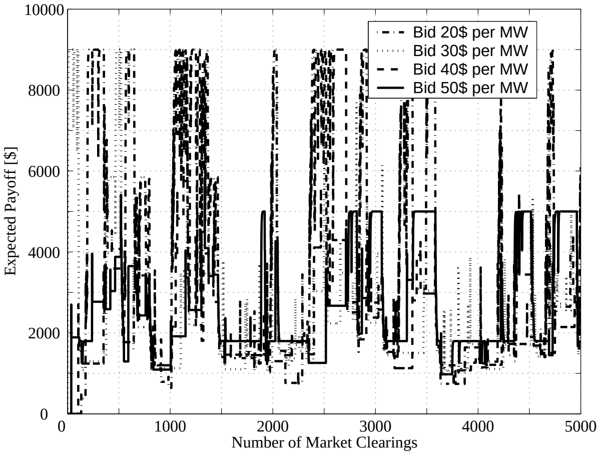

Fig. 5. Evolution of the Q-function for G_2 (3 active agents)

Note that for G_1, the same type of chaotic evolution has also been observed.

If in the previous case there was only one single pure Nash equilibrium, now two pure Nash equilibria exist. These two pure joint strategies are $(b_{G_1}, b_{G_2}, b_{G_3}) = (20, 50, 50)$ and $(b_{G_1}, b_{G_2}, b_{G_3}) = (50, 20, 50)$. This may eventually explain why the Q-functions of generators G_1 and G_2 evolve in such a chaotic way. Indeed, one may reasonably suppose that these generators are in some sense unable to decide around which Nash equilibrium their greedy strategies should stabilize.

V. Conclusions and future work

In this paper we have considered a spot market as being an n-agent matrix game, and used two types of approaches to analyze its characteristics. One approach directly computes the Nash equilibria of the market, and the other models each agent of the game through reinforcement learning algorithms and simulates the market dynamics so created to analyze its characteristics. These two approaches have been illustrated on some examples, and the similarities that exist between the results they generate have been highlighted.

One future research work would be to explore more carefully the properties of the Q-learning algorithm used in a multi-agent framework. In this respect, there exists in our opinion several research directions. One is to determine for which matrix games the different agents modeled with Q-learning algorithms could exhibit, after a certain learning time, a rational behavior. Under the assumption that the agents can indeed learn a rational behavior, it could also be useful to be able to assess the speed at which they learn it. Another direction is to further investigate the relationships that exist between Q-learning and Nash equilibria. For example, it might be interesting to determine under exactly which conditions the learned joint strategies converge to a Nash equilibrium.

All the simulations carried out in this paper were made under some strong assumptions concerning the market structure (the energy was only traded through a spot market) and the power system itself (no reliability problems, constant and inelastic load, etc.). In this respect, it would be particularly relevant to experiment reinforcement learning algorithms in a more realistic setup. This would also allow us to compare our simulation results with the data gathered on real markets, and thereby determine whether reinforcement learning can indeed be an efficient tool to reproduce the "real-world" behavior of some objective oriented agents.

References

[1] F. Schweppe, M. Caramanis, R. Tabors, and R. Bohn, *Spot Pricing of Electricity*. Kluwer Academic Publisher, 1988.

[2] J. Cardell, C. Hitt, and W. Hogan, "Market Power and Strategic Interaction in Electricity Networks," *Resource and Energy Economics*, vol. 19, pp. 109–137, 1997.

[3] C. Berry, B. Hobbs, W. Meroney, R. O'Neill, and W. Stewart Jr, "Understanding how market power can arise in network competition: a game theoretic approach," *Utilities Policy*, vol. 8, pp. 139–158, 1999.

[4] B. Hobbs, "Linear Complementarity Models of Nash-Cournot Competition in Bilateral and POOLCO Power Markets," *IEEE Transactions on Power Systems*, vol. 16, no. 2, pp. 194–202, May 2001.

[5] D. Bunn and F. Oliveira, "Agent-Based Simulation - An Application to the New Electricity Trading Arrangements of England and Wales," *IEEE Transactions On Evolutionary Computation*, vol. 5, no. 5, pp. 493–503, October 2001.

[6] D. Ernst, A. Minoia, and M. Ilic, "Market dynamics driven by the decision-making of power producers," in *Proceedings of Bulk Power System Dynamics and Control - IV Managing Complexity in Power Systems: From Micro-Grids to Mega-Interconnections*, August 2004.

[7] C. Day and D. Bunn, "Divestiture of Generation Assets in the Electricity Pool of England and Wales: A Computational Approach to Analyzing Market Power," *Journal of Regulatory Economics*, vol. 19, no. 2, pp. 123–141, 2000.

[8] M. Littman, "Markov games as a framework for multiagent reinforcement learning," in *Proceedings of the Eleventh International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann, 1994.

[9] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton, New Jersey: Princeton University Press, 1947.

[10] C. Lemke and J. Howson, "Equilibrium points of bimatrix games," *Journal of the Society of Industrial and Applied Mathematics*, vol. 12, pp. 413–423, 1964.

[11] J. Rosenmuller, "On a generalisation of the Lemke-Howson algorithm to noncooperative n-person games," *SIAM Journal of Applied Mathematics*, vol. 21, pp. 73–79, 1971.

[12] C. Watkins, "Learning from Delayed Rewards," Ph.D. dissertation, Cambridge University, Cambridge, England, 1989.

[13] J. Hu and M. Wellman, "Nash Q-learning for General-Sum Stochastic Games," *Journal of Machine Learning Research*, vol. 4, pp. 1039–1069, 2003.

[14] "European Energy Exchange - EEX, Market Model," Available Online: http://www.eex.de/spot_market/info/market_model/index_e.asp.

[15] "PJM - Interconnection, Presentation - FTR Auction Example," Available Online: http://www.pjm.com/services/training/downloads/lmpftr2.pdf.