A COMPARISON OF NASH EQUILIBRIA ANALYSIS AND AGENT-BASED MODELLING FOR POWER MARKETS

Th. Krause, G. Andersson
Swiss Federal Institute of Technology
Zürich, Switzerland
{krause, andersson}@eeh.ee.ethz.ch

D. Ernst
University of Liège
Postdoctoral Researcher FNRS
dernst@ulg.ac.be

E. V. Beck, R. Cherkaoui, A. Germond
École Politechnique Fédérale de Lausanne
Lausanne, Switzerland
{elena.vdovina, rachid.cherkaoui}@epfl.ch

Abstract - In this paper we compare Nash equilibria analysis and agent-based modelling for assessing the market dynamics of network-constrained pool markets. Power suppliers submit their bids to the market place in order to maximize their payoffs, where we apply reinforcement learning as a behavioral agent model. The market clearing mechanism is based on the locational marginal pricing scheme. Simulations are carried out on a benchmark power system. We show how the evolution of the agent-based approach relates to the existence of a unique Nash equilibrium or multiple equilibria in the system. Additionally, the parameter sensitivity of the results is discussed.

Keywords - Electricity market modelling, multi-agent modelling, game theory, matrix games, reinforcement learning, spot markets.

1 INTRODUCTION

In the early 1990's the power supply industries worldwide started to undergo a period of extensive changes. Electricity markets moved away from vertically integrated monopolies towards liberalized structures with power delivery being a bundle of several services mainly including generation, transmission and distribution. The main reason for restructuring is related to the expectation that competition could lead to a reduction of electricity prices and could stimulate the emergence of new technologies. However, several national markets (e.g. in California, the United Kingdom and Spain) were suspected to allow for 'gaming' and the exercise of market power. Thus, electricity markets have been re-reorganized and will continue to be subject to structural changes, as observed with the recent introduction of the New Electricity Trading Arrangements (NETA) in the UK and the upcoming inauguration of a market regulator in Germany. Ideally, the effects of such market restructuring proposals should be known prior to their implementation. Hence, there is a need for appropriate modelling and analysis concepts, where at least four distinct approaches can be distinguished [1]: a) *ex post* analysis of existing markets, b) market concentration analysis using current market data, c) equilibria analysis, and d) multi-agent modelling, where either individuals are interacting or artificial agents. The above concepts may be used to study effects concerning market concentration, efficiency, and market power. Nevertheless, in [1] it is pointed out that the different concepts are significantly sensitive to the underlying assumptions, the choice of the behavioral agent-models and the set of parameters used for the algorithms. Bunn and Oliveira in [2] state "that with the process of daily experimentation and learning of the market players multiple transient equilibria are likely to occur", where it has to be investigated how the different concepts 'cope' with this constellation.

The contribution of this paper is a comparison of Nash equilibria analysis and agent-based modelling in conjunction with reinforcement learning for a network-constrained pool market. We show the interdependencies of the two approaches, i.e. we focus on the assessment of the market dynamics obtained through an agent-based model with respect to the existence of Nash equilibria in the system. This paper is a further development of [3]. For sake of consistency and clarity we outline our previous findings, but then extend our analysis and describe the parameter-dependencies of the results.

The paper is organized as follows. In section 2 we introduce matrix and repeated games, define the notion of Nash equilibrium and introduce a behavioral agent model known as Q-learning. Section 3 describes the implementation of a pool market and shows how the process of bidding to a spot market may be formalized as a repeatedly played matrix game. In section 4 we set up a benchmark electricity market and discuss the simulation results obtained. Eventually, section 5 concludes the paper.

2 MATRIX GAMES, NASH EQUILIBRIUM AND AGENT-BASED MODELLING

2.1 Matrix Games and Repeated Play

Game theory is a branch of economic science focusing on the behavior related to interactive decision making problems. There are a vast variety of games that are analyzed in depth in literature (e.g. [4, 5]) and several types of games have been used by electricity market researchers (e.g. [6, 7]). In this paper, we consider non-cooperative games played repeatedly a finite number of times. First we outline the basic matrix game in a normal form defined through:

- a set of n agents $\{1, \cdots, n\}$
- $A_1, \cdots, A_n$ finite sets of pure *actions* available to the agents (A_i is the space of actions for agent i)
- p_i denotes the mixed strategy used by agent i to select its actions. $p_i(a_i)$ represents the probability for agent i to select action $a_i \in A_i$. A pure strategy is a degenerate case of a mixed strategy for which

$\exists a_i \in A_i$ such that $p_i(a_i) = 1$. $p = (p_1, \cdots, p_n)$ denotes the strategy profile for the matrix game.

- $r_i : A \rightarrow R$ is the reward function of the stage game for agent i where $A = A_1 \times \cdots \times A_n$. For mixed strategy case the expected reward is calculated as:

$$r_i(p_1, \cdots, p_n) = \sum_{a \in A_i} p_1(a_1) * \cdots * p_n(a_n) * r_i(a_1, \cdots, a_n) \quad (1)$$

where $a = (a_1, \cdots, a_n)$. In the repeated game repetition means that exactly the same single stage game is played a certain number of times. [8] The space of actions and corresponding payoffs is kept invariant. The choice of strategy might be influenced by the history of the game.

- $t \in \{1, \cdots, T\}$ refers to a particular period of the game.
- $a^t = (a_1^t, \cdots, a_n^t)$ is the action profile being played at t.
- Let $h^t = (a^1, a^2, \cdots, a^{t-1})$ denote a specified history of the game at period t (in other words it is the collections of actions that have been chosen in all previous iterations by all the agents).
- s_i denotes the mixed strategy used by agent i to select its actions. S_i is the set of possible mixed strategies for agent i. This strategy may be such that the probability to select an action at time t may depend on the history of the game h^t.[1] $s = (s_1, \cdots, s_n)$ denotes the repeated game strategy profile.
- The payoff of each agent is a weighted cumulative sum of payoffs it obtains in every period:[2]

$$u_i = r_i^1 + \delta r_i^2 + \cdots + (\delta)^{T-1} r_i^T = \sum_{t=1}^{T} (\delta)^{t-1} r_i^t \quad (2)$$

where δ is a discount factor (commonly a "time" factor). A discount factor close to 0 means that the agent puts most weight on the payoffs from the first periods (impatient about near-future profits). If this factor is close to 1 than the player is rather indifferent between the outcomes of any rounds. It does not affect much our discussions because the analysis of results is mostly based on winning strategies rather than on cumulative payoff's comparison.

2.2 Nash equilibrium

The fundamental solution concept in game theory is a *Nash equilibrium* (NE) point where each agent's strategy is a best response to the strategies of the others. A player has no motivation to deviate from NE strategy since it would lead to a decrease of its expected payoff. Nash equilibrium of the stage game is formally defined as follows: *The strategy profile $p^* = (p_1^*, \cdots, p_n^*)$ is a Nash equilibrium if for all $i \in \{1, \cdots, n\}$ we have*

$$r_i(p_1^*, \cdots, p_n^*) \geq r_i(p_1^*, \cdots, p_{i-1}^*, p_i, p_{i+1}^*, \cdots, p_n^*) \quad (3)$$

Several algorithms have been developed for computing Nash equilibria. The interested reader may refer to [3, 9]. In the case of finite repeated games the subgame consists of a sequence of single stage-game equilibria. *The repeated game strategy profile s^* is a subgame-perfect Nash equilibrium if for all $i \in \{1, \cdots, n\}$ we have*

$$s_i^* \in \arg\max_{s_i \in S_i} r_i(s_i, s_{-i}^*). \quad (4)$$

If there is a unique stage-game equilibrium then it is repeated over whole game.

For the particular problems studied in this paper we have only observed the presence of pure stage-game Nash equilibria (see Section 4). Since the action spaces A^i are finite in our examples, these Nash equilibria at every stage were computed by enumeration of all n-tuples of A and selection of those which were satisfying equation (3).

2.3 Agent-Based Modelling and Reinforcement Learning

Most economies incorporate a large number of market participants (also referred to as agents) interacting locally with each other by, e.g. selling or buying goods, where every participant may follow a set of individual objectives. This interaction on the micro-level determines to a large extent the overall market dynamics, i.e. the evolution of market characteristics, such as market prices, price volatility, overall trading volume etc. Hence, we observe a feedback between the micro- and the macro-level of markets.[3] One concept to account for this feedback is agent-based computational economics, where systems are described through a bottom-up approach by modelling the different market participants and letting them interact within a defined macro-structure. In section 3 we will describe the macro-structure of the studied electricity market, whereas in this section we outline reinforcement learning as one concept to be applied for the behavioral modelling of the agents.

Reinforcement learning is the problem faced by an agent that learns behavior from experience acquired from interaction with its environment (see [10] for a survey). In the context of reinforcement learning, we suppose that the matrix game defined in section 2.1 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

[1] In this work we consider a particular class of repeated-game strategies such us an *open-loop* strategy. This is a simple class of *history-independent dynamic* games.

[2] $(\delta)^t$ refers to δ to the power of t while r_i^t refers to the reward observed by agent i at time t.

[3] The feedback is mutual. Changes within the macro-structure, e.g. trading protocols, quotas, etc. will certainly influence the micro-level as the market players may adopt to the respective changes by modifying their objectives.

matrix games the well-known Q-learning algorithm [11], which was initially designed for learning through interaction with a Markov Decision Process. There are several papers that discuss extensions of Q-learning algorithm to various types of games and study the conditions under which the behavior of the players converge to a Nash equilibrium [12], [13].

When an agent i is modelled 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 high 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 action $(a_1^t, \cdots, a_n^t)$ represents the actions the different agents have taken. After the game is played and the different rewards r_i have been observed, agent i updates its Q_i-function according to the following expression:

$$Q_i(a_i^t) \leftarrow Q_i(a_i^t) + \alpha_i^t(r_i(a_1^t, \cdots, a_n^t) - Q_i(a_i^t)) \quad (5)$$

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 $a_i = a_i^t$ 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.

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 4), 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.

Figure 1 shows 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 8 of the algorithm) depends on the purpose of the study. 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] 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 ϵ-Greedy policy.
5] Play the game with the joint actions $(a_1^t, \cdots, a_n^t)$.
6] Observe for each agent i the reward $r_i(a_1^t, \cdots, a_n^t)$ it has obtained.
7] Update for each agent i its Q_i-function according to
$Q_i(a_i^t) \leftarrow Q_i(a_i^t) + \alpha_i^t(r_i(a_1^t, \cdots, a_n^t) - Q_i(a_i^t))$

8] If a sufficient number of games has been played, then stop. Otherwise, return to step 3.

Figure 1: Simulation of reinforcement learning agents interacting with a matrix game

2.4 *Agents Use Subgame-Perfect Nash Equilibria to Select Actions*

Later in this paper, we will suppose that the different sets S_i are composed only of history-independent strategies and that the agents play T times the matrix game and use the knowledge of the subgame-perfect Nash equilibria of the corresponding repeated game to select their strategies. If the matrix game has just one single Nash equilibrium p^*, there is only one subgame-perfect Nash equilibrium. Therefore, by using the knowledge of the subgame-perfect Nash equilibrium to select at period t its action, agent i will choose an action according to the mixed strategy p_i^*. Now, if the matrix game has $nbEq$ Nash equilibria, it implies that there are T^{nbEq} subgame-perfect Nash equilibria. We suppose in this case that every agent selects at random one of these subgame-perfect Nash equilibria to determine its strategy. By proceeding like this, agents will not necessarily have strategies which correspond to the same subgame-perfect Nash equilibrium and do not seek to select equilibria having some particular properties (e.g. Pareto optimality). Note that selecting at random a subgame-perfect Nash equilibrium or selecting T times at random a Nash equilibrium of the stage game are two "equivalent things". Therefore, we may consider that, when using subgame-perfect Nash equilibria to select its actions, agent i selects at every t a Nash equilibrium p^* at random and play an action according to the mixed strategy p_i^*.

3 MARKET STRUCTURE AND CORRESPONDING MATRIX GAME

3.1 *Market Structure*

We assume a mandatory spot market, where the suppliers submit bids in the form of linear marginal price functions. Besides the spot market no other transactions are allowed (no bilateral agreements etc.). We suppose dealing with a power system in which we have $nbGen$ generators (G_1, $\cdots$, G_{nbGen}), $nbNodes$ nodes (1, $\cdots$,

$nbNodes$) and inelastic and constant loads. Below the decision problem of the power suppliers (generators) is outlined, where we assume linear marginal cost for the suppliers.

3.2 Decision Problem of the Power Suppliers

In contrast to perfectly competitive markets where participants are assumed to be price takers and prices are equal to the marginal cost of supply we assume in our model an oligopoly market. Thus, suppliers may bid strategically above their marginal cost as they realize their possible influence on market prices. Subsequently, we consider that generators may deviate their bids from marginal cost (unknown to the outside world) to increase their profits where in [1] two ways of deviating are discussed: a) changing the slope s_{G_i} of the submitted function or b) changing the intercept ic_{G_i}. In our model the latter choice is implemented, generators only manipulate the intercept of their bid function. The line of argument follows the description in [1]: "Slopes of marginal cost function for individual generators are usually very shallow, so the very steep slopes that would result from manipulating s would not be credible. [...]". To manipulate the intercept ic_{G_i} generators set a certain markup mup_{G_i} in order to maximize their payoffs (see figure 2).

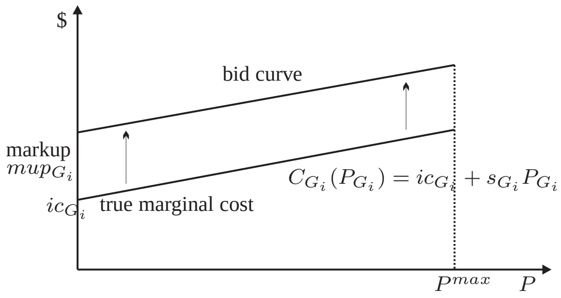

Figure 2: True Marginal Cost and Markup

3.3 Optimization Problem of the Independent System Operator

Above it was described that generators will submit a linear marginal cost or a parallel translated function (determined by the markup) to show their willingness to supply. 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 quadratic programming problem[4]
Determine

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

that minimizes

$$\sum_{G_i} \frac{1}{2} P_{G_i} \mathrm{diag}(s_{G_i}) P_{G_i} + ic_{G_i} P_{G_i}$$

subject to the constraints[5]

$$\begin{aligned} P_{load}(k) &= \sum_{G_i} P_{G_i}(k) + \sum_{nbNodes} y_{kl}(\theta_l - \theta_k) \\ P_{G_i} &\le P_{G_i}^{max} \\ |y_{kl}(\theta_k - \theta_l)| &\le P_{kl}^{max} \end{aligned}$$

Here P_{G_i} denotes the power injected by generator G_i, θ_k the voltage angle at node k, P_{kl}^{max} 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 $P_{load}(k)$ the power consumed at node k.

By solving this quadratic 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 are given.[6] We denote by n_{G_i} the nodal price at the node at which generator G_i is connected. After the market is cleared, each generator G_i is dispatched P_{G_i} and is paid n_{G_i} per MW produced.

3.4 Corresponding matrix game.

In our problem the one-stage matrix game consists of:

- $nbGen$ active agents $(G_1, G_2, \cdots, G_{nbGen})$ (the generators)
- their corresponding finite sets of pure actions A_{G_i}
- corresponding reward functions r_{G_i} that are actually functions of joint actions of all participants since the power dispatch and nodal prices depend on bid submitted by every generator. The reward function r_{G_i} is defined by:
$$r_{G_i} = n_{G_i} \cdot P_{G_i} - MC_{G_i} \cdot P_{G_i} \tag{6}$$
where P_{G_i} is a dispatched quantity for generator G_i, n_{G_i} is its nodal price and MC_{G_i} marginal cost of production.

4 CASE STUDIES

4.1 Test Market Description and Simulation Conditions

We have carried out simulations on the power system shown in figure 3. The market is cleared according to the procedure detailed in the previous section. 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 linear marginal cost function $C_{G_i}(P_{G_i}) = ic_{G_i} + s_{G_i} \cdot P_{G_i}$ and a finite set of markups mup_{G_i}. The values of these production limits and these marginal cost functions as well as the description of these sets of markups are given in Table 1. Note that the lowest markup of each generator is

[4] In this paper we consider the spot market to be operated in one hour intervals. Thus, we simplify the notation by writing MW instead of MWh.
[5] The constraints represent a power flow using the usual DC power flow approximations.
[6] The nodal price at node k may be seen as the price for extracting one additional MW at this node.

zero, while its highest possible markup is set to not exceed the price cap of 60$/MW at any possible production level. 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. The numbers close to the lines denote the value of their reactance expressed in pu.

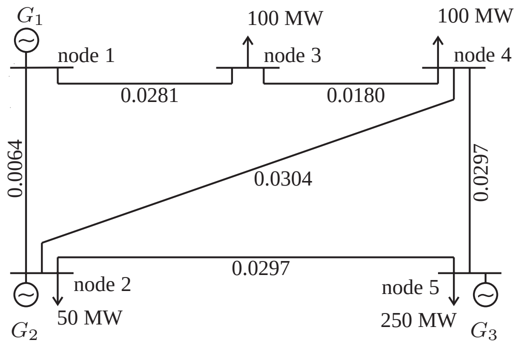

Figure 3: Power System Description

	$P_{G_i}^{max}$ [MW]	ic_{G_i} [$]	s_{G_i} [$/MW]	mup_{G_i} [$]
G_1	300	10	0.02	{0, 10, 20, 30}
G_2	300	10	0.02	{0, 10, 20, 30}
G_3	250	20	0.04	{0, 10, 20}

Table 1: Generation Data and Sets of Markups

We consider two different cases in our simulations. In the first case, we suppose that only generators G_1 and G_3 behave as active agents,[7] while G_2 always bids its marginal cost function to the ISO. 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 modelled through reinforcement learning algorithms (see Figure 1), and discuss several characteristics of this dynamics at the light of the information derived from the Nash equilibria analysis, i.e the direct computation of 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_i^t. We will first carry out our simulations with these parameters set to $0.1\ \forall i, t$. Furthermore, the value of ϵ, the parameter that determines the degree of randomness in the action selection process, is initially chosen equal to 0.1 for all agents. This means that each agent selects the action that maximizes its Q_i-function with a probability of 0.9 and with a probability of 0.1 an action at random.

4.2 Two Generators Behaving as Active Agents

In the following, for assessing our case studies we will distinguish between the agent-based model and Nash equilibria analysis. We will outline both approaches in separate paragraphs and then compare the results obtained focussing on the interdependencies between the two concepts. For the present case with two generators being modelled as active agents, we start with the Nash equilibria analysis.

Nash Equilibria Analysis

For computing the Nash equilibria of the market we clear the market for all combinations of bids (determined by the respective markups chosen by each generator). Thereby, we compute the reward functions for G_1 and G_3, and the corresponding results are gathered in Table II. We then explicitly search for the bids (and thus for the markups) which satisfy expression (3). Table 2 follows the layout of a payoff table as generally used in game theory to describe matrix games. In the present case, G_1 is the row player and G_3 the column player. As an example, if G_1 chooses a markup of 30$ and G_3 sets the markup to 20$ the reward of G_1 will be 1430$ and respectively 3050$ of G_3.[8]

	0$		10$		20$	
0$	140	0	290	1400	430	2800
10$	480	0	480	1520	480	3050
20$	0	0	1000	1520	1000	3050
30$	0	0	0	2000	1430*	3050*

Table 2: Reward functions when G_1 and G_3 are the only active agents

Agents use subgame perfect Nash equilibria to select actions

Now if we consider that the agents select actions from the knowledge of the subgame perfect Nash equilibria and this according to the procedure outlined in section 2.4, it is obvious that agent G_1 will always select as action the markup of 30$ and agent G_3 the markup of 20$. Indeed, there is only one Nash equilibrium for the matrix game which implies a unique subgame perfect Nash equilibrium for the repeated game.

Agent-Based Model

Figure 4 shows the evolution of the Q-function for G_3. Each curve in this figure represents the evolution of the expected reward for the different markups. Thus, each curve shows what G_3 believes it will obtain by choosing a certain markup and submitting the resulting supply function to the ISO.

From figure 4 it can be read that G_3 rapidly learns that it should choose its highest possible markup of 20$. G_3 obviously 'realizes' its advantageous position in the network. Due to the limited transfer capacity of the line between nodes 2 and 5 and a power consumption of 250 MW at node number 5, there is a high likelihood for G_3 to be dispatched. Hence, G_3 receives market power, which it exploits by choosing the highest possible markup. G_1 learns that its best strategy is to choose a markup of 30$ (see Figure 5). In comparison to G_3 the learning is somewhat slower, since only after approximately 100 clearings of the market 20$ becomes the markup that maximizes its Q-function.

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

[8]The reward of 140$ for G_1 with both generators having selected no markup results from congestion on the line connecting node 2 and node 5. Due to the congestion the nodal price at node 1 is 20.40$/MW resulting into a reward of 140$ for G_1.

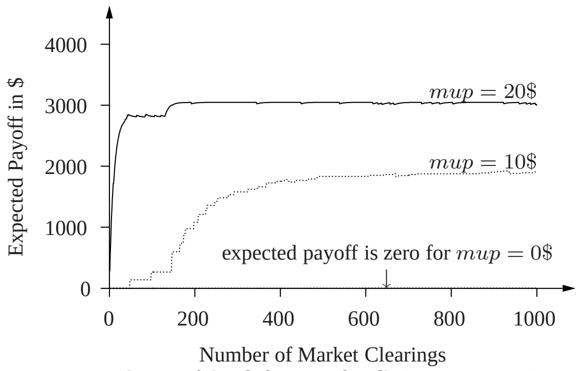

Figure 4: Evolution of the Q-function for G_3 (2 active agents)

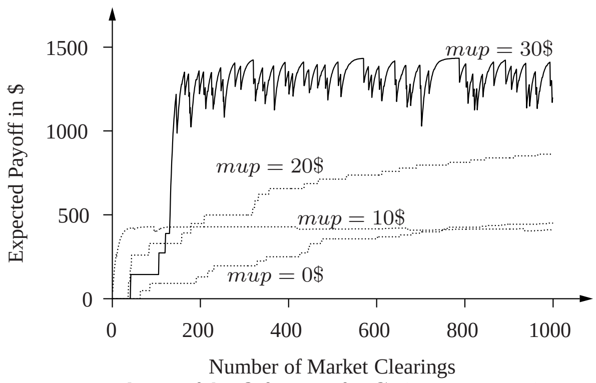

Figure 5: Evolution of the Q-function for G_1 (2 active agents)

The dips observed in the evolution of the different curves drawn in Figure 5 result from the ϵ-greedy strategies used by the different agents of the system. In one out of ten times, on the average, the generators will submit a bid (markup) totally at random. This may modify the power dispatches and the nodal prices and "perturb" therefore the previous estimates of the different Q-functions, where the perturbation influences G_1 much stronger than G_3. Table 3 gathers the information if indeed G_1 and G_3 would have submitted their greedy bid functions (determined by the respective markups). In the same table the corresponding power dispatches, nodal prices and rewards are given. Although with such power dispatches the line connecting nodes 2 and 5 is congested, we observe the same nodal prices, as the next MW will either be produced by G_1 or G_3, both manipulating the intercept of their bid functions to 50 \$ by choosing their highest markup. Although, the cost functions are not constant, the slope is so small that variations of the production level do not significantly influence nodal prices.

	mup_{G_i} [\$]	P_{G_i} [MW]	n_{G_i} [\$/MW]	Reward [\$]
G_1	30	48	50	1430
G_2	0	300	50	9000
G_3	20	152	50	3050

Table 3: Market input and output when after 1000 of market clearings the generators select their greedy bids.

4.3 *Three Generators Behaving as Active Agents*

We now assess the market dynamics with all generators being modelled as active agents. Thus, G_2 is no longer limited to bid its marginal cost function, but can now determine a markup mup_2 out of the discrete action set $\{0\$, 10\$, 20\$, 30\$\}$. We first focus on Nash equilibria analysis and then use the results obtained to describe the evolution of the system with respect to the agent-based approach.

Nash Equilibria Analysis

Consistent with the previous case we clear the market for all combination of bids and then compute the reward functions for G_1, G_2 and G_3. So we construct the payoff matrix for the one-shot game. We will restrain from presenting this table completely as it is a three dimensional matrix given by $r^1 \times r^2 \times r^3$ with r^i denoting the generators' reward functions. Searching for Nash equilibria we find the following two pure equilibrium points: (I) G_1 bidding its marginal cost function ($mup_{G_1} = 0\$$) and G_2 and G_3 choosing their highest markups of $mup_{G_2} = 30\$$ and $mup_{G_3} = 20\$$ and (II) G_2 bidding its marginal cost function ($mup_{G_2} = 0\$$) and G_1 and G_3 choosing their highest markups of $mup_{G_1} = 30\$$ and $mup_{G_3} = 20\$$. The computation shows that for this particular case there exists no equilibrium in mixed strategies. At both equilibrium points G_3 always chooses its highest markup, thus we may draw a payoff matrix assuming G_3 sets its markup mup_{G_3} to $20\$$. Table 4 displays the results. The two equilibrium points are highlighted by (*).

	0$		10$		20$		30$	
0$	430	0	3290	600	6140	1200	9000*	1800*
10$	480	2690	3290	600	6140	1200	9000	1800
20$	1000	5850	1000	5850	6140	1200	9000	1800
30$	1430*	9000*	1430	9000	1430	9000	5400	5230

Table 4: Payoff Table for G_1 (row) and G_2 (column) with G_3 choosing a markup of 20 \$

Other cells containing reward values identical to the highlighted Nash equilibria are not designated as equilibrium points because they do not satisfy the condition of stability. It means that for example if G_2 changes its strategy to $mup_{G_2} = 10\$$ (with equal expected profit $9000\$$) than at this point G_1 has an incentive to change its strategy to either $mup_{G_1} = 0\$$ or $mup_{G_1} = 10\$$ to increase its prospective payoff.

Agents use subgame perfect Nash equilibria to select actions

We consider here that the agents know the different Nash equilibria and use them to select their actions according to the procedure outlined in section 2.4.

By repeating the matrix game, we observe that agents G_1 and G_2 are switching between $mup = 0\$$ and $mup = 30\$$ whereas G_3 permanently adheres to his dominant strategy ($mup_{G_3} = 20\$$) (see Fig.6). There are two stable Nash equilibria in the system and agents unilaterally assess what strategy to play in order to get into one of these equilibria. Due to the lack of coordination between these agents, different situation may occur. Either they play the $(0, 30, 20)$ equilibrium, the $(30, 0, 20)$ equilibrium or no equilibrium at all (in which case either $(30, 30, 20)$ or $(0, 0, 20)$ is played).

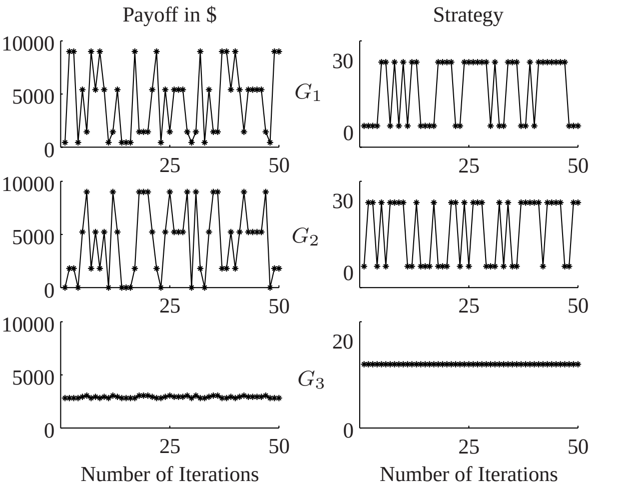

Figure 6: Evolution of the payoffs and of the actions when subgame perfect Nash equilibria are used to model the agents' strategies.

Agent-Based Model

In the two active agent case we found that for one pure Nash equilibrium the Q-functions indeed converged to this equilibrium. We will now assess the development of the Q-functions with all generators modelled as active agents. For G_3 we observe that the evolution of the Q-function is similar to the evolution displayed in figure 4.[9] G_3 learns that it has market power and that it should choose a markup of 20$ to maximize its reward. This development is in accordance with the results obtained by Nash equilibria analysis. At both equilibrium points the greedy action for G_3 is to choose the highest markup. However, the development of the Q-functions for G_1 and G_2 differs significantly. If when only G_1 and G_3 were active agents, we observed (see figures 4 and 5) that the Q-function learned by G_1 was clearly indicating that a markup of 30$ was the greedy action, it is no longer the case here. In the present case the greedy action always changes. Furthermore, the evolution of the Q-function seems now to respond to a cyclic process. Figures 7 (G_1) and 8 (G_2) show the evolution of the Q-functions. We see, that when a markup of 30$ is the greedy action for G_1, G_2 chooses a markup of 0$ and vice versa. These two combinations of markups indeed correspond to the single stage Nash equilibria (see table 4). We will now assess why the cycling occurs. It is helpful to keep in mind, that actions of one generator influence not only its own reward but also the reward of the others and that the randomness (introduced by the ϵ-parameter) plays an important role. For argumentation we use table 4, figure 7 (displaying time instants t_1 to t_3) and figure 9 (displaying time instants t_3 to t_5). Let us assume that after an arbitrary number of market clearings we are at time instant t_1, with $mup_{G_1} = 0\$$ and $mup_{G_2} = 30\$$ being the greedy actions (determining the first Nash Equilibrium point), where for G_1 the expected payoffs of the non-greedy actions are all below 4000 $. We now move on to time instant t_2, where G_1 and G_2 still keep their greedy actions of $mup_{G_1} = 0\$$ and $mup_{G_2} = 30\$$, but in case of G_1 the expected rewards for $mup_{G_1} = 10\$$ and $mup_{G_2} = 20\$$ develop close to the reward of the greedy action.[10] Thus, we are facing a situation where due to random actions of G_2 the greedy action of G_1 might change.

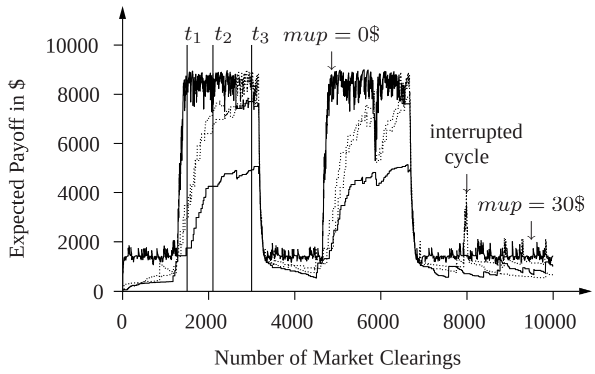

Figure 7: Evolution of the Q-function for G_1 (3 active agents)

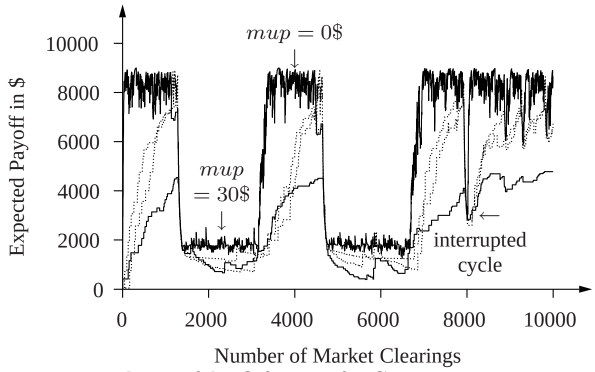

Figure 8: Evolution of the Q-function for G_2 (3 active agents)

This indeed happens at time instant t_3. Due to a random bid of G_2, choosing a markup of either 0$, 10$ or 20$, the expected reward of $mup_{G_1} = 0\$$ for G_1 falls below the expected reward of $mup_{G_1} = 10\$$. Thus, $mup_{G_1} = 10\$$ becomes the greedy action of G_1. In table 4 we see that given a markup of $mup_{G_1} = 10\$$, G_2 can do better by choosing a markup $mup_{G_2} = 0\$$. This behavior is indeed learned (time instant t_4). The same consideration applies to G_1. With $mup_{G_2} = 0\$$ G_1 can do better by bidding at $mup_{G_1} = 30\$$ (time instant t_5). Eventually, we reach the second Nash equilibrium.[11] Figure 9 provides a sample of the cyclic variation of the greedy actions for G_1 and G_2. For the other half of the cycle a similar line of argument applies. As the mechanism follows the considerations above, we do not deliver a detailed explanation. The path is displayed as dotted line in figure 9.

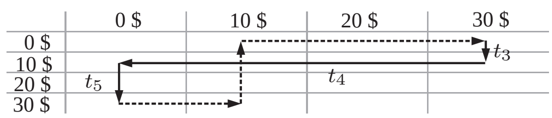

Figure 9: Variation of Greedy Strategies with Time for G_1 and G_2

Note, that the paths might deviate slightly as for a number of bid-tuples we face identical rewards. Thus, the

[9] Because of the similarity we do not provide an additional figure.

[10] From table 4 it can be read that the rewards for $mup = 0\$$, $mup = 10\$$ and $mup = 20\$$ are all equal to 9000$, assuming G_2 is bidding at 30 $.

[11] The transits at time instants t_4 and t_5 are occurring very fast. Thus, they can not be observed in the displayed Q-functions.

generators are indifferent between those bids and the action is determined by random influence. Nevertheless, this does not change the overall cycling mechanism. Furthermore, due to the randomness, cycles may not be fully completed and the generators may instead revert at any state back to the previous equilibrium point (see figures 7 and 8).

4.4 Parameter Dependency of Agent-Based Approach

In our previous analysis we kept the experimentation parameter ϵ and the learning rate α constant - both at values of 0.1. However, one may argue that a different choice of parameters will influence the model outcome. Hence, we carried out simulations with different discrete sets of parameters. For α and ϵ being smaller than 0.1, we observe less frequently oscillatory behaviors, and, when observed, the periods of oscillation seem to be larger as the generators act less randomly and the learning is slower. The frequency of the cycles tends to increase with α and ϵ but, with too large values for these parameters, the oscillatory behavior disappears and the evolution of the Q-functions seems to be driven by a totally random process. To explain this, let us first take ϵ large. In that case no learning takes places, as all actions are totally selected at random. A learning rate of 1 has a similar influence. As only the last reward received determines the value of the Q-function (the expected reward) learning can not evolve over time. Hence, we face an almost arbitrary development of the Q-functions.

Nevertheless, a cyclic or oscillatory model behavior occurred for almost every combination of α and ϵ in the three active agent case (two Nash equilibria). For one Nash equilibrium (two active agent case) we found that with smaller values of α and ϵ the learning is slower but the equilibrium is still approached, whereas for values close to 1 the Q-functions may not evolve to the equilibrium point and seem to develop in an almost arbitrary way as described above.

5 CONCLUSIONS

To compare Nash equilibria analysis and agent-based modelling we defined a pool market as a repeatedly played matrix game. Generators may act strategically, i.e. by bidding above their marginal production cost. To assess this behavior we employed a Q-learning algorithm as a behavioral agent model and carried out simulations on a benchmark power system. We analytically computed the Nash equilibria of the system and then compared the results with those obtained by the agent-based approach. We showed that in case of one Nash equilibrium there is high likelihood for the Q-learning algorithm to indeed converge to this equilibrium, whereas in case of two Nash equilibria we observe a cyclic behaviors. We have checked that these phenomena are robust with respect to different parameters. Therefore, we conclude that in the presence of multiple equilibria cyclic phenomena are likely to occur.

ACKNOWLEDGEMENTS

The authors would like to thank the Projekt- und Studienfond der Elektrizitätswirtschaft (PSEL), Chambre romande d'énergie électrique (RDP-CREE), Schweizerische Betriebsdirektorenkoferenz (SBDK) and Bernische Kraftwerke AG (BKW).

REFERENCES

[1] B. Hobbs, C. Metzler, and J.-S. Pang, "Strategic gaming analysis for electric power systems: an MPEC approach", *Power Systems, IEEE Transactions on*, vol. 15, no. 2, pp. 638–645, 2000.

[2] D. W. Bunn and F. S. 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, 2001.

[3] T. Krause, G. Andersson, D. Ernst, Elena V. Beck, R. Cherkaoui and A. Germond, "Nash Equilibria and Reinforcement Learning for Active Decision Maker Modelling in Power Markets", *6th IAEE Conference - Modelling in Energy Economics*, Zurich, September 2004.

[4] D. Fudenberg and J. Tirole, "Game Theory", Cambrigde: The MIT Press 1991.

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

[6] A. Minoia, D. Ernst, M. Dicorato, M. Trovato and M. Ilic, "Reference transmission network: a game theory approach - part I: model", Submitted to IEEE Transactions on Power Systems.

[7] S. de la Torre, J. Contreras and A.J. Conejo, "Finding multiperiod Nash equilibria in pool-based electricity markets" in *IEEE Transactions on Power Systems*, Vol.19(1), pp.643-651, February 2004.

[8] A. Haurie and J. Krawczyk, "An Introduction to Dynamic Games", Course notes. November 2001, Available Online: http://ecolu-info.unige.ch/ haurie/fame/.

[9] R. Porter, E. Nudelman and Y. Shoham, "Simple Search Methods for Finding a Nash Equilibrium" in *Proceedings of the 19th National Conference on Artificial Intellegence*, San Jose, CA, 2004.

[10] L. P. Kaelbling, M. L. Littman and A. W. Moore, "Reinforcement learning: a survey", *Journal of Artificial Intelligence Research*, vol. 4, pp. 237-285, 1996.

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

[12] M. Littman, "Markov games as a framework for multiagent reinforcement learning", in *Proceedings of the Eleventh International Conference on Machine Learning*. pp.157-163, San Francisco, CA: Morgan Kaufmann, 1994.

[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.