Automated bidding strategy adaption using learning agents in many-to-many e-markets

Daniel Veit, C. Czernohous

Angaben zur Veröffentlichung / Publication details:

Veit, Daniel, and C. Czernohous. 2003. "Automated bidding strategy adaption using learning agents in many-to-many e-markets." In Poster Proceedings of the Workshop on Agent Mediated Electronic Commerce V (AMEC-V), held at the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Melbourne, Australia, 1–5. Augsburg: Universität Augsburg.

Automated Bidding Strategy Adaption using Learning Agents in Many-to-Many e-Markets

Daniel Veit and Clemens Czernohous
Chair for Information
Management and Systems
University of Karlsruhe
Englerstrasse 14, D-76131 Karlsruhe, Germany
{veit, czernohous}@iw.uka.de
http://www.iw.uni-karlsruhe.de

ABSTRACT
In this paper the issue of bidding strategy learning in electronic markets is addressed. The primary aim is to identify machine learning techniques which are best suited to learn bidding behaviour in electronic markets. The developed methodologies are applied within a structured market engineering process to improve the quality of market designs. Market simulations are carried out based on a discriminatory price double auction market design.
The simulations of the market behavior are modelled in a multi-agent system. The market participants which are (i) market maker agents (ii) supplier agents (iii) consumer agents act as autonomous agents on a simulated market. Supplier agents are static, i.e. not equipped with learning techniques. Consumer agents are modelled using different machine learning methods for price determination. We develop the market simulation library MELBOURNE (Market Engineering Library for Bidding Objectives Using Realtime Negotiation Environments). In the simulation runs using the library, we show that learning algorithms always outperform heuristic methods in price discovery.

Keywords
Market engineering, bidding and negotiation strategies, market-based problem solving, simulation and evaluation of properties of novel and complex mechanisms

1. INTRODUCTION
Beyond the initial hype, electronic markets and electronic negotiations are now gaining more and more importance within concrete industry applications and therefore also target driven research (see also [6]). Hence, the focus in e-market research has changed towards the application of structured market engineering processes before introducing concrete markets as business cases[1]. These steps are mainly applied to enhance the chances for their success in every day practise. Within e-markets, automated negotiation strategies and mechanisms are increasingly integrated into autonomously acting, self-oriented agents.

In many concrete application domains such as financial, electricity, bandwidth, radio frequency or in future also software certification rights trading, bidding mechanisms and auction protocols are applied to solve allocation problems using market mechanisms. On this background the tools which can be provided by a structured market engineering process are of great benefit. The market engineering process covers

(i) a transaction product design section and within this the product and/or service specification

(ii) a transaction process design section and

(iii) an evaluation section and within this efficiency, functionality, performance and acceptance tests.

Besides the identification of the participants, the design of the market structure (see also e.g. [11]) as well as the theoretical definition of the concrete market in sections (i) to (ii), one of the main steps within this process is the simulation and experimental evaluation of a concrete market within a domain using autonomously acting, self-oriented agents.

The work described in this paper is based on the idea of simulating a concrete market combining tools from the multi-agent domain with several machine learning approaches. The framework which we designed for this purpose is called MELBOURNE (Market Engineering Library for Bidding Objectives Using Realtime Negotiation Environments).

The general setup for the market simulation has the following form: A central market maker agent runs continuous double auctions in which both, the supplier agents and the consumer agents place their bids. In each round the bidders obtain as a result information about the auction. Using that information they adapt their bidding strategy using the learning algorithm incorporated into the agent.

[1]See also the EU-FP6 Integrated Project initiative "ETrading Europe" (http://www.etrading-europe.org, 04/24/2003) in which the authors are participating.

AAMAS 2003, Second International Joint Conference on Autonomous Agents and Multi Agent Systems - Workshop on Agent Mediated Electronic Commerce, Melbourne, Australia.

MELBOURNE is a tool to simulate market performance and outcome on different levels. It is developed to simulate market behavior starting from a basis of real world data from past market results. The decisions and bidding strategies are derived from these past data and used together with the experiences collected to learn and adapt bidding strategies for future behavior in markets.

This paper is structured as follows. In Section 2 the market environment for MELBOURNE is introduced. Here, market participants, market mechanisms and the auction framework is described. In Section 3 the machine learning techniques applied in market simulation are described. In Section 4 the preliminary results from competitive bidding in a concrete market are provided. Finally in Section 5 the key issues of this paper are provided and an outlook on future research is given.

2. MARKET ENVIRONMENT FOR MELBOURNE

In this section the market mechanism, the market participants as well as the market implementation and simulation setting are introduced.

2.1 Market Mechanism

Markets are not only based on allocation theory but also empirically known as efficient and transparent instruments for coordination (see [9]). The problem of coordination within a market is solved by an allocation mechanism which assigns offers to requests in order to clear the market.

In the MELBOURNE library discriminatory price double auctions are chosen to be the central allocation mechanism. This kind of auction offers an efficient allocation with only little communication overhead. and provides a good overview even in market models where the participants are represented partially in an aggregated way.

The discriminatory price double auction (see also [7]) has the following properties. It generates individual prices based on bid/ask pairs. An auction round starts with both supplier and consumer agents submitting a tuple of an amount and a price.

2.2 Market Participants

The multi-agent simulation environment MELBOURNE consists of three different roles of agents:

(i) The **market maker agent** clears the market as the central coordinator between supplier and consumer agents.

(ii) The **supplier agent** provides goods into the market for consumer agents.

(iii) The **consumer agent** is looking for goods which are provided in the market. This agent submits bids on offers of supplier agents.

These roles are modelled according to the properties of autonomous agent design in multi-agent systems (see [12]). For the technical implementation the JADE multi-agent environment is used (see [1], JADE version 2.61).

2.3 Simulation Setting

The simulations carried out on top of the market environment provided in the last two sections are designed as follows. The central idea is to evaluate market performance and the quality of machine learning methods for competitive bidding. Therefore, supplier agents are equipped with machine learning algorithms.

The discriminatory price double auctions are carried out either single, 100 or 1000 after each other. To prove the convergence of the learning techniques, firstly a liquid market is provided. After that, the same auction setting is set up and an illiquid market is set up in which several consumer agents are concurring to obtain their needed capacities. It is now shown which learning technique adapts best to the market mechanism in order to obtain the highest amount of goods for the least amount of payment.

In this paper an artificial electricity market is taken as case study. Hence, the traded good is electricity capacity. Supplier agents are representants of electricity suppliers and consumer agents are private as well as public electricity consumers.

3. APPLIED MACHINE LEARNING TECHNIQUES

Having described the MAS environment, this section deals with the main question: *"Which Machine Learning Technique is appropriate for the present market simulation problem?"* This includes the subproblems in designing a learning system: *(i) What* and *(ii) How* to learn?

In the described simulation environment, we want to use agents to represent market participants in an electricity market. Agent representing the customers should learn to adapt their bidding behavior in a competitive market environment to satisfy their demand for electricity in each period. There exist various learning mechanisms, consequently, we give a brief introduction to learning techniques and present two variants of learning algorithms for the present market problem. These learning algorithms are applied and evaluated in Section 4.

3.1 Agent-based learning

The learning process can be performed either within one single agent or in cooperation of two or more agents. [10] denominate the first *centralized (isolated)* and the latter *decentralized (interactive)* learning. In a competitive market environment agents naturally learn isolated and use the learned knowledge for their own advantage. Therefore, we focus on centralized learning techniques.

Besides supervised learning, where a teacher assess the performed action, learning algorithms can be classified in unsupervised learning (without feedback) and reinforcement learning. Psychology has analyzed human learning behavior. It was observed, that the probability increases for choosing a particular action again in the future, if the feedback was positive and accordingly decreases, if a negative feedback was received. This effect is called *reinforcement* and was taken advantage of in machine learning in environments, where it is impossible for the agents to compare the action's result with a specified goal. Instead, the agent receives feedback for a performed action and deduce the coherence of action and its performance. Generally, a given feedback is assigned not only to one action, but to the action of other agents or earlier performed actions. Q-learning is one type of reinforcement learning to handle the temporal credit assignment problem. [5] have studied human behavior and developed a reinforcement algorithm to represent human bidding behavior.

The market environment has different requirements on a learning algorithm:

(i) learning without previous knowledge or cognition of a model of other agents,

(ii) reaction on interaction partners behavior,

(iii) the own behavior is assessed indirectly,

(iv) the implication of an action might not be measurable or is observed with time delay

Consequently, these requirements fit best with a reinforcement learning algorithm. [2] and [8] have successfully used reinforcement-learning algorithms for the simulation of electricity markets (see also [3] and [4]).

3.2 Reinforcement Learning

The agents observe the environment and perceive an environmental state $s \in S$ (set of environmental states S), and decides on an action $a \in A$ (set of possible actions A). Action a causes a change of the environmental state to s' and effects the reinforcement signal $r = R(s, a)$ as reward/punishment. The reinforcement function is of type $R : S \times A \rightarrow \mathbb{R}$. The function $P : S \times A \times S \rightarrow [0, 1]$ provides the probability for the state change from $s \rightarrow s'$ by performing action a.

Strategy π maps the states $s \in S$ on actions $a \in A$ and maximizes the long-term reinforcement signal. The value function $V^*(s) = \max_x (R(s,a) + \gamma \sum_{s' \in S} P(s,a,s')\, V^*(s'))$ determines the value of state s as the sum of direct reward r and discounted value of the next state s' by choosing best action a.

To apply the described reinforcement learning model the probability function and the reward function have to be known. This is not the case for the present scenario of a market simulation. Hence, we use Q-learning, a specific reinforcement learning method.

3.3 Q-Learning

Q-Learning has no need for a model of other agents or of the environment. The learning function $Q^*(s, a)$ determines the expected reward for action a in state s. The reinforcement value function is $V^* = \max_a Q^*(s, a)$. Consequently, $Q^*(s, a)$ can be defined recursively:
$Q^*(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(sa, s') \max_{a'} Q^*(s', a')$. In the current state, the agent chooses the action with the maximum Q-value. Assuming that the sample $r + \gamma \max_{a'} Q(s', a')$ is correct with a higher probability, because it contains the exact reward value, the value $Q(s, a)$ is modified towards $r + \gamma \max_{a'} Q(s', a')$:

$$Q(s,a) := Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s,a))$$

For an infinite repetition of each state transition and a slow decrease of the learning rate α Q converges to Q^* with probability 1.

In simple contexts a table-based representation of the Q-function is possible, whereas complex problems require generalization, which influences the convergence characteristics. Consequently, it appears useful to test different mechanisms for the Q-function and analyze its performance and influence on the system's behavior. In the following section we compare two different mechanisms.

4. RESULTS

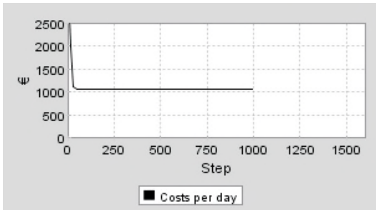
In the previous section we have outlined the idea of reinforcement learning and Q-learning as a specific reinforcement learning method. In this section we present the results of a comparison of two different Q-functions implemented for consumer-agents acting in an electricity market.

The consumer agents compete for electricity to satisfy their needs. Electricity is not provided in sufficient amounts. Therefore, consumer agents have to learn a competitive strategy to buy the demanded amount of electricity. We have implemented a method realizing a Q-function $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ in an n-dimensional table. The second Q-function was implemented as neuronal feed-forward network with one hidden layer and a definable amount of neurons.

Four different cases have been analyzed to compare the characteristics of the learning agents and their performance in different market environments.

4.1 Case 1: Isolated Homogeneous Market Simulation

Two supplier and one consumer agents act in the market. The simulation was conducted three times to test the single performance



Rule based agent

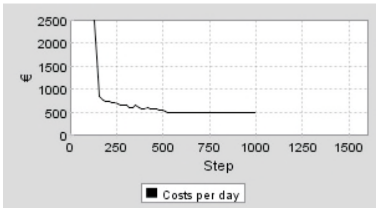


Table based agent

Neuronal network agent

Figure 1: One single consumer agent with different learning algorithms

of one type of consumer agent. Figure 1 shows the results of a rule based agent, table based agent and a neuronal network agent in such a situation.

The rule based agent augments the bidding price to get the demanded amount of electricity, but reaches the maximum bidding price after a short period, which results in penalty-costs. The table based agent learns a strategy to bid always the minimum price, which results in higher costs during the learning phase and lower overall costs. Also, the neuronal network agent learns a bidding strategy with lower prices. The higher costs in comparison to the table based agents are result of particular network-factors. The mapping of input- and output-vectors within a neuronal network agent is subject to statistical characteristics. If one agent searches in a small part of the state space only, he steadily forgets knowledge collected in an other part.

In the current situation the table based agents performs best.

4.2 Case 2: Competing Algorithms

There is also a shortage of electricity supply in the second scenario. Now, a table based agent and a neuronal network agent compete to satisfy their demand. Figure 2 shows the cost evolution of both

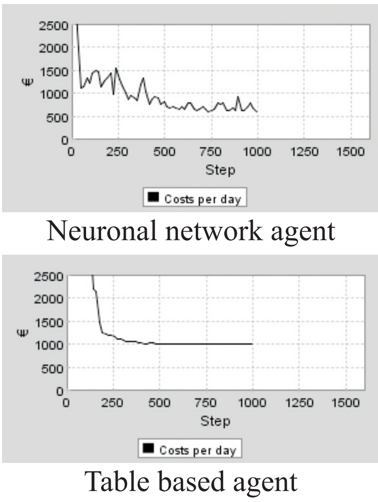


Figure 2: Neuronal network agent vs. table based agent

agents. The table based agent cannot find a strategy to outbid the neuronal network agent, because the latter explores the action space faster. Consequently, the neuronal network agent finds an overall strategy earlier. This is not the case in competition with an other table or rule based agent.

4.3 Case 3: Learning In A Dynamic Market Environment

The learning strategy and the length of the learning period is important for the learning behavior. A table based agent searches the action space systematically. The more often an action was chosen, the more exact the weighting of the action will get. The agent has to evaluate each action of the action space to improve the values. In a dynamic environment a table based agent has to explore the action space continuously. Figure 3 shows the cost evolution of

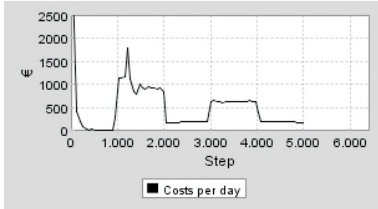


Table based agent

Figure 3: Table based learning agent in a dynamic environment

a table based agent in a dynamic environment. For the first 1000 rounds only, the table based agent acts in the market. Then, two rule based agents enter the market for the next 1000 simulation rounds to increase competition and leave the market afterwards. An other 1000 rounds later two rule based agents enter the market again. The cost-function shows, that the strategy with and without competition converge to an overall strategy.

4.4 Case 4: Multi-Algorithms in a Competitive Market

The fourth scenario shows two table based, one neuronal network based and one rule based agent in a market with electricity shortage. Each agent develops a stable strategy, but with different total cost. The neuronal network agents shows fluctuations due to its learning mechanism. Figure 4 illustrates the total costs of the agents. Both

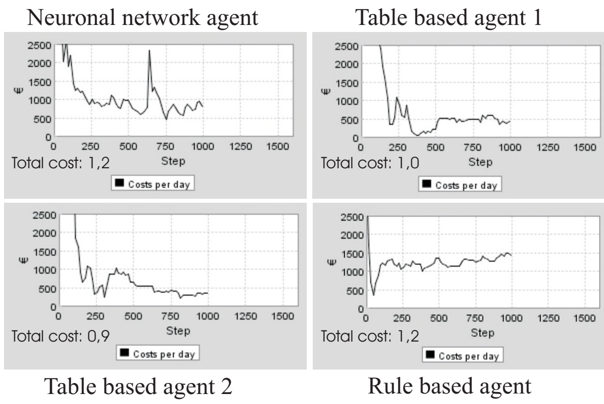


Figure 4: Rule based agent, table based agent and neuronal network agent in one market

types of learning agents achieve better results than the rule based agent.

Highly visible is the diverse success development during the learning phase and the different performance of the tested algorithms. As was shown in the presented cases, learning algorithms perform better in some situation than in others. To use learning algorithms for representation of human behavior, it is inalienable to previously test its performance in different situation. Otherwise simulation results are questionable. Before applying learning algorithms in an agent-based simulation, learning algorithms also need a training phase, in which the different strategies are rated.

5. CONCLUSIONS & OUTLOOK

In this paper we shortly introduce the market engineering process as a structured way of designing electronic markets. As one component of this process multi-agent based simulation for market evaluation is described. The central aspect of this paper is to focus different machine learning algorithms in order to identify algorithms for future market mechanism evaluation.

The basis for market simulation is a discriminatory price double auction. Applying this market design, a static supplier agent is chosen in an illiquid market. Several consumer agents equipped with different machine learning algorithms are instantiated. We (i) show that in liquid markets convergence in supply prices is reached. As soon as markets become illiquid and several consumer agents are concurring to obtain their needs we show that (ii) consumer agents which are equipped with a neural network outperform simple table based learning agents. (iii) we show that in a highly competitive scenario with several types of agents simple table based learning consumer agents outperform rule based agents but also outperform complex neural network agents. In this scenario the neural networks have been untrained.

More elaborated evaluations of price/amount tuples and longer auction periods will be performed. Additionally supervised training on the neural network consumer agents will be performed. Then the aim is to show that trained neural network consumer agents outperform all other types of consumer agents. After stability in learning results is proven, the next research issue is to equip also the supplier agents with learning algorithms. Finally, in a fully learning environment, we will then modify the market mechanisms. Therefore different auction protocols will be run comparatively. MELBOURNE is aimed to establish as an evaluation environment for market mechanisms in electronic negotiations. To enable the domain independent objective MELBOURNE will be applied and improved throughout six prominent European market scenarios within the ETrading Europe initiative (see Section 1).

Acknowledgements

For his great engagement and help developing the JADE-based e-market as well as evaluating the learning algorithms we would like to thank Werner Hunger. Also we would like to thank Dominik Möst from University of Karlsruhe, IIP for his support concerning the application domain of energy markets.

6. REFERENCES

[1] F. Bellifemine, A. Poggi, and G. Rimassa. Developing multi agent systems with a fipa-compliant agent framework. *Software - Practice And Experience*, 31:103–128, 2001.

[2] J. Bower and D. Bunn. A model-based comparison of pool and bilateral market mechanisms for electricity trading. *Energy Journal*, 21(3):1–29, July 2000.

[3] J. Bower, D. W. Bunn, and C. Wattendrup. A model-based analysis of strategic consolidation in the german electricity industry. *Energy Policy*, 29:987–1005, 2001.

[4] D. W. Bunn and F. S. Oliveira. Agent-based simulation: An application to the new electricity trading arrangements of england and wales. Technical report, IEEE - TEC, special issue: Agent-based Computational Economics, 2002. http://www.geocities.com/fsic12/Images/pdf/neta.pdf.

[5] I. Erev and A. Roth. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *The American Economic Review*, 88(4):848–881, September 1998.

[6] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

[7] J. Nicolaisen, P. Petrov, and L. Tesfatsion. Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transations on Evolutionalry Computation*, 5, 2001.

[8] J. Nicolaisen, V. Petrov, and L. Tesfatsion. Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transactions on Evolutionary Computation*, 5(5):504–523, October 2001. www.econ.iastate.edu/tesfatsi/mpeieee.pdf.

[9] B. Schmid. Elektronische Märkte. *Wirtschaftsinformatik*, 35(2):465–480, 1993. in German.

[10] S. Sen and G. Weiss. Learning in multiagent systems. In G. Weiss, editor, *Multiagent Systems*, pages 259–298, Cambridge (Massachusetts), London (England), 1999. The MIT Press.

[11] D. Veit, J. P. Müller, and C. Weinhardt. Multidimensional matchmaking for electronic markets. *Applied Artificial Intelligence*, 16(9-10):833–869, 2002.

[12] G. Weiss. *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.