

MASCEM: Electricity Markets Simulation with Strategic Agents

Zita Vale, Tiago Pinto, Isabel Praça, and Hugo Morais

Electricity markets are complex environments, involving numerous entities trying to obtain the best advantages and profits while limited by power-network characteristics and constraints.¹ The restructuring and consequent deregulation of electricity markets introduced a new economic dimension

MASCEM uses reinforcement learning algorithms to provide players with strategic capabilities in electricity markets, helping them react to the dynamic environment and adapt their bids accordingly.

to the power industry. Some observers have criticized the restructuring process, however, because it has failed to improve market efficiency and has complicated the assurance of reliability and fairness of operations.

To study and understand this type of market, we developed the Multiagent Simulator of Competitive Electricity Markets (MASCEM) platform based on multiagent simulation. The MASCEM multiagent model includes players with strategies for bid definition, acting in forward, day-ahead, and balancing markets and considering both simple and complex bids. Our goal with MASCEM was to simulate as many market models and player types as possible. This approach makes MASCEM both a short- and medium-term simulation as well as a tool to support long-term decisions, such as those taken by regulators.

This article proposes a new methodology integrated in MASCEM for bid definition in electricity markets. This methodology uses reinforcement learning algorithms to let players perceive changes in the environment, thus helping them react to the dynamic environment and adapt their bids accordingly.

Electricity Markets

The electricity market environment typically consists of a *pool* that players submit their bids to, which can be symmetric or asymmetric, and a floor for bilateral contracts. Additionally, some countries also include a balancing market, in which each market player must decide whether to, and how to, participate in each market type.

Besides the electricity sellers and buyers that negotiate in the market, these markets usually also include a market operator and a system operator. The *market operator* is responsible for regulating the market; it manages the pool using a market-clearing tool to set market price and a set of accepted selling and buying bids for every negotiation period. The *system operator* is usually responsible for managing the transmission grid and all the involved technical constraints. Every established contract, either through bilateral contracts or the pool, must be communicated to the system operator, who analyzes their technical feasibility from the point of view of the power system.

Players negotiating on the pool must prepare a bid for the 24 periods of the spot market.

These bids are subjected to complex conditions, including both technical and economic constraints, such as

- the load gradient that refers to the ramping up and down of plants;
- indivisibility, wherein a generation facility only agrees to be dispatched if its offer is completely accepted; and
- minimum income, wherein a block bid will not be accepted by the matching algorithm if the minimum income requested by the participant is not fulfilled.

The market operator must assure that the economical dispatch accounts for the specified conditions, which might imply removing entities that have presented competitive bids but whose complex conditions were not satisfied. Complex conditions instigate the development of new kinds of player strategies for bid definition.

The balancing market's goal is to take care of the necessary adjustments on the viable daily program and the last final hourly program, correcting possible deviations from forecasted production or consumption. It is, therefore, a complementary platform to the day-ahead market.

Although only sellers can present complex conditions to the spot market, in the balancing market, both sellers and buyers may present complex conditions.

Another important issue is that sellers may become buyers and buyers may become sellers on the balancing market. That is also a new subject to be explored by market players when defining strategies for bid definition.

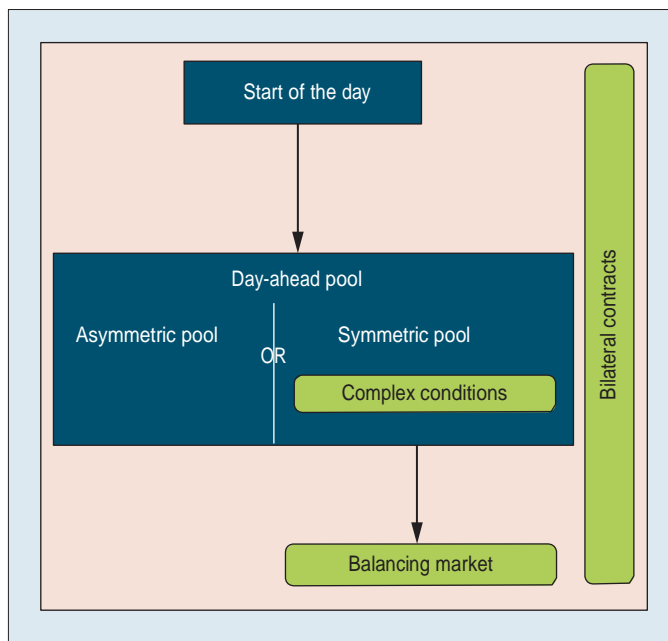


Figure 1. Negotiation sequence for one day in electricity markets. Market negotiations on the pool and through bilateral contracts can evolve simultaneously. The pool mechanism might be symmetrical or asymmetrical and include complex conditions. Considering these complex conditions is essential for the balancing market.

Figure 1 presents the common negotiation sequence for one day in electricity markets.

The need for understanding these mechanisms and how the involved players' interactions affects the outcomes of the markets has contributed to the increased use of simulation tools in order to determine the best possible results in each market context for each participating entity. Multiagent-based software is particularly well fitted to analyzing dynamic and adaptive systems with complex interactions among their constituents. Several of such modeling tools—designed to help researchers study restructured wholesale power markets—have emerged. In addition to MASCEM,² other relevant tools in this domain are AMES,³ EMCAS,⁴ and MASI.⁵

We implement players in MASCEM as independent agents, with their own ability to perceive the states and changes in the world and to act accordingly. These agents are provided

with bidding strategies, which must be adequate and refined to let them gain the highest possible advantage from each market context.

MASCEM Overview

MASCEM includes a complex simulation infrastructure that can cope with the diverse time scales of the supported negotiation mechanisms and with several agents competing and cooperating with each other. Figure 2 illustrates MASCEM's most important features.²

Unlike traditional tools, MASCEM does not postulate a single decision

maker with a single objective for the entire system⁶. Rather, we allow agents representing the different independent entities in electricity markets to establish their own objectives and decision rules. Moreover, as the simulation progresses, agents can adapt their strategies based on the success or failure of previous efforts. In each situation, agents dynamically adapt their strategies according to the present context and using the dynamically updated detained knowledge.²

MASCEM's key players reflect actual entities from real markets and provide a means for aggregating consumers and producers. Presently, we have agents representing market-independent entities such as the *system operator*, which is another simulator⁷ that gets the economical dispatch and undertakes power-flow analysis to assure economical agreements can be implemented without disturbing power-grid stability and technical constraints.

The *market operator agent* regulates pool negotiations. This agent analyzes bids presented to the pool and defines the market price and economical dispatch. It cooperates with the system operator by sending it the economical dispatch. The market operator agent uses different algorithms to account for complex conditions.

The *seller* and *buyer agents* are the two key players in the market. Sellers represent entities able to sell electricity in the market—for example, companies holding electricity production units. Buyers represent electricity consumers or even distribution companies. Sellers compete with each other because each seller is interested in maximizing its profits. They also cooperate with buyers while trying to establish a mutually profitable agreement.

Virtual power player (VPP) agents represent a set of producers, mainly based on distributed generation and renewable sources.^{8,9} They can provide the means to adequately support distributed generation increasing use and its participation in the context of competitive electricity markets. VPP agents are implemented as a coalition of agents, each one acting as an independent multiagent system.

VPPs are responsible for managing the coalition of producers, which includes negotiating in the electricity market on behalf of the coalition and negotiating internally with their members, to guarantee that the terms of each member's contract are fair and suited to the VPP's characteristics and objectives. For this process, we have developed a classification algorithm that analyzes each producer's characteristics and tests their suitability to the VPP's objectives.⁹ This provides the VPP with knowledge about which producers are most likely to favorably contribute to the

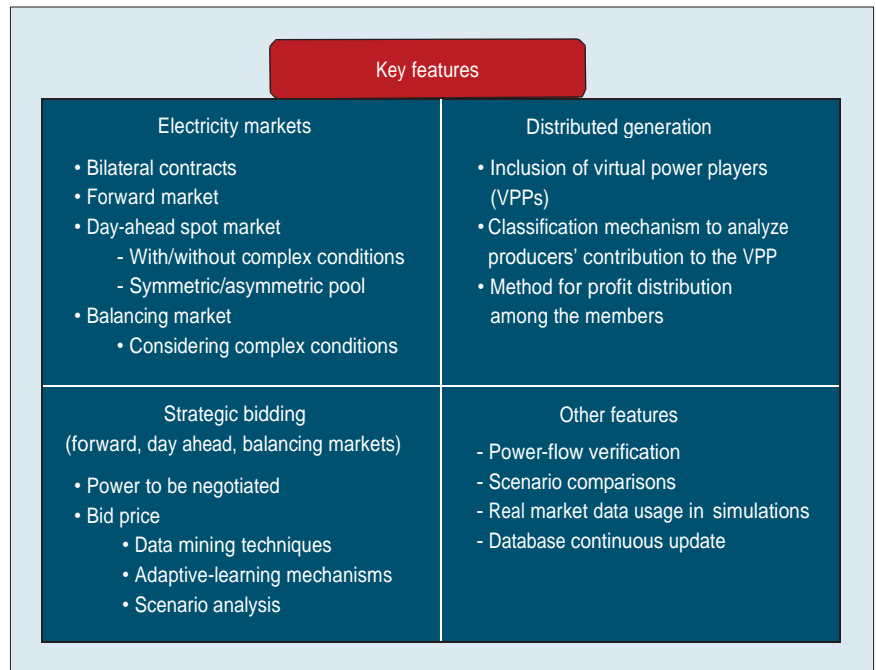


Figure 2. MASCEM key features. The figure summarizes MASCEM's most relevant characteristics such as the ability to simulate several types of negotiations found in electricity markets, consider algorithms for bids price definition and the inclusion of distributed generation. Also, important features such as power-flow analysis and scenarios definition based on real data are available.

VPP's results, which lets it decide which producers to aggregate.

MASCEM is implemented on top of the Open Agent Architecture (OAA, www.ai.sri.com/oa) using the AgentLib library, Java Virtual Machine 1.6.0, and LPA Win-Prolog Intelligence Server (www.lpa.co.uk).

All the agents share the OAA's Interagent Communication Language, no matter which machine they run on or which programming language they are programmed in, which lets us integrate the various software modules. This is a flexible framework, and its ability to develop agents in different programming languages is an advantage over other platforms. OAA is not a framework specifically devoted to develop simulations; we made some extensions to make it more suitable and to include, for example, time evolution.

Figure 3 displays a screenshot of a running buyer agent, showing its bids, its sold and unsold power, and the requests it is receiving. The top

part of Figure 3 shows the graphical representation of this agent's results in the pool. It includes the amount of energy that it bought in each period and compares its bid price to the market price.

The bottom shows the requests that this agent is receiving at each time and some information about its actions. In this case, period 20 has ended and the agent is getting ready to start negotiations for period 21. First, we can see that it received a notification from the market operator indicating that the time for the present period of negotiations ended. So, the agent performs the necessary arrangements to be ready for the next period. The second line gives a summary of this agent's results from the last period. The third line indicates the bid price and amount of power that this agent will negotiate in this market in the next period. The agent will use these output values in its bidding strategy in the next negotiation period.

Strategies for Bid Definition

The first issue to consider when defining strategies for players' action in electricity markets is to take advantage of the individual characteristics that each particular market offers. Using data mining mechanisms^{10,11} and machine-learning algorithms,^{12,13} we can predict the prices that are expected in each of these markets, given the required amounts of energy to be traded and the costs that must be covered.

Strategy for power Definition

Given the expected production of one player for each period of each day, the amount of power to be negotiated in each market is optimized to get the maximum profit that can be achieved. The inputs are

- the weekday, d ;
- the number of days, $Nday$;
- the negotiation period, p ;
- the number of periods, $Nper$;
- a Boolean variable for each distinct market or negotiation platform, $Asell_{M1}, \dots, NumM$, indicating if this player can enter it to sell;
- a Boolean variable for each session of the balancing market, $Abuy_{S1}, \dots, NumS$, indicating if this player is allowed to buy in each of them;
- the markets $M1, M2, \dots, NumM$; and
- the balancing market sessions $S1, S2, \dots, NumS$.

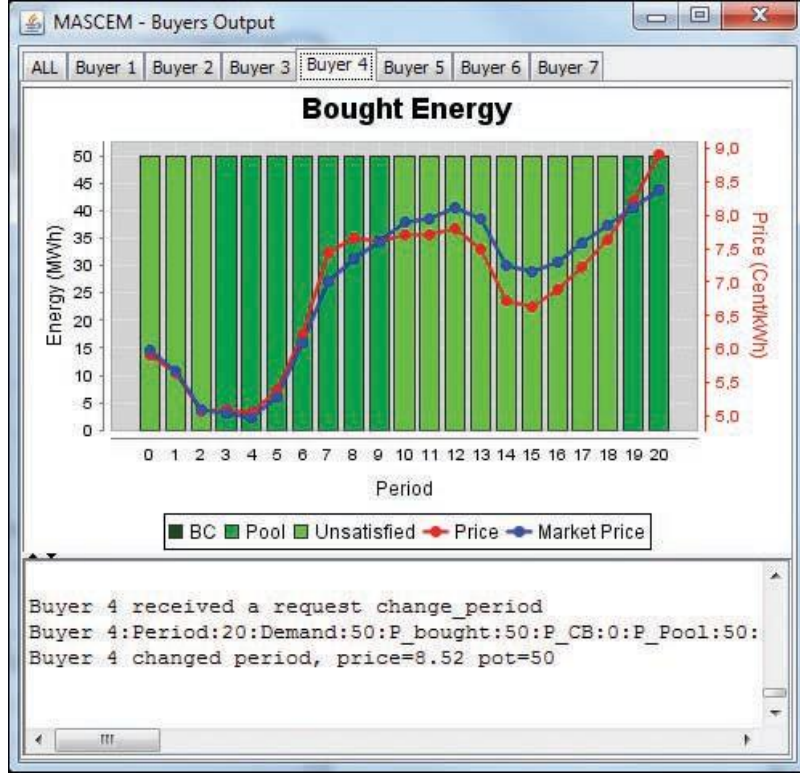


Figure 3. Buyer agent's output. The top shows the graphical representation of this agent's results in the pool. The bottom shows the requests that this agent is receiving and some information about its actions.

The outputs are $Spow_{M1}, \dots, NumM$, representing the amount of power to sell in each distinct market, and $Bpow_{S1}, \dots, NumS$, representing the amount of power to buy in each session of the balancing market. As Equation 1 shows,

$$f(Spow_{M1 \dots NumM}, Bpow_{S1 \dots NumS}) = \left. \begin{aligned} & \text{Max} \left[\sum_{M=M1}^{NumM} (Spow_{M,d,p} \times ps_{M,d,p} \times Asell_M) - \right. \\ & \left. \sum_{S=S1}^{NumS} (Bpow_S \times pb_{S,d,p} \times Abuy_S) \right] \\ & \forall d \in Nday, \forall p \in Nper, Asell_M \in \{0,1\}, \\ & Abuy \in \{0,1\} \\ & ps_{M,d,p} = \text{Value}(d, p, Spow_M, M) \\ & pb_{S,d,p} = \text{Value}(d, p, Bpow_S, S) \end{aligned} \right\} \quad (1)$$

where $ps_{M,d,p}$ is the expected selling price and $pb_{S,d,p}$ is the expected buying price.

The value function returns the expected value of the power for each

particular period of each day and for each market. That also depends on the amount of power available to trade. When a player tries to establish a bilateral contract, the deals are dependent on the amount of power that is being negotiated. This prediction takes that into consideration by applying fuzzy logic on the absolute amount of power to classify it in one of the categories defined by a clustering mechanism, which groups the ranges of amounts that present similar prices in each market.

We calculate the value function as follows: $Value(day, per, Pow, market) = Data(fuzzy(Pow), day, per, market)$. This equation has constraints dependent on the individual characteristics and requirements of each particular market. Therefore, these constraints are formulated based on the set of markets and their characteristics

considered for each run. We must also take into consideration the constraints imposed by each player's conditions. For every situation, we apply the main constraint in Equation 2:

$$\sum_{M=M1}^{NumM} Spow_M \leq TEP + \sum_{S=S1}^{NumS} Bpow_S \quad (2)$$

which ensures that the total power reserved to be sold in all markets is never higher than the total expected production (TEP) plus the power expected to be bought along all balancing market sessions.

The optimization mechanism lets agents

- negotiate with different players in the bilateral contracts, giving them the chance to get higher or lower prices, depending on the circumstances, and
- wait for the later balancing market sessions to provide higher amounts of energy if they expect the price to go up.

This mechanism also lets sellers buy and buyers sell in the balancing market so they can obtain better business opportunities. This is possible in two ways. They can use arbitrage opportunities, buying extra energy when the prices are expected to be lower and then selling it later when the prices go up. On the other hand, if the prices show the opposite tendency, they can offer more energy than the player actually expects to produce to increase profit and then buy that difference in the expected lower-price opportunities.

Strategy for price Definition

For each market, we predict prices using statistical methods, data mining techniques,^{10,11} neural networks,^{10,13} support vector machines (SVM), or several other methods.^{12,14,15} No method is best for every situation, but each might be the best for one or more particular cases.

To take advantage of the best characteristics of each technique, our method integrates several technologies and approaches. We place the set of algorithms below the main reinforcement learning algorithm, which means that in each moment and circumstance the technique that presents the best results for every scenario is chosen as the simulator's response. So, given as many answers to each

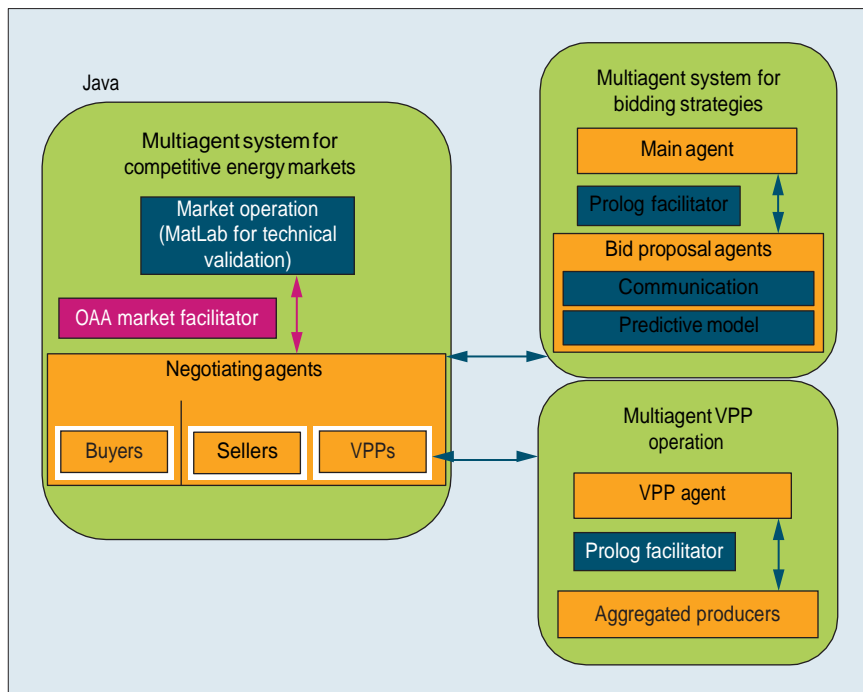


Figure 4. Multiagent system for bidding strategies integration with MASCEM. This figure presents the three multiagent systems that are integrated in this simulator: MASCEM main system and markets operation, virtual power player (VPP) agent modeling, and the multiagent system for bidding strategies. It also includes the programming languages and design tools used for the implementation.

problem as there are algorithms, the reinforcement learning algorithm will choose the one that is most likely to present the best answer given their past responses and each situation's present characteristics, such as the weekday, period, and market that the algorithms are being asked to forecast.

We implement this method as a multiagent system itself. There is one agent per distinct algorithm, with only the knowledge of how to perform it. This way the system can increase performance by executing all the algorithms in parallel; each agent gets its answer at the same time and sends it to the main agent, which then chooses the most appropriate answer. Figure 4 illustrates this multiagent system for bidding strategies' integration with MASCEM, along with the design tools we used to build each component.

The agents use several forecast algorithms during this process. The

statistical approaches use these elements:

- average market prices of the same weekday for the last month,
- average market prices of the last week considering only business days,
- average market prices of the last four months,
- regression on the market prices of the last four months, and
- regression on the market prices of the last five business days.

They also use a *feed-forward neural network* trained with the historic market prices. It has an input layer of eight units that includes the prices and powers of the same period of the previous day as well as the same weekdays of the previous three weeks. The intermediate hidden layer has four units and the output has one unit—the predicted market price for the period in question. Figure 5 shows the neural network topology.

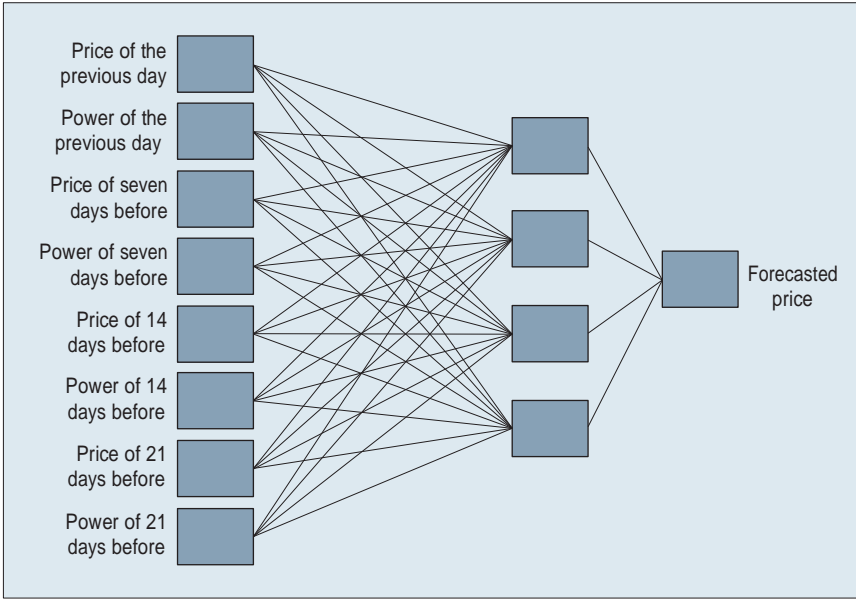


Figure 5. Neural network used to forecast the market price for one period. This neural network has an input layer of eight units, an intermediate layer of four units, and one output unit—the predicted market price.

An adaptation of the *AMES bidding strategy*³ uses the Roth-Erev reinforcement learning algorithm¹⁶ to choose the best among a set of possible bids that are calculated based on the cost/profit relation that the player presents when producing electricity. The various possible bids differ from each other because of the distinct combination of input parameters. The more combinations we set, the better chance we have of getting a good result. However, the number of combinations affects the processing time and the number of runs required for a satisfactory convergence.

The *composed-goal directed strategy* is based on two consecutive objectives: the first is increasing the profit (reducing the payoff), and the second is reducing the greenhouse effect emissions. This strategy will try to obtain the highest profit, decreasing the price if in the same period of the previous day the first objective was not completely satisfied, and then try to fulfill the second goal, while still satisfying the first.

The *adapted derivative-following strategy* is based on a derivative-following strategy proposed by Amy R. Greenwald and Jeffrey O. Kephart.¹⁷

The adapted derivative-following strategy adjusts its price by looking at the amount of revenue earned in the same period of the previous day as a result of that period's price change. If that period's price change produced more revenue per good than the same period of two days before, then the strategy makes a similar price change. If the previous change produced less revenue per good, then the strategy makes a different price change.

The *market-price-following strategy*, as the name suggests, follows the market price of the same period of the previous day. This simple strategy presents good results when prices show a tendency to stabilize for a certain period.

The main reinforcement algorithm presents a distinct set of statistics for each period, which means that an algorithm that is presenting good results for a certain period, with its output chosen more often when bidding for this period, might never be chosen as the answer for another pe-

riod because the various periods are completely independent. The tendencies observed when looking at the history of individual negotiation periods

show that they vary greatly from each other, which suggests that distinct algorithms can present distinct levels of results when dealing with such different tendencies.

The user can define the way the statistics are updated and, consequently, the best answer chosen. MASCEM provides three reinforcement learning algorithms. All the algorithms start with the same value of confidence, which is then updated according to their particular performance. The user can also attribute a weight value to each that defines their importance to the system. This means that a strategy that has a higher weight value will detach faster from the rest in case of either success or failure.

Our methods includes these three versions of reinforcement learning algorithms:

- a simple reinforcement learning algorithm, for which we update the values using a direct decrement of the confidence value C in the time t , according to the absolute value of the difference between the prediction P and the real value R : $C_{t+1} = C_t - |R - P|$;
- the revised Roth-Erev reinforcement learning algorithm, which includes all the previous algorithm's features and a weight value W that defines the importance of past experience: $C_{t+1} = C_t \times W - |R - P| \times (1 - W)$; and
- a learning algorithm based on the Bayes theorem of probability,^{18,19} for which we update the values by propagating the probability that each algorithm will be successful given its past performance:

$$EU(A|E) = \sum_i P(O_i | E, A) \times U(O_i | A)$$

where E is the available evidence, A is an action with possible outcomes O_i ,

$U(O_i | A)$ is the utility of each of the outcome states given that action A is taken, and $P(O_i | E, A)$ is the conditional probability distribution over the possible outcome states, given that evidence E is observed and action A taken.

Experimental Findings

To test our method, we generated three simulations for the same 14 consecutive days, starting from Wednesday, 29 October 2008. The data we used in this case study was based on real data from the Iberian market, extracted from OMEL (www.omel.es).

These simulations involve seven buyers and five sellers (three regular sellers and two VPPs). We created this group of agents to represent the real Spanish market, but we reduced it to a smaller version that contains the essential aspects of the market to allow a better individual analysis and study of the interactions and potential of each actors.

In the first simulation, we considered a different strategy for each agent. In the second and third simulations, all the strategies remained the same except for seller 2, which was our test subject. The first simulation used the neural network strategy for the bid definition. In the second, it used a statistical approach, a regression on the data of the last five business days. Finally, in the third simulation, it used our proposed strategy for bid definition. The selected reinforcement learning algorithm for this third simulation was the revised Roth-Erev, with equal value of the algorithm weights and a past-experience weight W value of 0.4. We used a small past-experience value to grant higher influence to the most recent results so that the algorithm could learn quickly and catch new tendencies in the changing market.

After the simulations, we compared the profits obtained by seller 2

using each strategy. Seller 2's power supply remained constant at 550 MW for each period throughout the simulations.

We defined the other players' bids as follows:

- Buyer 1 bought power independently of the market price. (The offer price was 18.30 c€/kWh, which was much higher than average market price.)
- Buyer 2's bid price varied between two fixed prices, depending on its need to buy. (The two variations were 10.00 and 8.00 c€/kWh.)
- Buyer 3's bid price was fixed at 4.90 c€/kWh.
- Buyer 4's bid considered the average prices of the last four Wednesdays.
- Buyer 5's bid considered the average prices of the last four months.
- Buyer 6 considered the average prices of the last week (considering only business days).
- Buyer 7 only bought power if market prices were lower than average market price.
- Seller 1 needed to sell all the power that it produced. (The offer price was 0.00 c€/kWh. The price at which it actually sold was the market price returned by the market operator after all bids were received.)
- Seller 3's bid considered the average prices of the last four months with an increment of 0.5 c€/kWh.
- VPP 1 included four wind farms and offers a fixed value all day. (The offer price was 3.50 c€/kWh.)
- VPP 2 included one photovoltaic, one cogeneration, and one mini-hydro plants. Its offer price was based on the costs of cogeneration and on the total forecasted production.

Because the reinforcement learning algorithm treated each period of

the day as a distinct case, we had to analyze each period individually. Figure 6 presents the evolution of seller 2's profits in the first period of the day, over a period of 14 days. Figure 6a presents the results from the first simulation, Figure 6b shows seller 2's profits in the second simulation, and Figure 6c gives seller 2's profits in the third simulation using our proposed method.

Figure 6 shows that the third simulation was clearly the most profitable for seller 2. In the first day using the proposed strategy, the profit was below the value of the profit using the regression. That is because the confidence values for all the algorithms were initially equal so the selection of the answer is made by chance. The selected algorithm answer originated a low profit.

After the reinforcement learning algorithm was updated on the second day, it chose the strategy that got the best reward the first day. At that point and on the third day, its profit was above both the other comparison strategies.

On the fourth day, its profit value was below the neural network results because the algorithm was still selecting the algorithm that got the best results in the first three days.

On the fifth day, the value was equal to the neural network. This trend continued until day 10, despite its lower values on days nine and 10. On day 11, the reinforcement learning algorithm no longer chose the neural network because its value was higher than those of the other two strategies.

In the last two days, the reinforcement learning algorithm selected the regression algorithm, catching its high value tendency in time to get two good final results. This was because of the low past-experience weight value; otherwise this algorithm

would have been chosen because of its weaker performance in the first days. (Further details about this case study are available at www.mascem.com.)

These results show that our proposed method achieves better results than the individual strategies. We were able to catch the good result trends by reacting quickly to the decreasing tendencies. This result demonstrates that we can combine several algorithms with distinct characteristics. Because we can combine all the algorithms and they work in parallel, this system in fact becomes more intelligent as it learns, adapts, and makes the most of its environment.

Because MASCEM can simulate many types of markets, we are using it in several classes to show students the differences between each type of negotiation. We are also adding a simplified version of MASCEM to our website (www.mascem.com) to let the Web community learn about electricity markets.

In the future, we will continue to improve this strategy by including additional algorithms with new approaches, including other significant parameters such as weather and economy indexes that would allow greater coverage of unexpected situations. Along with refining the reinforcement learning techniques and the inclusion of clustering to group periods that present similar tendencies, this next step will provide

the algorithm with more data to analyze, by assuming that what was observed in one case can be projected to others that present similar characteristics. We also plan to test the implementation of a metalearner that combines the outcome of different predictive models to create the final result. ■

Acknowledgments

We acknowledge FCT, FEDER, POCTI, POSI, POCI, POSC, POTDC, COMPETE,

and the Knowledge Engineering and Decision Support Research Center (GECAD) unit for their support of this work.

References

1. M. Shahidehpour, H. Yamin, and Z. Li, *Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management*, Wiley-IEEE Press, 2002.
2. I. Praça et al., "MASCEM: A Multi-Agent System that Simulates Competitive Electricity Markets," *IEEE Intelligent Systems*, vol. 18, no. 6, 2003, pp. 54–60.
3. H. Li and L. Tesfatsion, "Development of Open Source Software for Power Market Research: The AMES Test Bed," *J. Energy Markets*, vol. 2, no. 2, 2009, pp. 111–128.
4. V. Koritarov, "Real-World Market Representation with Agents: Modeling the Electricity Market as a Complex Adaptive System with an Agent-Based Approach," *IEEE Power and Energy*, vol. 2, no. 4, 2004, pp. 39–46.
5. P. Kádár and M. Csapodi, "MASI—Market Simulator for Deregulated Energy Markets," *Proceedings of IEEE Intelligent System Application to Power Systems (ISAP 2001)*, 2001, pp. 213–218.
6. G. Marreiros et al., "Context-Aware Emotion-Based Model for Group Decision Making," *IEEE Intelligent Systems*, vol. 25, no. 2, 2010, pp. 31–39.
7. R. Zimmerman and D. Gan, *Matpower Manual*, Cornell Univ. Press, 1997.
8. T. Rahwan and N. Jennings, "Coalition Structure Generation: Dynamic

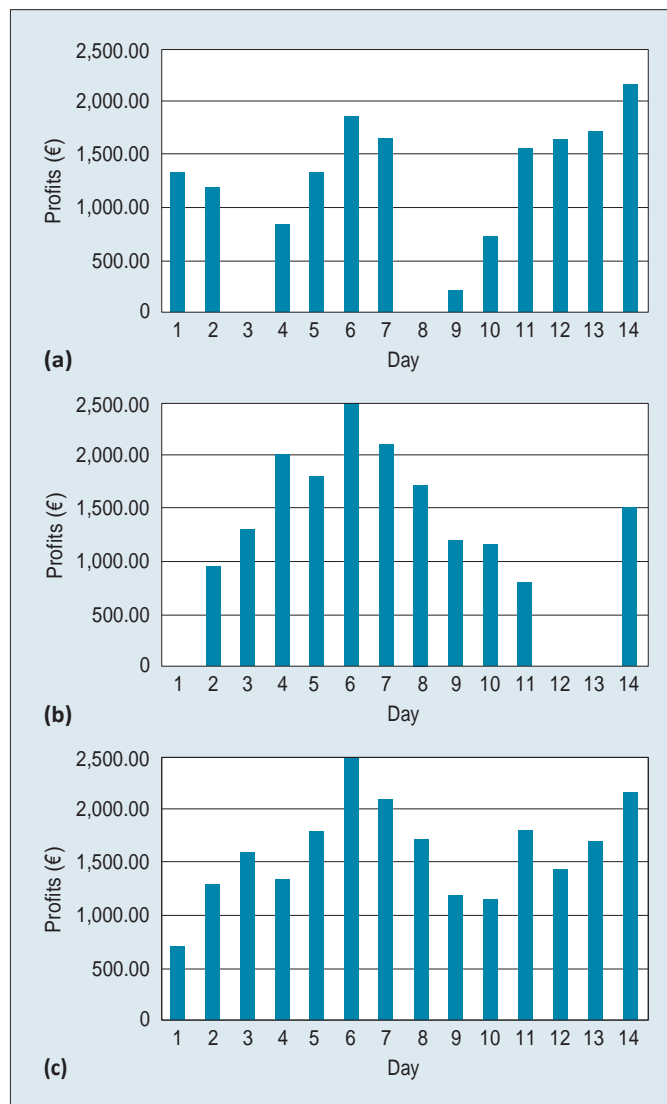


Figure 6. Seller 2's profits in the first period of the day, over a period of 14 days. Three separate simulations used (a) the regression on the data of the last five business days, (b) a neural network, and (c) our proposed strategy for bid definition, respectively.

- Programming Meets Anytime Optimization," *Proc. 23rd Conf. Artificial Intelligence (AAAI)*, AAAI Press, 2008, pp. 156–161.
9. P. Oliveira et al., "MASCEM—An Electricity Market Simulator Providing Coalition Support for Virtual Power Players," *Proc. Int'l Conf. Intelligent System Application to Power Systems*, IEEE Press, 2009, pp. 1–6.
 10. F. Azevedo and Z. Vale, "Forecasting Electricity Prices with Historical Statistical Information Using Neural Networks and Clustering Techniques," *Proc. Power Systems Conf. and Expo. (IEEE PES)*, IEEE Press, 2006, pp. 44–50.
 11. L. Cao, V. Gorodetsky, and P. Mitkas, "Agent Mining: The Synergy of Agents and Data Mining," *IEEE Intelligent Systems*, vol. 24, no. 3, 2009, pp. 64–72.
 12. K. Gopalratnam and D. Cook, "Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm," *IEEE Intelligent Systems*, vol. 22, no. 1, 2007, pp. 52–58.
 13. N. Amjady, A. Daraeepour, and F. Keynia, "Day-Ahead Electricity Price Forecasting by Modified Relief Algorithm and Hybrid Neural Network," *IET Generation, Transmission & Distribution*, vol. 4, no. 3, 2010, pp. 432–444.
 14. L. Wu and M. Shahidehpour, "A Hybrid Model for Day-Ahead Price Forecasting," *IEEE Trans. Power Systems*, vol. 25, no. 3, 2008, pp. 1519–1530.
 15. M. Negnevitsky, P. Mandal, and A. Srivastava, "An Overview of Forecasting Problems and Techniques in Power Systems," *Proc. Power and Energy Soc. General Meeting*, IEEE Press, 2009, pp. 1–4.
 16. Z. Jing et al., "Study on the Convergence Property of RE Learning Model in Electricity Market Simulation," *Proc. 8th Int'l Conf. Advances in Power System Control, Operation and Management (APSCOM 2009)*, IEEE Press, 2009, pp. 1–5.
 17. A. Greenwald and J. Kephart, "Shopbots and Pricebots," *Proc. 16th Int'l Joint Conf. Artificial Intelligence (IJCAI)*, Morgan Kaufmann, 1999, pp. 506–511.
 18. K. Korb and A. Nicholson, *Bayesian Artificial Intelligence*, Chapman & Hall/CRC, 2003.
 19. A. Dore and C.S. Regazzoni, "Interaction Analysis with a Bayesian Trajectory Model," *IEEE Intelligent Systems*, vol. 25, no. 3, 2010, pp. 32–40.