Metalearning to Support Competitive Electricity Market Players' Strategic Bidding

Tiago Pinto [1, *], Tiago M. Sousa[1], Hugo Morais[2], Isabel Praça[1], Zita Vale[1]

[1] *GECAD - Knowledge Engineering and Decision Support Research Centre - Polytechnic of Porto (IPP)*
R. Dr. António Bernardino de Almeida, 431, 4200-072 Porto, Portugal

[2] *AUTomation and Control Group – Technical University of Denmark (DTU)*
Richard Petersens Plads, Building 326, 118, 2800, Kgs. Lyngby, Denmark

ABSTRACT. Electricity markets are becoming more competitive, to some extent due to the increasing number of players that have moved from other sectors to the power industry. This is essentially resulting from incentives provided to distributed generation. Relevant changes in this domain are still occurring, such as the extension of national and regional markets to continental scales. Decision support tools have thereby become essential to help electricity market players in their negotiation process. This paper presents a metalearner to support electricity market players in bidding definition. The proposed metalearner uses a dynamic artificial neural network to create its own output, taking advantage on several learning algorithms already implemented in ALBidS (Adaptive Learning strategic Bidding System). The proposed metalearner considers different weights for each strategy, based on their individual performance. The metalearner's performance is analysed in scenarios based on real electricity markets data using MASCEM (Multi-Agent Simulator for Competitive Electricity Markets). Results show that the proposed metalearner is able to provide higher profits to market players when compared to other current methodologies and that results improve over time, as consequence of its learning process.

KEYWORDS: Adaptive Learning, Artificial Neural Network, Electricity Markets, Metalearning, Multi-Agent Simulation.

[*] Corresponding author: Tiago Pinto is with GECAD, Dr. António Bernardino de Almeida, 431, 4200-072 Porto, Portugal; Tel.: +351 22 8340500; Fax: +351 22 8321159, WebSite: http://www.gecad.isep.ipp.pt / E-mail: tmcfp@isep.ipp.pt

1. Introduction

Electricity Markets (EM) have been introduced in several countries in the early 1980s, and the worldwide spread has reached its peak during the 1990s. Since then, regulator entities have done several reforms in order to guarantee the market competition and transparency [1]. The majority of European countries have already joined together into common market operators, resulting in joint regional EM composed of several countries. Additionally, in early 2015, several of these European EM have been coupled in a common market platform, operating on a day-ahead basis [2].

EM are environments with significant dynamic characteristics due to their restructuring [3]. This process was conducted with the purpose of increasing the competition in this sector, leading to a decrease in energy prices. In the future, EM prices are expected to be more volatile, depending on the renewable based generation, especially wind and solar [4]. The complexity in EM operation also suffered an exponential increase, bringing new challenges to players' participation [5]. In order to overcome these challenges, it became essential for the market entities to fully understand the principles of EM, and how to evaluate their investments in such a competitive environment. The need for understanding those mechanisms and how the involved players' interaction affects the outcomes of EM contributed to increase the use of simulation tools. Multi-agent based software is particularly well fitted to analyze dynamic and adaptive systems with complex interactions among its constituents [6-9]. Several EM simulators have been developed, allowing the study of different EM types and models. Relevant examples in this context are: AMES Wholesale Power Market Test Bed [6], EMCAS - Electricity Market Complex Adaptive System [7], or GAPEX - Genoa Artificial Power-Exchange [9].

Although the existing simulators include some machine learning capabilities, the decision support that is provided is not yet adequately explored. In order to overcome this gap, MASCEM (Multi-Agent System for Competitive Energy Markets) [8, 10] has been developed. MASCEM supports EM simulation, considering an extensive set of different market models, and including all the most important entities that take part in such transactions. The decision support of EM players is performed through ALBidS (Adaptive Learning strategic Biding System) [11], which considers several different approaches to analyse market data and to define adequate action profiles for market players.

It is in the decision support field that this paper gives its main contribution. Using the outputs of ALBidS' strategies (bid prices), a new methodology is proposed, which creates a new strategic proposal, by combining the existent ones using the metalearning concept [12]. The proposed approach is defined as a metalearner since it uses meta-data related to the performance confidence values that each strategy is obtaining, in order to define the weights with whom each strategy affects the metalearner's output, thus improving the performance of existing learning algorithms. The combination of meta-data (bid prices resulting from the other strategies outputs and confidence values of each of the strategies) is done using a dynamic artificial neural network (ANN). This way the metalearner is able to adapt its results, giving higher influence to the results of the best strategies, while lowering the contribution of the strategies which are presenting worst results.

After this introduction, an overview of MASCEM and ALBidS is presented in section 2. Section 3 describes the proposed ANN based metalearner. In Section 4 simulation results using the proposed approach, considering real EM data are presented. Finally, section 5 presents the main conclusions and contributions of this work.

2. MASCEM and ALBidS

2.1. MASCEM simulator

MASCEM [8, 10] is a modelling and simulation tool with the purpose of studying complex restructured EM operation. MASCEM models the complex dynamic market players, including their interactions and medium/long-term gathering of data and experiences. The main goal of MASCEM is to simulate as many market models and player types as possible, so it can reproduce, in a realistic way, the operation of real EM. This enables it to be used as a simulation and decision-support tool for short/medium term purposes but also as a tool to support long-term decisions, such as those taken by regulators. Unlike traditional tools, MASCEM does not postulate a single decision maker with a single objective for the entire system. Rather, it allows agents representing the different independent entities in EM to establish their own objectives and decision rules. Moreover, as the simulation progresses, agents can adapt their strategies based on previous successes or failures. MASCEM's key players reflect actual entities from real markets and provide a means for aggregating consumers and producers.

MASCEM includes several negotiation mechanisms usually found in electricity markets, being able to simulate pool markets, bilateral contracts, balancing markets, forward markets, and ancillary services. The different market types offer players the chance to approach market negotiations strategically, taking advantage of the several opportunities that arise at each time. The need for adequate decision support to the strategic behaviour of negotiating market players must be addressed by means of intelligent approaches that can take the most advantage out of the surrounding environment and context. Metalearners have an important role, as they use meta-data to improve the performance of existing

learning algorithms. Using meta-data derived from other learning algorithms, a metalearner creates flexibility in solving different kinds of learning problems [12], especially when dealing with dynamic environments with a large associated uncertainty, such as EM.

The need for this type of decision support extends, not only to the requirement of maximizing players' gain from market participation (maximization of profits or minimization of costs), but also to the improvement of several other essential features of intelligent agents, such as [13]: (i) the capability of interacting and reacting on the same environment as other intelligent agents, (ii) social abilities that allow the interaction with other agents, (iii) autonomy to enable agents deciding and controlling their own actions, (iv) learning abilities, to allow an agent to change its behaviour based on prior experience, and (v) flexibility, so that agents' tasks do not need to be pre-determined, rather adapted according to previous events and to current context of the environment.

With the purpose of providing decision support to market players, ALBidS has been developed and integrated with MASCEM [11], and has been improved by integrating the metalearner proposed in this paper.

2.2. ALBidS decision support system

ALBidS integrates several strategies, taking advantage of each one in different contexts. The algorithms are placed below the main reinforcement learning algorithm (RLA), which allows that in each moment and circumstance the technique that presents the best results for the current scenario is chosen as the system's response [11]. ALBidS is implemented as a multiagent system itself, where each method is implemented in an individual agent.

The diversity of algorithms that are used by ALBidS bring out the need for the development of a mechanism that is able to manage the balance between the Efficiency and

Effectiveness (2E) of the system. This mechanism provides the means for the system to adapt its execution time to the purpose of the simulation, i.e., if the expected results from ALBidS are as best as it is able to achieve, or, on the other hand, if the main requirement is for the system to be executed rapidly. The 2E Management mechanism decides which methods are used at each moment; depending on their performance in terms of efficiency and effectiveness. In this way certain strategies can be excluded when they are not fulfilling ALBidS' requirements. Strategies are also manipulated internally, so that they adapt their individual results quality/execution time balance to each simulation's needs.

A highly dynamic environment such as the EM forces players to be equipped with tools that allow them to react to diverse negotiation circumstances. The existence of a variety of different strategies grants ALBidS the capability of always being prepared for the diversity of situations that a market player may encounter during market negotiations. The different natures of the considered strategies offer coverage over a diversity of areas, guaranteeing a high probability that there is always one strategy suited for each different context. The considered strategies are [11]: Based on statistical approaches; Composed Goal Directed; Adapted Derivative-Following [14]; Market Price Following; Dynamic Feed Forward ANN [15]; Adaptation of the AMES bidding strategy [6]; Simulated Annealing-Q-Learning [16]; Game Theory [17]; Economic Analysis [18]; Determinism Theory [19]; and Error Theory.

The main RLA, which is responsible for choosing among the various strategic alternatives, presents a distinct set of statistics for each context. This means that an algorithm that may be presenting good results for a certain context may possibly never be chosen as the answer for a distinct context [11]. ALBidS provides three alternative reinforcement learning algorithms, namely: (i) a simple reinforcement learning algorithm that updates strategies' confidence values according to the absolute value of the difference

between the prediction and the real value; (ii) the revised Roth-Erev reinforcement learning algorithm [6], which besides the features of the previous algorithm, also includes a weight value for the definition of the importance of past experiences; (iii) learning algorithm based on the Bayes theorem of probability [20], which updates the values through the propagation of the probability of each algorithm being successful given its past performance.

3. ANN based Metalearner

3.1. Metalearner description

Metalearning means learning about learning. This type of approach intends to take control of the process of learning itself [21]. The meta-data used by a metalearner can be related with details of the data used for learning, or even particularities of the learning algorithms that are used to learn from. Based on the concept of metalearner, the proposed approach uses as inputs the outputs of the various strategic approaches of ALBidS, i.e. bid prices suggested by each strategy, and also assigns importance weights to each of these inputs. These weights enable the metalearner to adapt its output, giving higher importance to the strategies that are proving to be more adequate, while partially or completely ignoring the contribution of the strategies which are presenting worse results. The final output of the metalearner is a bid price, which is considered by the reinforcement learning algorithms of ALBidS' Main Agent as a proposal, similarly to those of all the other strategies' outputs. This means that the output of this strategy is not a direct suggestion to the supported market player, but a proposal to the Main Agent, which considers it as an option when deciding the final action for the market player.

Considering the learning process of the other strategies with different importance weights allows the metalearner to adapt its output according to the observed results of each one. The

weights used for defining the strategies relevance for the metalearner are based on the confidence values of the main reinforcement learning algorithm already implemented in ALBidS. The reinforcement learning algorithm's confidence values are adapted and updated according to the results each strategy is achieving, hence being exactly what this metalearner requires: understanding which are the strategies' outputs that it should consider as most influent for the final output. The combination of the several inputs, including the bid prices suggested by all the learning algorithms, and the weight values associated to each of those responses, is achieved through the application of a dynamic ANN. The ANN is trained with the historic values of the strategies' outputs (bid prices) and their confidence values at each time, by validating these inputs with the log of actual market prices verified during the considered periods. Fig. 1 presents a flowchart of the metalearning process, including the initialization of the ANN, the training process using historic data, and the execution of the ANN metalearner. The process explanation is provided in section 3.2.

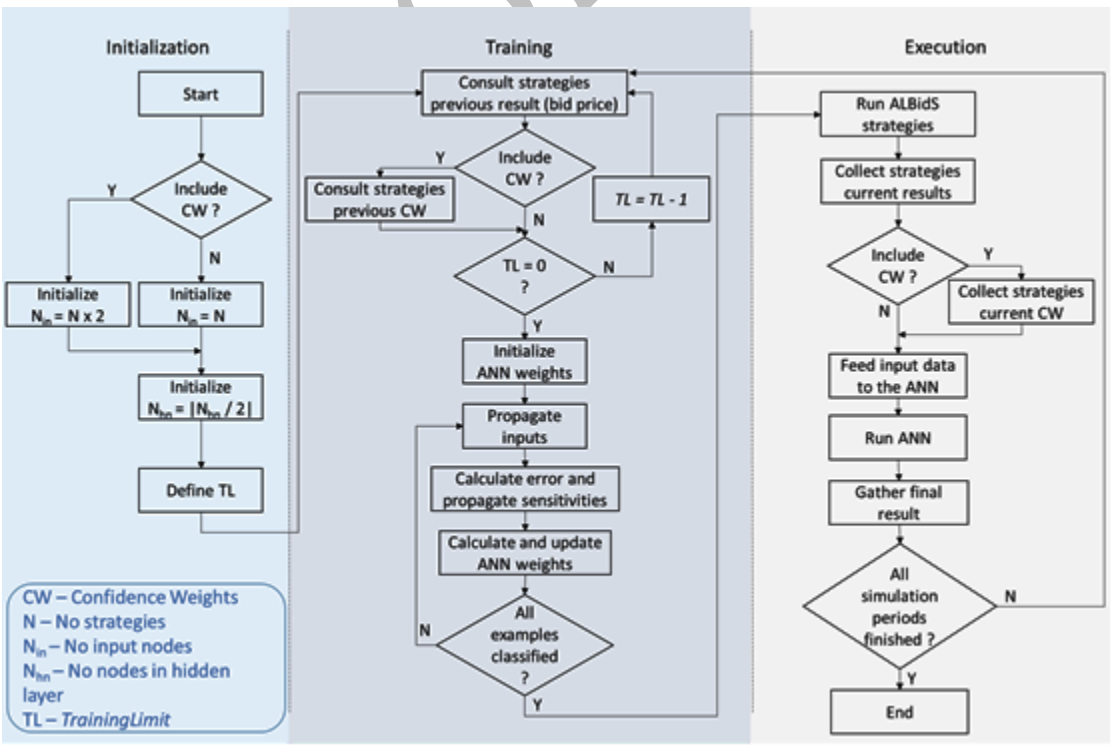


Fig. 1. Flowchart of the metalearning process

3.2. Dynamic artificial neural network

The main contribution of this paper is the development of a metalearner based on a dynamic ANN. The dynamism of the ANN is achieved by a retraining of the network in each iteration, so that it can always consider the most recent information. The considered ANN is a feedforward ANN, receiving as inputs the meta-data concerning ALBidS strategies outputs (bid prices), and, when required, their respective confidence weights. The ANN considers an adaptive intermediate layer, and one output: the suggested bid price.

The hidden layer and the respective nodes play a very important role in many successful applications of neural networks. The nodes in the hidden layer are the means that allow neural networks to detect the feature, to capture the pattern in the data and to perform complicated nonlinear mapping between input and output variables. Much theoretical work has been done in the area, showing that most authors use only one hidden layer [22].

The problem of determining the optimal number of hidden nodes is very important to obtain a good prediction. In general, networks with fewer hidden nodes are preferable as they usually have better generalization ability, thus dealing better with the overfitting problem. But networks with too few hidden nodes may not have enough power to model and learn the data. There is no universal rule to select this parameter, although some systematic approaches are reported, e.g. Jeff Heaton [22] proposes the following guidelines:

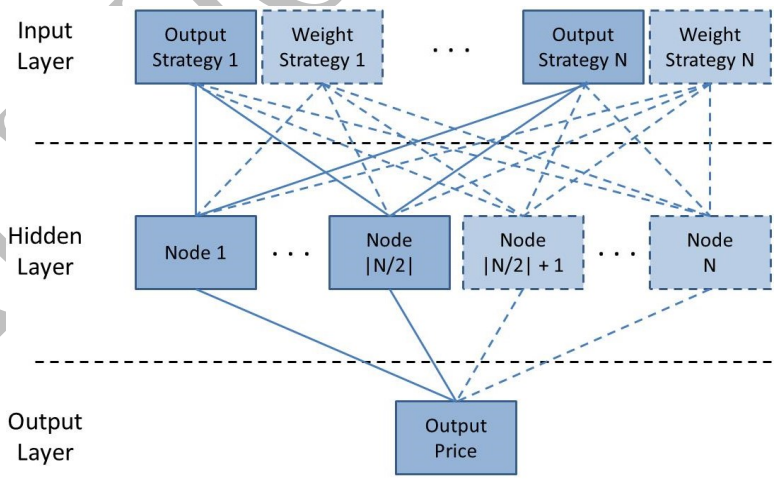
- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
- The number of hidden neurons should be less than twice the size of the input layer.

Other authors have proposed that the number of hidden nodes should be calculated according to the input node number (n) in the following proportions: 2n [23], n [24], n/2 [25]. However, since all problems present distinct characteristics the only way to determine the ideal number of hidden nodes is through experience or trial and error [22].

After a large set of testing and refinement runs, the conclusion was that the ideal number of intermediate hidden nodes should be about half the number of the input nodes. Therefore, and given that the number of inputs is also dynamic (dependent on the number of strategies used by ALBidS at each time, and the inclusion or not of the respective strategies' confidence weights), the number of intermediate nodes must be adaptive as well. The number of hidden nodes (N_{hn}) is calculated as in (1), where N_{in} is the number of input nodes, making the intermediate layer dependable on the number of inputs, assuming a number that is always around half the number of inputs.

$$N_{hn} = \lfloor N_{in} / 2 \rfloor \quad (1)$$

The structure of the proposed ANN considering N ALBidS strategies is presented in Fig. 2.



Proposed ANN topology Fig. 2.

From Fig. 2 it is visible that the number of input nodes can be equal to the number of strategies used by ALBidS (N), or twice that number, when the strategies' confidence

values are used as strategies' weights. The number of hidden nodes in the intermediate layer is half the number of input nodes (N when using weights, or $|N/2|$ when not using these values). The dashed lines in Fig. 2 represent the optional nodes and respective connections (related to strategies' weights, which are optional for the metalearning process).

Backpropagation using the gradient descent method [15] has been used as training algorithm for the ANN. This requires calculating the derivative of the squared error function with respect to the weights of the network. The squared error function E for the single output neuron is defined as in (2).

$$E = \frac{1}{2}(t - y)^2 \tag{2}$$

where t is the target output for a training sample, and y is the output of the output neuron.

For each neuron j, its output o_j is defined by feedforward calculation, as in (3).

$$o_j = f\left(\sum_{k=1}^{n} w_{kj} x_k\right) \tag{3}$$

where n is the number of input units to neuron j, and w_{kj} is the weight between neurons k and j. Hence, the input for the activation function f of a neuron is the weighted sum of outputs o_k of the previous neurons. The used activation function f is the logistic function, a log-sigmoid function, which can be defined as in (4).

$$f(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

The backpropagation algorithm includes the following steps [15]:

1. Initialize weights as small random numbers;
2. Introduce training data to the NN and calculate the output by propagating the input forward through the network using (3);
3. Calculate the error using (2);

4. Propagate the sensitivities backward through the network by simply taking the derivative of the activation function (4) with respect to the network parameters;
5. Calculate w_{kj} updates;
6. Update the values of w_{kj};
7. Repeat steps 2 to 6 until all examples are classified correctly.

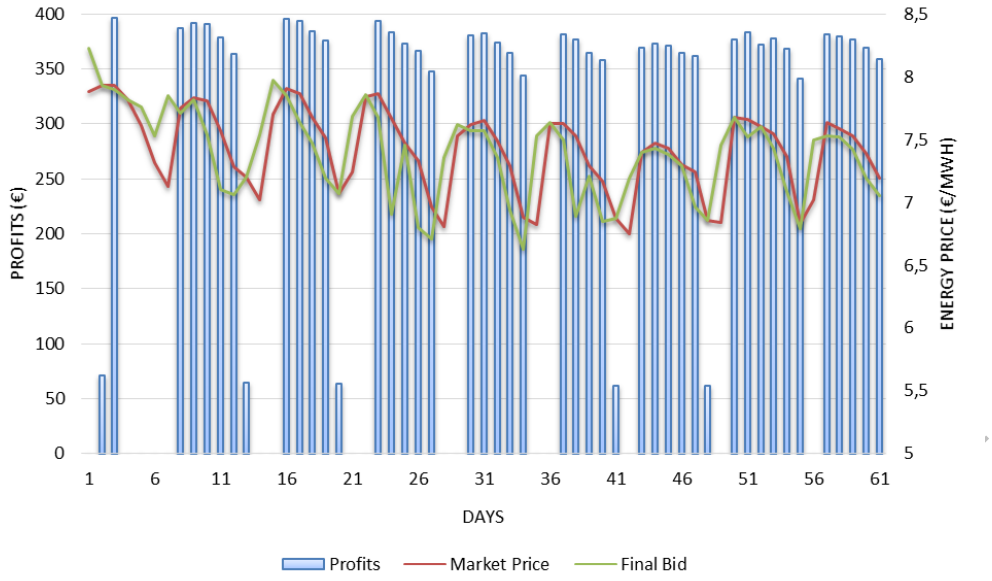
The number of previous days considered for training the dynamic ANN metalearner is defined by the *TrainingLimit* variable. This variable indicates whether the ANN will use more or less values for training, meaning a faster but actually worse forecast, or a slower but more effective response. Considering an exaggerated amount of data may lead to over-training, making the ANN memorize the examples instead of learning the relationship between data. Also, it may lead to the consideration of inadequate data from a long time before, which is likely to bring no added value to the learning process.

Fast execution times are critical for situations when a response must be given quickly. However, execution time grows as the amount of data used for the ANN training increases. This balance is important for the 2E balance mechanism of ALBidS. The proposed metalearner varies its execution time depending on the amount of data that is used from training the ANN, *i.e.*, the *TrainingLimit* variable. Moreover, although the *TrainingLimit* variable defines the number of days that will be considered as training data for the metalearner, not all the data from these days is used. Only the information regarding the strategies outputs and respective confidence values, related to the same context as the current one will be used. This selection is performed using the context analysis mechanism of ALBidS, and it is used so that the data used for training the ANN is only the most relevant data given the specific situation at each time, rather than using all available information, which could mislead the ANN into wrong directions.

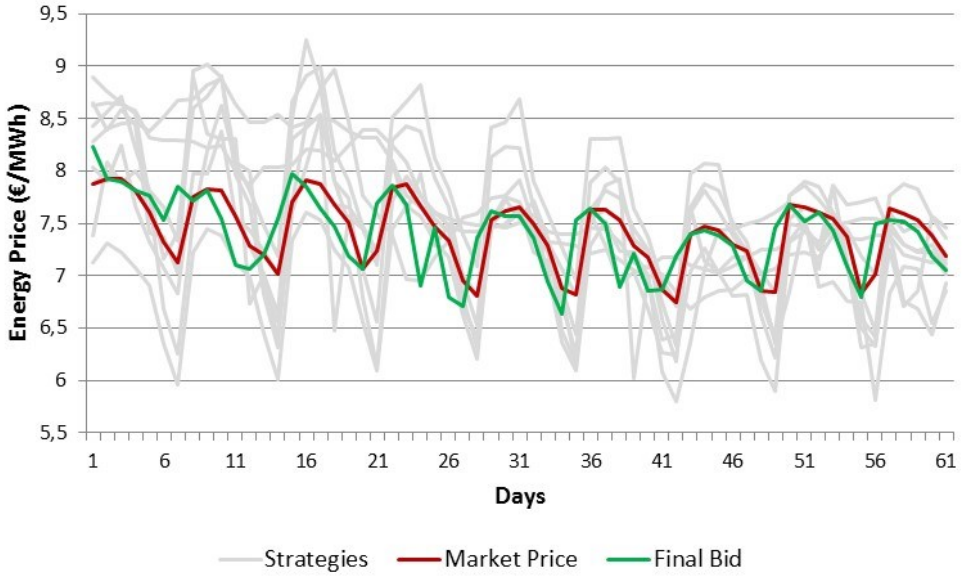
4. Experimental Findings

This case study presents two simulations undertaken using MASCEM. Real data extracted from the Iberian market - MIBEL [26] has been used. The considered scenario includes 7 buyer agents and 10 seller agents, acting in an auction based day-ahead spot market [4, 26]. This scenario has been described in detail in [10, 11]. Simulations concern 61 consecutive days, starting from October 15th 2012. In the first simulation Seller 2 uses the proposed metalearner without considering strategies' weights; in the second one it uses the proposed metalearner with strategies' confidence weights as complementary inputs for the ANN. The main purposes of the presented results are: to demonstrate the use of the proposed metalearner in supporting a market player's decisions; and to show the influence of the supporting strategies in this metalearner's performance. The reinforcement learning algorithm used by ALBidS' Main Agent is the Bayes Theorem algorithm. The confidence values that this algorithm defines for each strategy are used by the proposed metalearner. Additionally, as supporting strategies, nine ALBidS strategies have been used. Fig. 3 presents the incomes achieved by Seller 2 in the first period of the 61 considered days of the first simulation, and a comparison between the proposed bid and the market price.

Analysing Fig. 3 it is visible that the metalearner starts by achieving bad results, which improve over time. This is due to some supporting strategies' worst suggestions at the start, while they do not have the experience to learn adequately. As this metalearner considers all suggestions in a similar way, the good outputs that some strategies may be presenting are muffled by the bad ones. As time progresses and strategies start to learn and provide better individual results, the metalearner's results improve as well. Fig. 4 presents a comparison between the metalearner's bid price and the supporting strategies' proposals.



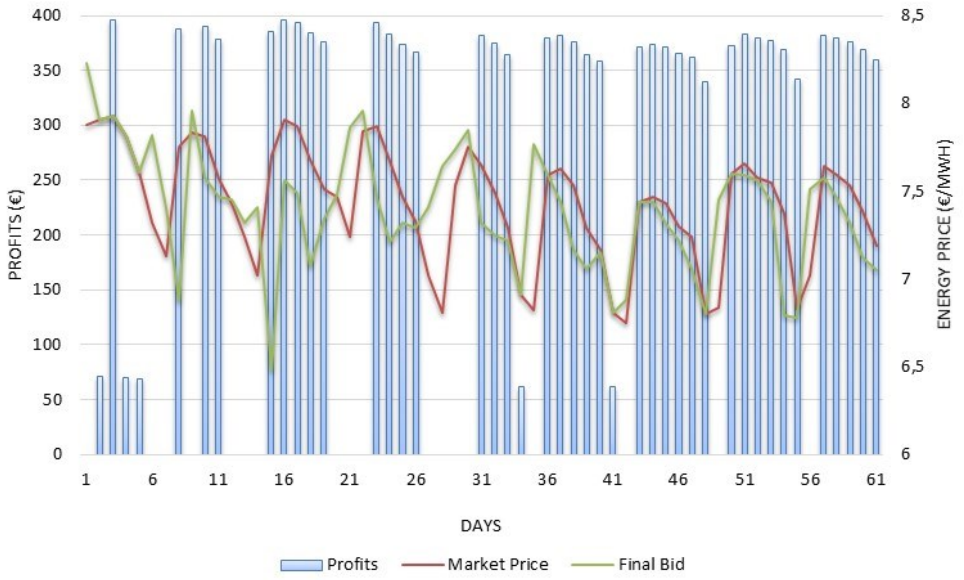
Results of the Metalearner without confidence weights, in the 1st period of the 61 days Fig. 3.



Metalearner without confidence weights and supporting strategies' bids Fig. 4.

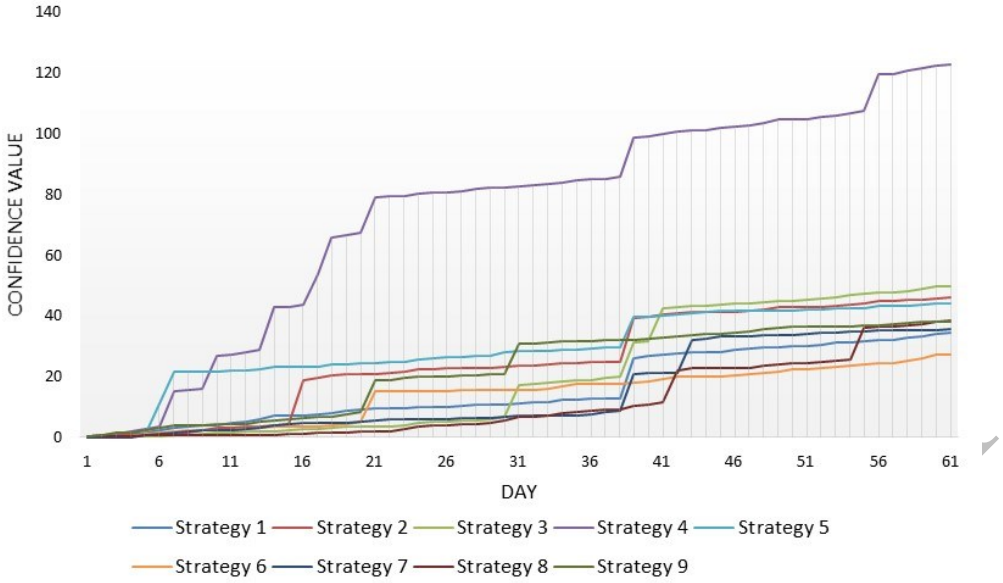
Fig. 4 shows that, during the first days, strategies' proposals present a higher variance; hence the metalearner's bid follows that variance trend. In the later days, proposals start to converge and improve their quality through the individual learning of each strategy; consequently, the metalearner's bid gets closer to the market price. The Metalearner's

usage of all the suggestions in an equal way results in a bad performance during the first days. This is why the use of the strategies' weights as inputs for the Metalearner is essential. This method considers adequate weights to instigate the attribution of higher importance to strategies that present the best results at each time. Fig. 5 presents Seller 2 results using the proposed metalearner considering strategies' confidence values as meta-data in the form of weights. These results concern the first period of the 61 simulated days.



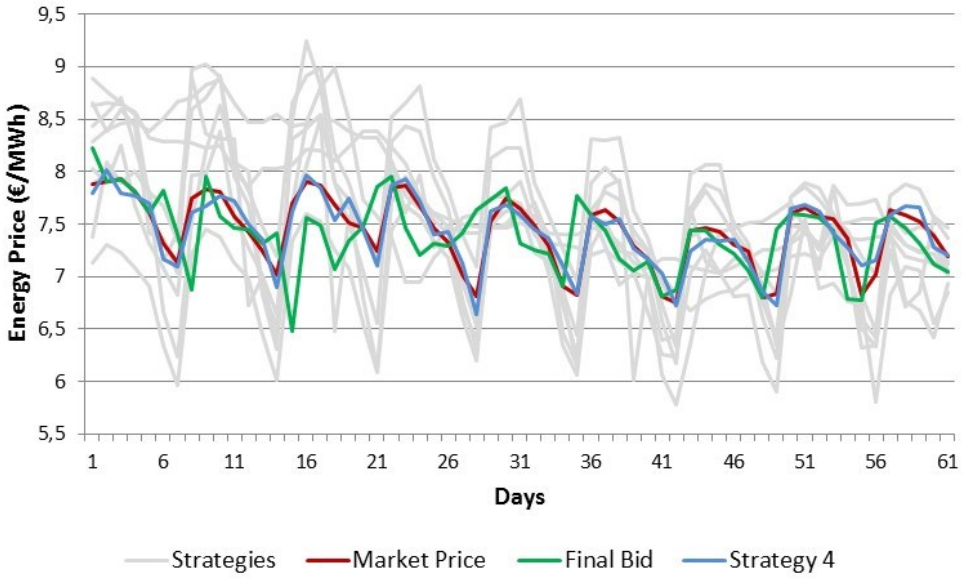
Results of the Metalearner with confidence weights, in the 1st period of the 61 days Fig. 5.

Fig. 5 shows that these results follow the same tendency as the metalearner without strategies' weights: with bad results at first, and a visible improvement over time. However, in this situation it is able to achieve good results earlier, proving the advantage of attributing higher weights to the most appropriate strategies' proposals. Also, by taking advantage on the evolution of strategies' learning process, the metalearner improves the quality of its results as well as time progresses. Fig. 6 presents the strategies' confidence values, which determine the attributed weights.



Supporting strategies' confidence values Fig. 6.

From Fig. 6 it is visible that Strategy 4 detaches from the others; i.e. this strategy has the higher influence on the metalearner's output, increasing over time. Fig. 7 shows how this metalearner's bid follows the Strategy 4's responses, comparing to the other strategies.



Metalearner's and supporting strategies' bids Fig. 7.

Fig. 7 shows that even in the first days the metalearner's bid stays close to the market price, influenced by Strategy 4. This tendency remains visible throughout the days. Comparing the metalearner's performance with and without the use of strategies' weights as inputs, a much more constant behavior is found when using the confidence weights. The adjustment of the final bid, taking into account the strategies' results has proven to be an added value in adjusting the metalearner's results. Fig. 8 presents the comparison of the profits achieved by Seller 2 when using the proposed metalearner with and without confidence weights, and the results of each of the supporting strategies, in the total of the 24 hourly periods of negotiation of the 61 considered days.

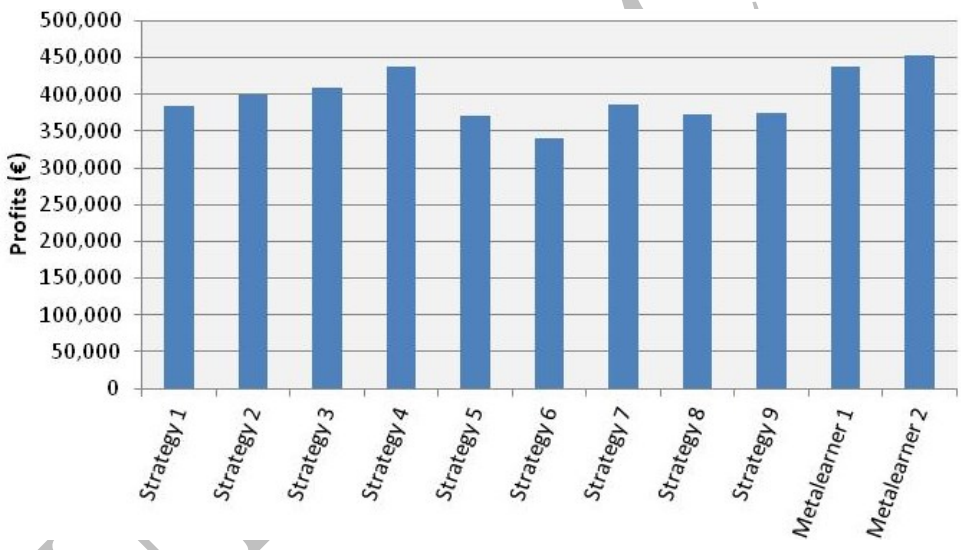


Fig. 8. Total profits of the proposed metalearners (using and not using confidence weights) and of the supporting strategies, in the total of the 24 periods of the 61 considered days

From Fig. 8 it is visible that the proposed metalearner using confidence weights (Metalearner 2 in Fig. 8) is able to achieve the best results from all strategies (with a total of €452.531,38). These results surpass the ones achieved by the supporting strategy that reached the best results (Strategy 4, with a total of €437.191,12), and which has consequently been attributed the higher confidence weight (as seen by Fig. 6). Although the

high confidence weight of Strategy 4 makes the values proposed by the metalearner (when using confidence weights) become closer to the ones proposed by this strategy, the other supporting strategies still provide their contribution to the final output of the metalearner, although with smaller influence on the metalearner's output. This has proven to be advantageous, since the metalearner was not restricted to using the same output values of Strategy 4, and the consideration of other strategies' proposals has led the metalearner to achieve even more advantageous bidding values as its output. The metalearner results when not using confidence weights (Metalearner 1 in Fig. 8) are below the ones achieved by the metalearner using confidence weights, but they are very close to the profits achieved by Strategy 4, and superior to all other supporting strategies' results (total of €436.890,64). These results mean that even without confidence weights, the metalearner is able to choose the supporting strategies that should be used at each time, leading to promising results.

5. Conclusions and future work

The metalearning concept – learning about learning– is an important asset for the machine learning field. When enhanced by solid machine learning approaches, e.g. ANN, it becomes a powerful tool for decision support. This paper presents a new metalearner for decision support to EM players, using meta-data concerning other strategic approaches to create its own output through the use of a dynamic ANN.

The metalearner achieves good results in the market, both with and without the use of strategies' weights as meta-data. Most importantly, improvement is visible over time, taking advantage on the learning abilities of the available strategies. Using confidence weights as inputs, the metalearner is able to adapt its outputs, highlighting strategies that achieve the best results in each negotiating context. The ability to improve the performance

over time is accentuated by the achievement of increasing profits on behalf of the supported market player, as demonstrated by the case studies based on real EM data. The profits achieved by the proposed metalearner using the supporting strategies' confidence weights are higher than the ones achieved by all the other supporting strategies, proving the advantage and breakthrough provided by the proposed methodology.

As future work, the implementation of heuristic processes to determine the optimal number of nodes in the hidden layer of the ANN is proposed. Additionally, the experimentation of different types of ANN and other methods such as neuro-fuzzy inference systems and support vector machines, is also expected.

Acknowledgements

This work is supported by FEDER Funds through COMPETE program and by National Funds through FCT under projects FCOMP-01-0124-FEDER: PEst-OE/EEI/UI0760/2011, PTDC/SEN-ENR/122174/2010 and SFRH/BD/80632/2011 (Tiago Pinto PhD).

References

[1] D. Huppmann, R. Egging, "Market power, fuel substitution and infrastructure – A large-scale equilibrium model of global energy markets", Energy, 75, 483-500, 2014

[2] PCR report, "EUPHEMIA: Description and functioning", May 2014, Available: http://www.apxgroup.com/wp-content/uploads/Euphemia-Public-Documentation-to-be-published.pdf [accessed in November 2015]

[3] A. Boersen, B. Scholtens, "The relationship between European Electricity Markets and emission allowance futures prices in phase II of the EU (European Union) emission trading scheme", Energy, Volume 74, Pages 585-594, September 2014

[4] C. Mendes, I. Soares, "Renewable energies impacting the optimal generation mix: The case of the Iberian Electricity Market", Energy, Volume 69, 1 May 2014, Pages 23-33

[5] F. P. Sioshansi, "Evolution of Global Electricity Markets – New paradigms, new challenges, new approaches", Academic Press, 2013

[6] H. Li and L. Tesfatsion, "Development of open source software for power market research: The AMES test bed," J. Energy Mark., 2009.

[7] V. S. Koritarov, "Real-world market representation with agents," IEEE Power Energy Mag., vol. 2, no. 4, pp. 39–46, Jul. 2004.

[8] I. Praca, C. Ramos, Z. Vale, and M. Cordeiro, "MASCEM: a multiagent system that simulates competitive electricity markets," IEEE Intell. Syst., vol. 18, 2003.

[9] S. Cincotti and G. Gallo, "The Genoa Artificial Power-Exchange," Agents Artif. Intell., 2013.

[10] G. Santos, et al, "Multi-Agent Simulation of Competitive Electricity Markets: Autonomous systems cooperation for European Market modelling", Energy Conversion and Management, vol. 99, pp. 387-399, July 2015.

[11] T. Pinto, et al, "Adaptive Learning in Agents Behaviour: A Framework for Electricity Markets Simulation", Int. Comp-Aided Eng., 21, 4, 399-415, 2014.

[12] I. Bruha, "Meta-Learner for Unknown Attribute Values Processing: Dealing with Inconsistency of Meta-Databases" J. Intell. Inf. Syst., v. 22, no. 1, pp. 71–87, Jan. 2004.

[13] M. Wooldridge, "An Introduction to Multiagent Systems", Wiley, New York, 2002

[14] A. Greenwald and J. Kephart, "Shopbots and pricebots," Agent Mediat. Electron. Commer. II, 2000.

[15] K. Rudd, G. Di Muro and S. Ferrari, "A Constrained Backpropagation Approach for the Adaptive Solution of Partial Differential Equations," IEEE Trans. on Neural Networks and Learning Systems, vol. 25, no. 3, pp. 571,584, Mar. 2014.

[16] C. Juang and C. Lu, "Ant colony optimization incorporated with fuzzy Q-learning for reinforcement fuzzy control," Syst. Man Cybern. Part A, 2009.

[17] C.G. Min, M.K. Kim, J.K. Park, Y.T. Yoon, Game-theory-based generation maintenance scheduling in electricity markets, Energy, 55, 10-318, June 2013

[18] M. Porter, "How competitive forces shape strategy," Strateg. Plan. Readings, 2000.

[19] B. Berofsky, "Determinism,", Princeton University Press, Princeton, 1971.

[20] K. Korb and A. Nicholson, "Bayesian artificial intelligence", Second Edition (Chapman & Hall/CRC Computer Science & Data Analysis), December 2010.

[21] J. Biggs, "The role of meta-learning in study process," Br. J. Educ. Psychol., vol. 55, pp. 185–212, 1985.

[22] Jeff Heaton, "Introduction to Neural Networks for Java", 2nd Edition, Heaton Research, 2008

[23] Phat, V.N.; Trinh, H., "Exponential Stabilization of Neural Networks With Various Activation Functions and Mixed Time-Varying Delays", Neural Networks, IEEE Transactions on vol.21, no.7, pp.1180-1184, July 2010

[24] Wong, F.S.,"Time series forecasting using backpropagation neural networks", Neurocomputing 2, 147–159, 1991

[25] Tang, Z., Fishwick, P.A., "Feedforward neural nets as models for time series forecasting", ORSA Journal on Computing 5 (4), 374 – 385, 1993

[26] MIBEL, "Mercado Ibérico de Electricidade," accessed October, 2014. [Online]. Available: http://www.mibel.com [accessed in November 2015]

Figure Captions List