Forecasting Negotiation Counterpart's Offers

A Focus on Session-long Learning Agents

Marisa Masvoula
Department of Informatics and Telecommunications
National and Kapodistrian University of Athens
Athens, Greece
marisa.diuoa@gmail.com

***Abstract*— Predictive decision making is characteristic to current state of the art socio-technical systems that guide negotiation processes under electronic settings. Back end participants are particularly benefited by the use of models of computational intelligence, which help them adapt their strategy and evaluate risks and dynamics of the current negotiation. In this paper, the skill of forecasting the counterpart's future offers with the use of neural networks is investigated. Current systems base their learning models on data acquired from previous interactions. Such systems are once trained in an offline mode and are thereafter expected to operate in a real environment. However, when data distributions change, the systems no longer provide accurate estimations. A new perspective to the issue is introduced, by highlighting the need of learning during the negotiation session, with the use of "session-long learning" agents. These agents prove capable of capturing the negotiation dynamics by training their learning models with the data from the current negotiation thread. In this paper a static session-long learning agent, based on a simple neural network model, as well as an adaptive session-long learning agent, based on a neural network which evolves its structure and input features with the use of a genetic algorithm in each negotiation round, are presented and assessed.**

Keywords-Predictive negotiator; genetic algorithm; adaptive negotiation strategy; neural network applications

I. INTRODUCTION

Electronic Marketplaces (E-markets) is an important component of e-business that brings demand and supply of commodities and services into balance. They are the meeting places of producers and consumers that use exchange mechanisms, such as catalogues, negotiations, and auctions [1].

This paper is focused on the negotiation mechanism, defined as an iterative communication and distributed decision-making process, where participants, humans or agents, are searching for an agreement. Computer science has significantly contributed to the field, since the use of information systems has moved the negotiation arenas to electronic settings, and the development of models of computational intelligence has extended the cognitive abilities of negotiators. Current research efforts concentrate on the enhancement of support systems that assist negotiators, and on software agents that fully automate negotiation processes using learning techniques. One such technique is related to the ability of negotiators to forecast their counterpart's future offers and accordingly adjust their strategy. This research considers agent to agent interactions, and investigates the skill of forecasting the other party's responses.

In section 2, related work is presented and the value of forecasting, measured in terms of utility gain, is highlighted. In section 3, it is argued that neural network models trained with data from previous interactions are not capable of retaining their accuracy when used in open dynamic environments. The key issue is to retrain the neural networks involved, with data extracted from the current negotiation thread. Agents that retrain the employed networks are termed Session-long Learning. In section 4, two types of such agents are illustrated; Static session-long learning Agents (SSLAs), which are enhanced with a Multi-layer Perceptron (MLP) that has a static architecture, and Adaptive Session-long Learning Agents (ASLAs), which make use of an MLP that evolves its architecture and input features with a genetic algorithm. In section 5, the two types of session-long learning agents are compared, while in section 6, conclusions and future research issues are discussed.

II. TERMINOLOGY AND RELATED WORK

The outcome of a negotiation can be a compromise or a failure, and the gain (profit) an offer X incurs to participant (agent) α is measured by a utility function $U^{a}(X)$ that takes values in [0,1]. In multi-issue negotiations multiple attributes (issues) are considered negotiable and are exchanged between the engaged parties. Each offer X can be expressed as a vector in the n-dimensional space, where n is the number of issues under negotiation. For each issue, participants specify a range of permissible (reservation) values (a minimum and a maximum), which they are not willing to exceed. Additionally in many cases time is crucial and participants set a deadline indicating the maximum time they can spend in a negotiation encounter.

The specific rules of communication that guide the interaction constitute the negotiation protocol and determine the way messages are exchanged. The decision making rules or strategies are used to determine, select and analyze the decision alternatives. In a simple case where negotiation is

 ISBN: 978-1-61208-273-8

conducted between two non learning agents, alternatives are generated with the use of formal decision functions, and three groups of strategies are identified (time, behavior and resource dependent) as described in [2]. More sophisticated agents use AI-based techniques aiming to maximize the incurred utility. In the work presented in [3] categorization of such agents to those that follow explorative, repetitive and predictive strategies is given. The first category consists of agents that search the strategy space usually through trial-and-error learning processes, the second category consists of agents who repeat strategies that have proved efficient in past similar situations, while the third category consists of agents that adopt a strategy, based on estimations of environmental parameters and/or opponent.

This research is focused on the third category and particularly on the issue of predicting the counterpart's future offers. Predictive agents are distinguished into those who engage in single-lag predictions and estimate the very next offer of their counterpart, and into those who engage in multi-lag predictions and foresee future offers of their counterpart several time steps ahead. For applications of single-lag predictive decision making in negotiation support systems the interested reader may refer to [4,5], which present a neural network that simulates the possible responses to the alternative offers the negotiator is contemplating. For strategies of automated negotiators that are based on single-lag predictions the reader may refer to [6–9], where agents are developed with the scope to increase individual gain of the final outcome. In the case of multi-lag predictions, [10-12], demonstrate agents who decide to withdraw from pointless negotiations based on the forecasts of their counterpart's future values. Finally, Brzostowski and Kowalczyk [13-15] depict an agent who determines the sequence of optimal offers, "knowing" the sequence of opponent's responses.

This paper uses the protocol and strategy described in [9] where different agent strategies emerge from different attitudes towards risk. For a more thorough understanding a brief review of the strategy is provided. The negotiation environment considered is tied to bilateral (two parties are involved) multi-issue negotiations, where all issues are bundled and discussed together (package deal). The formal model of negotiation is comprised by the set of agents $A=\{a,b\}$, a finite set of quantitative issues under negotiation $I=\{i_1,i_2,\ldots,i_n\}$, the domain of reservation values $D_i^a:[\min_i^a,\max_i^a]$ for issue i attributed by agent α, and the deadline $T_{\max}^a$ of agent α, where $i\in$ I and $a\in$A. In the cases studied time variable t is discrete and expresses the interaction step (negotiation round). The possible outcomes of a negotiation can be understood in terms of utility $U^a(X_{(a\to b)}^t)$ where $X_{(a\to b)}^t=(\mathrm{x}_{1(a\to b)}^t,\mathrm{x}_{2(a\to b)}^t,\ldots,\mathrm{x}_{n(a\to b)}^t)^T$ is the negotiation offer sent from agent α to b at time t, and each x_i denotes the offered value of negotiable issue i. Each agent α is configured with a default strategy S^{α}, which determines the level of concession in each round [2]. In each time step t agent α estimates the next offer of his counterpart $\hat{X}_{(a\to b)}^t=(\hat{\mathrm{x}}_{1(a\to b)}^t,\hat{\mathrm{x}}_{2(a\to b)}^t,\ldots,\hat{\mathrm{x}}_{n(a\to b)}^t)^T$.

The decision rule makes use of the default strategy (S^{α}) of the predictive agent to generate offers until the detection of a "meeting point" (MP) with the "opponent". MP is a point which would result an established agreement if the agent was guided solely by his default strategy. When such point is detected, and according to the agent's attitude towards risk, agent risks staying in the negotiation in order to maximize the utility of the final agreement. At that point agent makes use of the estimation and refines the offer he sends at each time step. Two extreme attitudes can be generated: risk-seeking and risk-averse. The risk-seeking agent is willing to spend all the remaining time until expiration of his deadline engaging in an adaptive behavior to turn the estimations of his counterpart's responses to profit. On the other hand risk-averse agents follow a more conservative behavior when they detect an MP. They do not make any further concessions and insist on sending their previous offer, waiting for the opponent to establish an agreement.

In the following section, shortcomings of existing systems and the approach to address them is discussed.

III. PROBLEM STATEMENT

Methodologies that have been used for the purpose of forecasting the counterpart's future offers can be summarized into those based on statistical approaches (particularly non-linear regression) [10,14], mathematical models based on differences [13,15], and connectionist approaches, particularly some special types of neural networks [4-8,11,12].

Experiments have shown that mathematical models give poorer results when compared to non-linear regression models [14]. Non-linear regression models are restrictive, since they require the assumption of a known function form of the counterpart's behavior, and mathematical models are empirically proved less accurate than neural networks in the negotiation domain [16]. Focus is set on the application of neural networks, which can be utilized in the general case and have proved efficient in the problem of forecasting the counterpart's next offer.

However, two issues need to be addressed. The first concerns lack of homogeneity. Artificial Neural Networks (ANNs) employed by current state of the art negotiators have significant differences in terms of network architectures and input features. The second is that these models are particularly tied to bound domains, since in the majority they are trained and applied to environments with data of the same underlying distributions. The networks are trained before the initiation of the current negotiation instance with data from previous interactions, and are then set to operate in the current discourse. As a consequence, the predictors' accuracy depends heavily on data acquired from previous negotiations.

To address the second issue, this work highlights the need to retrain the MLPs with data acquired from the current

 ISBN: 978-1-61208-273-8

negotiation thread. In this respect two types of agents termed Session-long learning are developed and assessed. The first type, Static Session-long Learning Agent (SSLA), makes use of a Multi-layer Perceptron (MLP) with a static structure which is retrained at each predictive step, while the second, Adaptive Session-long Learning Agent (ASLA), makes use of an MLP which optimizes its structure and subset of input features during negotiation.

In the following sections SSLA and ASLA are described and compared.

IV. SESSION-LONG LEARNING AGENTS

A. *Static Session-long Learning Agent*

In this section we describe a Static Session-long Learning Agent (SSLA), which is defined as a session-long learning agent with a fixed MLP architecture during the discourse. Without loss of generality, the predictive agent is assumed to be the consumer who initiates the negotiation process at time $t_1=0$. The two agents take alternate turns until an agreement is established or until any of the two agents decides to terminate the procedure. In the general case, the forecasting tool of the SSLA makes use of the n previous counterpart's offers to estimate the next offer (at time $t+1$), as is illustrated in Figure 1. At time t the consumer formulates a new training set which is constructed from the series of the counterpart's offers.

It should be noted that in order to apply the Levenberg and Marquardt (LM) method, at least two training patterns are required, therefore the MLP is initially trained at round

$$t_{init} = 2*n+4 \tag{1}$$

The size of the dataset $|Dataset|$ at time $t \geq t_{init}$ is given by

$$|Dataset| = \frac{t}{2} - n \tag{2}$$

$|Dataset|$ is initially 2 in order to apply the LM method, and increases by 1 in each turn of the predictive agent. After training the MLP, SSLA makes use of the network to estimate his counterpart's next offer.

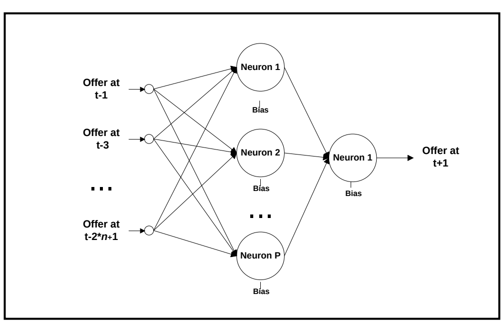

Figure 1. Forecasting tool of the negotiator.

More specifically, the actions an SSLA undertakes at each predictive round t are described as follows:

Step 1. Receive Opponent's Offer,

Step 2. Update Negotiation Thread by storing the received offer

Step 3. Formulate training set:

Consider a time series of the opponent's past offers: $\{X^{1}_{(\Pr->Con)}, X^{3}_{(\Pr->Con)}, ..., X^{t-1}_{(\Pr->Con)}\}$

Formulate the set of input-output patterns with respect to the number of input nodes

Step 4. Use the patterns yielded in Step 3 to train the network with the LM method

Step 5. Formulate current input pattern $\{X^{t-2*n+1}_{(\Pr->Con)}, ..., X^{t-1}_{(\Pr->Con)}\}$

Step 6. Apply input to the trained network

Step 7. Obtain forecast of opponent's next offer,

Step 8. Generate next offer based on the predictive strategy

The forecasting tool of the SSLA was selected to be very small and consist of three inputs (n=3), representing the three previous offers of the counterpart (as in [8]), and two hidden nodes (P=2). This architecture is even simpler than the one proposed in [8], since it uses one hidden neuron less. Although the optimal network architecture cannot be extracted from theoretical findings, it is rather empirically found that the ratio of learning parameters with respect to the size of the training data should be kept small. As stated in [17,18] the generalization error can be decomposed into an approximation error due to the number of parameters and to an estimation error due to the finite number of data available. A bound for the generalization error E is given by

$$E \leq O\left(\frac{1}{P}\right) + O\left(\left[\frac{Pn\ln(P|Dataset|) - \ln\delta}{|Dataset|}\right]^{1/2}\right) \tag{3}$$

where n is the number of input units, P is the number of hidden nodes, δ is a confidence parameter, $\delta \in (0,1)$, and $|Dataset|$ is the size of the dataset. Since in each subsequent step $|Dataset|$ increases, the bound of the generalization error E is expected to decrease, therefore the learning model provides more accurate estimations as the negotiation proceeds. Applying in (2) n=3 and $|Dataset|$ = 2 (minimum value required by the LM method), yields that the agent can initially train and use the MLP at the tenth round.

As far as complexity is concerned, storage of the Jacobian matrix ($|Dataset|$ xP), as well as computations for matrix inversion that are of order $O(P^3)$, are required at each iterative step of the LM method. The LM is considered efficient since it can be defined as a polynomial time algorithm (an algorithm that has time complexity that is bounded by a polynomial in the length of the input) [19].

B. Adaptive Session-long Learning Agents (ASLAs)

Unlike SSLA, the ASLA considers not only the series of his counterpart's past offers, but also the series of his own past offers, to formulate the subset of input features. Particularly, in order to find the optimal subset which will guide the prediction, two time series are taken into account: one resulting from past offers of the predicting agent $\{X^{0}_{(Con\to\mathrm{Pr})}, X^{2}_{(Con\to\mathrm{Pr})},...., X^{t-2}_{(Con\to\mathrm{Pr})}\}$, and one resulting from the past offers of the opponent $\{X^{1}_{(\mathrm{Pr}\to Con)}, X^{3}_{(\mathrm{Pr}\to Con)},...., X^{t-1}_{(\mathrm{Pr}\to Con)}\}$. The encoded information represents the number of previously offered values of each agent. Using a binary grammar, three bits are sufficient to encode up to seven past offers for each agent. Consequently a 6-bit length string represents the subset of input features. Since it has been proved that an MLP with one hidden layer can conduct function approximation, and since it has been widely used by existing predicting agents, the architecture of a two layered MLP is assumed, and focus is set on searching the optimal number of hidden units. In an attempt to keep the network small, three bits are used for the representation of the hidden units, resulting to a chromosome of nine bits which simultaneously evolves the subset of input features and architecture of the network (Figure 2).

The ASLA appropriately adjusts the architecture of the employed MLP by applying the following genetic algorithm.

Step 1. Randomly generate the initial population P
Step 2. Decode each individual (chromosome) into an architecture
Step 3. Evaluate individuals:
  a. Train each network with a predefined training algorithm and parameters
  b. Define the fitness of each individual according to the training result and other performance criteria, such as the complexity of the architecture
Repeat
Step 4. Select a set of promising individuals and place them in the mating pool
Step 5. Apply crossover to generate offspring individuals
Step 6. Apply mutation to perturb offspring individuals
Step 7. Replace P with the new population
Step 8. Evaluate all individuals in P (as in step 3)
Until 10 generations

$$\underbrace{010}_{\substack{number.of\\ predicting\\ agents.past\\ offers}} - \underbrace{101}_{\substack{Number.of\\ opponents'\\ past.offers}} - \underbrace{001}_{\substack{Numberof\\ hidden.neurons}}$$

Figure 2. A chromosome consisting of 9 bits is used to evolve the input subset and the number of hidden neurons of the neural network.

Every time the genetic algorithm is run, the agent selects the MLP with the lowest fitness function. He then applies the MLP to forecast his counterpart's response in a similar way to that of the SSLA. More specifically, the ASLA initially generates a random population of individuals (Step 1). Each individual is translated to the respective MLP (Step 2), which is then trained and evaluated (Step 3).

The training patterns are extracted from the current negotiation thread. If the available number of previous predicting agent's offers at decision making time *t* is m, and for opponent's offers is n, where m,n є {0,1,…,t/2} and m+n>0, the first observation is extracted at time t' is:

$$t' = \begin{cases} 2m+2, if.2m-2n-1>0 \\ 2n+2, if\ .2m-2n-1<0 \end{cases} \tag{4}$$

and the size of the available dataset at time *t* is:

$$\left|Dataset\right| = 1 + \frac{t-t'}{2} \tag{5}$$

As far as the objective (fitness) function is concerned, since $\left|Dataset\right|$ must be at least 2 to apply the LM method, the ASLA favors solutions that result to $\left|Dataset\right| \geq 2$. Furthermore, in cases where it is possible to divide the available data in three sets (training, validation and test set), the objective (fitness) function, which is minimized through the GA solver, is proportional to the Mean Squared Error (MSE) of the test set. Preference is given to solutions which result to more data patterns, in order to apply an early stopping learning method, which guarantees better generalization.

After evaluation, the most promising individuals are placed in the mating pool (Step 4), and GA operators are applied (Steps 5 and 6) to formulate the new population (Step 7). The new individuals are in turn evaluated (Step 8) and the process is repeated for 10 generations. The trained MLP that yields from the most promising individual is applied for the purpose of forecasting the counterpart's next offer.

It is important to note that implementation of ASLA advances the state of the art in the field of applying Neural Networks in negotiations to predict the counterpart's responses. It is based on an optimization technique and illustrates a pathway of finding a sub-optimal structure and subset of input features for the network. It could be used as a reference point in the development of other forecasting tools that assist negotiators. Additionally, it is a way of addressing the issue of heterogeneity of existing systems when it comes to selecting the offers of the negotiation thread which will constitute the input of the forecasting tool. In the following section SSLAs and ASLAs are compared.

V. Comparison of SSLAs and ASLAs

For the conduction of the experiments we have developed a simulator that produces negotiator objects in Java (Jdk version 1.6), which are then extended in Matlab (version 2008R) and are enhanced with learning techniques. The negotiator objects are capable of conducting bilateral multi-issued negotiations. Experiments involve the generation of different negotiation environments, with provider and consumer agents, which can perform learning tasks, and engage in negotiations following the predictive strategy discussed in [9].

To compare session-long learning agents and current state of the art agents we conducted numerous experiments, simulating different negotiation environments. The cases studied involved short negotiations, where each counterpart set a deadline of 50 steps, as well as long negotiations, where the counterparts set a deadline of 350 steps. It was observed that SSLAs reduced the average of absolute prediction error by 92.67% compared to agents that only trained the MLP at a pre-negotiation stage, in cases where data distributions changed (turbulent settings).

In this research, focus is set on the comparison of the two types of session-long learning agents. In this respect 1,359 negotiation environments, where the participants adopted various strategies, deadlines and reservation values, were simulated. Negotiations were conducted between SSLAs and non-learning agents, and between ASLAs and non-learning agents with the objective to measure the accuracy of the predictions in each case. More specifically, in each negotiation round the absolute error, defined as the difference between the prediction and the actual value, is computed. Assessment is provided through the computation of statistical information (mean, standard deviation and maximum value of the absolute errors) in each negotiation instance. The purpose of the comparison is to illustrate the deviation of the error as the agents negotiate in new settings.

Results which illustrate average and maximum values of the computed variables are summarized in Table 1. Avg Mean and Max Mean refer to the average and maximum value of the mean of absolute errors. Accordingly, Avg Std and Max Std refer to the average and maximum standard deviation observed, while Avg Max and Highest Max stand for the average and maximum of the highest error values acquired in negotiations. The ASLA is shown to be more accurate in the general case since it yields reduction of the mean of absolute errors (Avg Mean) by 38.34%, reduction of Avg Max by 44.74% and reduction of Avg Std by 38.03%. More specifically, when the ASLA deals with counterparts following time dependent (TD) strategies the same measures (Avg Mean, Avg Max and Avg Std) are reduced by 36.11%, 37.24%, and 31.52% respectively, while when it deals with counterparts following behavior dependent (BD) strategies Avg Mean, Avg Max and Avg Std are reduced by 38.45%, 45.29%, and 38.32%.

SSLAs and ASLAs can be safely used in cases where the counterpart's strategy can be expressed by continuous functions. In the scenarios described, these are the cases with TD strategies, yielding to SSLA and ASLA Avg Mean of 0.36 and 0.23, Avg Max of 6.66 and 4.18, and Avg Std of 0.92 and 0.63 respectively.

On the contrary, when opponents' behavior is sharp (as is the case of BD strategies), neural networks are less accurate. In the experiments conducted, cases with BD strategies yield to SSLA and ASLA Avg Mean of 11.91 and 7.33, Avg Max of 88.96 and 48.67, and Avg Std of 19.23 and 11.86 respectively.

ASLAs are not as fast as SSLAs and have higher storage requirements. However, they yield better results as they prove more accurate with decreased standard deviation and maximum error values.

VI. Conclusions and Future Reasearch

Current state of the art negotiating agents and support tools are enhanced with learning techniques in order to provide increased benefit to the parties they represent or support. A very promising skill is to foresee the counterpart's future moves and accordingly adapt one's decisions. The trend lies on neural networks, which have been proved efficient for various systems and domains. These models are capable of mapping input to output space, as long as appropriate data are used for training. The networks' accuracy is dependent on the training set. The problem that arises from current implementations is that (in the majority) the employed networks are trained at a pre-negotiation step. Results are impressive if data with similar underlying distributions are considered. However this is not the case in turbulent environments. In this research, the perspective of using data solely extracted from the actual negotiation thread is considered, and focus is set on the employment of very simple networks initiated without any a priori knowledge (random initial weights). A static session-long learning agent (SSLA), using a network with fixed architecture and standard input features, and an adaptive session-long learning agent (ASLA), evolving its network structure and feature subset in each negotiation round, are described and implemented.

The adaptive agent, ASLA, is empirically proved to be more accurate when dealing with opponents that adopt time and behavior dependent strategies; however it is observed that both agents yield high utility gain in cases where the counterpart's strategy is defined by continuous functions (which is the case of time dependent strategies), and do not score that high when opponents adopt behavior dependent strategies. Refinement of the predictive strategy discussed in [9] will be considered in the future, in order to also tackle opponents with smart or hybrid behaviors.

SSLAs and ASLAs have been implemented without any assumption of data distributions and for this reason they could also be applied in different types of negotiation arenas.

ASLAs have proved more efficient than SSLAs, however the trade-off is the increased time of convergence. Other efficient adaptive structures can be considered, such as Evolving Fuzzy Neural Networks (EFuNNs) or DENFIS, which are Evolving Connectionist Systems (ECoS) that continuously evolve their structure and functionality to capture the dynamics of turbulent settings [20].

TABLE I. COMPARISON OF SESSION-LONG LEARNING AGENTS

Measured Variables		Avg Mean		Max Mean		Avg Max		Highest Max		Avg Std		Max Std	
		SSLA	ASLA	SSLA	ASLA	SSLA	ASLA	SSLA	ASLA	SSLA	ASLA	SSLA	ASLA
Totals	TD	0.36	**0.23**	8.9	**4.67**	6.66	**4.18**	476.61	**127.14**	0.92	**0.63**	49.85	**15.19**
	BD	11.91	**7.33**	183.92	**52.16**	88.96	**48.67**	3050	**264.78**	19.23	**11.86**	346.99	**68.11**
Overall	TD,BD	6.13	**3.78**	183.92	**52.16**	47.81	**26.42**	3050	**264.78**	10.07	**6.24**	346.99	**68.11**

REFERENCES

[1] G. E Kersten, E. Chen, D. Neumann, R. Vahidov, and C. Weinhardt, "On comparison of mechanisms of economic and social exchanges: The times model," Proc. of Negotiation and Market Engineering, Heidelberg, Berlin: Springer, 2006.

[2] P. Faratin, C. Sierra, and N.R. Jennings, "Negotiation decision functions for autonomous agents," Int. Journal of Robotics and Autonomous Systems, vol. 24, no. (3 - 4), 1998, pp. 159-182.

[3] M. Masvoula, P. Kanellis, and D. Martakos, "A review of learning methods enhanced in strategies of negotiating agents," Proc. Int. Conf. on Enterprise Information Systems, Madeira Portugal, 2010, pp. 212-219.

[4] R. Carbonneau, G.E. Kersten, and R. Vahidov, "Predicting opponent's moves in electronic negotiations using neural networks," Expert Systems with Applications: An International Journal, vol. 34, no. 2, 2008, pp. 1266-1273.

[5] C.C. Lee and C. Ou-Yang, "A neural networks approach for forecasting the supplier's bid prices in supplier selection negotiation process," Expert Systems with Applications, vol. 36, no. 2, 2009, pp. 2961-2970.

[6] I.V. Papaioannou, I.G. Roussaki, and M.E. Anagnostou, "Comparing the performance of MLP and RBF neural networks employed by negotiating intelligent agents," Proc. IEEE/WIC/ACM international Conf. on intelligent Agent Technology, Washington, DC, 2006, pp. 602-612.

[7] I.V. Papaioannou, I.G. Roussaki, and M.E. Anagnostou, "Towards successful automated negotiations based on neural networks," Proc. 5th IEEE/ACIS Int. Conf. on Computer and Information Science, Washington DC, 2006, pp. 464-472.

[8] M. Oprea, "The use of adaptive negotiation by a shopping agent in agent-mediated electronic commerce," Proc. 3rd Int. Central and Eastern European Conf. on Multi-Agent Systems, 2003, pp. 594-605.

[9] M. Masvoula, C. Halatsis, and D. Martakos, "Predictive automated negotiators employing risk-seeking and risk-averse strategies," Proc. 12th EANN/7th AIAI Joint Conf., Sept. 2011, pp.325-334.

[10] C. Hou, "Predicting agents tactics in automated negotiation," Proc. of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology, Washington, DC, 2004, pp. 127-133.

[11] I. Roussaki, I. Papaioannou, and M. Anagnostou, "Building automated negotiation strategies enhanced by MLP and GR neural networks for opponent agent behaviour prognosis," Proc. IWANN 2007, Heidelberg, Berlin: Springer-Verlag, 2007, pp. 152-161.

[12] I. Papaioannou, I. Roussaki, and M. Anagnostou, "Detecting unsuccessful automated negotiation threads when opponents employ hybrid strategies," Proc. 4th Int. Conf. on Intelligent Computing: Advanced Intelligent Computing Theories and Applications - with Aspects of Artificial intelligence, Heidelberg, Berlin: Springer-Verlag, 2008, pp. 27-39.

[13] J. Brzostowski and R. Kowalczyk, "Modelling partner's behaviour in agent negotiation," in AI 2005: Advances in Artificial Intelligence, LNCS 3809, Berlin, Heidelberg : Springer-Verlag, 2005, pp. 653-663.

[14] J. Brzostowski and R. Kowalczyk "Adaptive negotiation with on-line prediction of opponent behaviour in agent-based negotiations," Proc. IEEE/WIC/ACM Int. Conference on intelligent Agent Technology, Washington, DC: IEEE Computer Society, 2006, pp. 263-269.

[15] J. Brzostowski and R. Kowalczyk, "Predicting partner's behaviour in agent negotiation," Proceedings of the 5th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems, New York USA: ACM, 2006, pp. 355-361.

[16] I.V. Papaioannou, "Optimizing automated negotiations with various learning techniques," Doctoral Dissertation, Athens: National Technical University of Athens, 2009.

[17] P. Niyogi and F. Girosi, "Generalization bounds for function approximation from scattered noisy data," Advances in Computat Math, vol. 10, 1999, pp. 51–80.

[18] A.R. Barron, "Approximation and estimation bounds of artificial neural networks," Machine Learning, vol.14, no. 1, 1994, pp. 115-133.

[19] O. Goldreich, Computational complexity: A conceptual perspective. New York: Cambridge University Press, 2006.

[20] N. Kasabov, Evolving connectionist systems: The knowledge engineering approach. London: Springer-Verlag, 2007.

 ISBN: 978-1-61208-273-8