

A MULTI-AGENT BASED FRAMEWORK FOR SUPPORTING LEARNING IN ADAPTIVE AUTOMATED NEGOTIATION

Rômulo Oliveira, Herman Gomes
Federal University of Campina Grande
Av. Aprígio Veloso, 882 - Campina Grande, Paraíba, Brazil

Alan Silva, Ig Bittencourt, Evandro Costa
Federal University of Alagoas
Campus A.C. Simões - BR 104 - Km 14 - Maceió, Alagoas, Brazil

Keywords: Trading agents, multi-agent architecture, cognitive models, machine learning.

Abstract: We propose a multi-agent based framework for supporting adaptive bilateral automated negotiation during buyer-seller agent interactions. In this work, these interactions are viewed as a cooperative game (from the idea of two-person game theory, nonzerosum game), where the players try to reach an agreement about a certain negotiation object that is offered by one player to another. The final agreement is assumed to be satisfactory to both parts. To achieve effectively this goal, we modelled each player as a multi-agent system with its respective environment. In doing so, we aim at providing an effective means to collect relevant information to help agents to make good decisions, that is, how to choose the “best way to play” among a set of alternatives. Then we define a mechanism to model the opponent player and other mechanisms for monitoring relevant variables from the player’s environment. Also, we maintain the context of the current game and keep the most relevant information of previous games. Additionally, we integrate all the information to be used in the refinement of the game strategies governing the multi-agent system.

1 INTRODUCTION

A recent trend within electronic commerce systems is to provide automated negotiation, that is one of the hottest research topic in Artificial Intelligence and Economy. Lately this topic is receiving more attention from the scientific community with the challenges related for providing more realistic and feasible solutions. Following this track, is proposed a multi-agent based framework for supporting adaptive bilateral automated negotiation during buyer-seller agent interactions. In this work, these interactions are viewed as a cooperative game (from the idea of two-person game theory, nonzerosum game), where the players try to reach an agreement of a certain negotiation object that is offered by one player to another. The final agreement is assumed to be satisfactory to both parts. A particular challenge for researchers in adaptive automated negotiation concerns how to effectively model the opponent and for monitoring the environment, here assumed as dynamic.

To achieve effectively the issues above, each player is modelled as a MAS with its respective environment. In doing so, we aim at providing an effective means to collect relevant information to help agents to make good decisions, that is, how to choose the “best

way to play” among a set of alternatives. Following, is used some machine learning technique, neural network, to define a mechanism to model the opponent players and other mechanisms for monitoring relevant variables from the player environment. Also, is maintained the context of the current game and keep track the most relevant information of previous games. Additionally, is integrated all the information to be used in the refinement of the game strategies governing the multi-agent system. To this end, is used a Q-learning algorithm.

2 ENVIRONMENT DESCRIPTION

In a system to negotiate in the e-commerce, the negotiator, before being seen as only one entity (Fatima et al., 2004; Narayanan and Jennings, 2005), is designed as a Multiagent System (MAS), where their members collaborate between them (Filho and Costa, 2003; Bartolini et al., 2004; Sardinha et al., 2005). Thereby, necessities tasks as analyze of proposals, learning, perception of changes in the market and others are distributed for the members in the MAS.

An essential item in a negotiation is the exchange of messages, however, two things have to be resolved:

- *Physical Communication between the agents*:: involves all the interaction process between the agents, since the communication protocol until the use of software structure. FIPA (Foundation for Intelligent Physical Agents) resolved part of this stage by means of the development of an universal standard protocol to MAS. Jade (Java Agent Development) implements FIPA's standards, resolving this item completely;
- *Negotiation Protocol*: protocol to manage the exchange of proposals in the negotiation environment. There are some works that propose solution for this stage (Filho and Costa, 2003).

What follows, are approached some fundamentals details involved in framework building, divided into three subsections: *negotiation objects*, *negotiation protocol* and *negotiator model*.

2.1 The Negotiation Object

The negotiation object is represented for a vector of attributes that assumes the values in a determinate range (Filho and Costa, 2003). Thereby, in negotiation, the negotiation object has a subset of attributes that belongs to a set of attributes pre-adjusted of the negotiation object. In others words:

Consider $A = \{a_1, a_2, a_3, \dots, a_m\}$ a set of possible attributes pre-adjusted in a negotiated process;

and also, $C = \{c_1, c_2, c_3, \dots, c_n\}$ a set of characteristics that composes a physical description of the negotiation object.

Finally, a negotiation object is defined as: $o = \{C', A'\}$, where $A' \subseteq A$ and $C' \subseteq C$. The Figure 1 illustrates this definition. All attributes $a \in A'$ and characteristics $c \in C'$ are decision variables with discreet or linear values, and its respectives domains should be predefined. During this paper, the decision variables are called as the set $(a_i \cup c_i)$ represented by the object o_i .

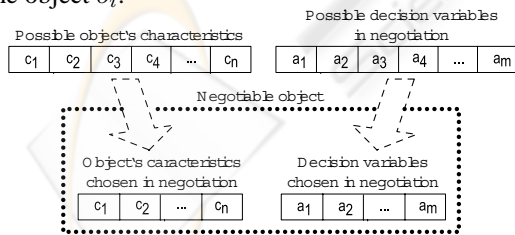


Figure 1: Composition of a negotiation object.

2.2 Negotiation Protocol

The Negotiation Protocol is responsible for organizing the actions of the entities involved in the negotiation concerning the time. According to (Filho and Costa, 2003; Debenham, 2003), are necessities that

the agents holds actions defined during the negotiation: accept a proposal; give up of the negotiation; send a proposal; suggest alternatives products; suggest correlates products.

Algorithm 1 General Algorithm of Negotiation

```

1 - The Buyer-MAS search the Seller-MAS;
2 - The negotiators define the negotiation objects with its weight and
   restrictions in the pre-negotiation;
3 - One of the negotiators send a propose;
4 - The other negotiator analyze his utility criteria and send a response
   informing if the proposal was accepted or not;
IF Proposal Rejected THEN
  IF Negotiators want continue THEN
    Go to step 3;
  ELSE
    Go to step 7;
  END IF
END IF
5 - Negotiators evaluate the general result, that is kept in a historic;
    
```

Is perceived that for each action, there is a necessity of a reasoning considering the knowledge of the domain and motivations that justify the presence of AI techniques as a possible solution.

The Negotiation Protocol is divided into three steps, showed in Algorithm 1: pre-negotiation, negotiation and general evaluation of negotiation. Each one of these steps are approached below.

2.2.1 Pre-Negotiation

As far as Rubens's work (Filho and Costa, 2003) concerned, the negotiation should be started with a pre-negotiation, where the agents should define which one of the negotiate object and which ones attributes that should be considered in the negotiation. Preferences and restrictions of the negotiation object are combined between the negotiators, as showed in Subsection 2.1.

Each attribute a_i and c_i considered in the negotiation receives influence for a weight w_i (where $\sum w_i = 1$) that represents the importance degree of the attribute in the negotiation. In this way, for instance, a negotiator can give more importance for an attribute, such as price, than the others, such as *time of delivery*. If the characteristics of the negotiated object were fixable and unnegotiable (for instance, a specific book), all the weights referents to c_i will be nulls.

The negotiator can solicits some personal datas for his opponent, with the aim the treatment negotiation in a different way, $D = d_1, d_2, d_3, \dots, d_n$. Then, the negotiator can use only the set $D' \subseteq D$.

2.2.2 Competitive Negotiation

Here occur the exchange of proposals aiming a deal between the two parts. For the successful of a safe

deal and to measure how interesting are the generated proposals, is necessary the existence of auxiliary functions (Faratin et al., 2002; Klein et al., 2003). With this in mind, each agent has two own functions, often used during the negotiation:

- *Utility Function*

This function evaluates how useful is a proposal. Considering criterions defined for each negotiator, and adjusts of each weight w_i to the attributes involved in the negotiation, the utility function U_T , for convenience, returns a value inside the range $[0, 1]$.

$$U_T(P) = \sum_{i=1}^n w_i u_{ai}(P), \text{ where } U_T(P) \in [0, 1] \quad (1)$$

- *Similarity Function*

Is a function based in fuzzy similarity that calculates the similarity degree between two proposals, returning values in the range $[0, 1]$. Two proposals are similar when their contents (attributes and values) are similar. The similarity degree is a physical measure of semblances and is not influenced for the utility of the proposals. Then, two proposals can have a high similarity degree and hold different utilities and vice versa.

Consider the following proposals, each one of them is based and composed in a unique object.

$$R = \{r_{a1}, r_{a2}, \dots, r_{an}, r_{c1}, r_{c2}, \dots, r_{cm}\}$$

$$S = \{s_{a1}, s_{a2}, \dots, s_{an}, s_{c1}, s_{c2}, \dots, s_{cm}\}$$

The function that calculates the similarity between two attributes r_i, s_i is given for: $sim(r_i, s_i) = 1/|h_i(r_i) - h_i(s_i)| \in [0, 1]$, where $h_i()$ is the similarity function to the i^{th} attribute that returns values in the range $[0, 1]$.

Now, the function $SIM()$ that calculates the similarity between two proposals is defined for:

$$SIM(R, S) = \sum (w_i * sim(r_i, s_i)) \in [0, 1] \quad (2)$$

When a proposal P_1 is received, the negotiator agent responsible for receiving the proposal can perform any attitude mentioned in the begin of this section. Besides that, to send another proposal P_2 , the negotiator should have how measure the proposal received, and then:

- To generate proposals P_{2i} with utility higher than the P_1 utility;
- To choose the most similar for P_1 between the proposals P_{2i} generated;
- Send the proposal;

This process occurs repeatedly until that the negotiation finish due to a negative deal (Zeng and Sycara, 1997). A deal occurs in the negotiation process when a proposal is accepted for both parts (Jennings et al.,

2001; Faratin et al., 2002). There are facts that can influence in change of behavior and attitudes of the negotiators agents: the time, data of the negotiation environment, analyze of opponent profile (Faratin et al., 2002). An attitude can be viewed as an action of the agent that influences the trajectory of the negotiation (Zhang et al., 2004). Are cited as examples of attitudes: send ultimates, finish the negotiation, change the negotiation strategy, send alternative products, show correlates products, others.

Grouping negotiators profiles for their characteristics and associate for this profiles a set of adequate weight w_i , the chances of the proposals been accepted by opponent negotiator improve considerable. Retaking the exposition of the prior paragraph, we will try to find a set of weight $w'_i \in B'$ that best represent, or assimilate, with the set of weight $w_i \in B$.

2.2.3 General Evaluation of Negotiation

Here, an evaluation process classifies how good was the negotiation (general outcomes) and also the set of techniques used to reach these outcomes. Based on the data acquired during the various negotiations, are used some AI techniques to tuning the strategy and to minimizing the time spent in the process (Wong et al., 2000); data mining to propose correlate products; Q-learning to reinforce the set of ideal weight to the elaboration of good proposals. In the next section, is showed that this evaluation will be kept as historical, and is provided to the agent negotiator.

3 MAS ARCHITECTURE

This section shows how each agent behavior inside a e-commerce scenery following the protocol specifications approached in the prior section. Then agent descriptions based in Figure2 is explained below:

- Mediator Agent: is responsible for communicating with others agents, managing decision of the collaborator agents and executing the negotiation protocol specified in Subsection 2.2.
- Manager Negotiator Agent: this agent is responsible for keeping the status of each negotiation that has been performed. Also, the agent receives informations about existent profile by *Profile Cataloguer Agent* (flows “b” in the Figure 2), jointly with the credibility level.
- Manager Proposal Agent: This agent is responsible for generating a good proposal to a negotiation that has been performed. This agent searches the *Manager Strategies Agent* the best strategy to be used to send a proposal to MAS partner in question (flows “g” in the Figure 2).

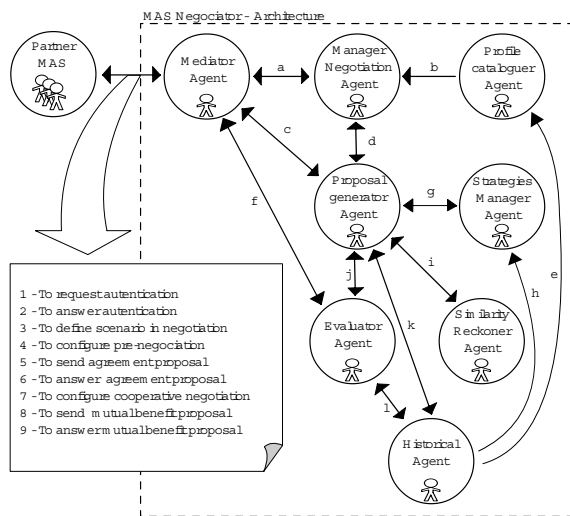


Figure 2: Negotiator MAS - Architecture.

- **Evaluator Agent:** This agent is responsible for evaluating how to determine if the proposal is satisfactory in a certain moment (flows “f” and “j” in the Figure 2).
- **Historical Agent:** This agent is responsible for maintaining the historical of each action performed by the agent. It always notifies the agents *Manager Strategies* and *Profile Cataloguer* (respectively, the flows “h” and “e” in the Figure 2).
- **Profile Cataloguer Agent:** This agent is able to identify the partners profiles under various view, and, at the same time, inform the credibility level that can be mapped in a determined client.
- **Similarity Reckoner Agent:** This agent is responsible for evaluating how two proposals are similar, according to certain criterions are evaluated and how each ones of these criterions are relevant to this evaluation (flows “i” in the Figure 2).
- **Manager Strategies Agent:** This agent is responsible for inferring a strategy that is probably more effectiveness in the negotiation.

4 LEARNING TECHNIQUES

One of the advantages in an multiagent approach refers the use of artificial intelligence techniques in a independent view, in each agents, and then optimize all the system. In a negotiation process, learning is present in the elaboration of the proposals, client classification, negotiation management and in any other process associated in an agent of section 3.

The profile grouping comes to help the identification of the MAS Negotiator's, choosing a strategy. The *Profile Cataloger Agent* is responsible for profiles

grouping. This agent uses a set of negotiation data sent to learn about these classified groups. The profile identification is made through the use of a Neural Network SOM technique due its high generalization power. As is used in each new business domain, the network should be trained, using the n first occurred negotiation to train the network. Besides this, the agent should train the network periodically, making it learn with new cases. After trained, the network is able to identify quickly a new set of inputs between the n generated profiles. The agent goals made the classification based on personal client datas associated with features of the object negotiated. The sets of weight w_i for each proposal can be used in the future, however, makes necessary a treatment to the identification of standard variation in each negotiation. Below, following the Learning techniques explored in the Profile Cataloger Agent.

Algorithm 2 Neural Network Algorithm

- 1 - Obtain the first n negotiation in the historic;
- 2 - Define the quantity q of learning profiles;
- 3 - Performing a Network SOM with q output neurons and use the set of negotiation n in the historic for treatment.
- 4 - Receive a new case and classify it as one profile q ;
- 5 - Use a set of strategies associated to the profile;
- 6 - In the end of the negotiation, insert a new case in the historic;
- 7 - IF The number of new cases in historic justifying treatment
THEN Go to Step 3
ELSE Go to Step 4;

The profile identification algorithm defines the profiles grouping, thereby, before a new negotiation been started, is defined the kind of profile that the negotiation is embedded. In negotiation, is used the Q-Learning approach objectifying improve the negotiation process through the selection of better strategies.

The goal of the agent in a Reinforcement Learning problem is to learn an action policy that maximizes the expected long term sum of values of the reinforcement signal, from any starting state (Bianchi et al., 2004). The Reinforcement Learning Problem can be resolved as a discrete time, finite state, finite action Markov Decision Process (MDP). In the present work, the problem is defined as a MDP solution.

The chosen of better strategies can modelled by a 4-tuple (S, A, T, R) , where: S is finite set of states; A is finite set of actions; $T : S \times A$: is state transition function represented for the probability value, signaling the better strategy to be chosen; $R : S \times A$: is a reward function.

The Reinforcement Learning algorithm, concerning Markov Decision Process, the estimated reward function is defined as

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

where $\hat{Q}(s, a)$ is a recursive algorithm used to pre-

dict the best action; $\alpha = 1/(1 + visits(s, a))$, where $visits(s, a)$ is a total number of times this state-action pair has been visited up to and including the current interaction; r is a reward received; γ is a discount factor ($0 \leq \gamma < 1$).

We use in the e-commerce environment a proposal approached in (Bianchi et al., 2004), the implements an algorithm that is used in the action choice rule, which defines which action must be performed when the agent is in state s_t . The heuristic function included

$$\pi(s_t) = \begin{cases} \text{argmax}_{a_t} [\hat{Q}(s_t, a_t) + \xi H_t(s_t, a_t)] & \text{if } q \leq p, \\ a_{random} & \text{otherwise,} \end{cases}$$

- $H : S \times A \rightarrow R$ is the heuristic function;
- ϵ : is a real variable used to weight the influence of the heuristic function;
- q : is a random uniform probability density mapped in $[0, 1]$ and $p(0 \leq p \leq 1)$ is the parameter which defines the exploration / exploitation balance;
- a_{random} is a random action selected among the possible actions in state s_t ;

Then, the value heuristic $H_t(s_t, a_t)$ that can be defined as:

$$H(s_t, a_t) = \begin{cases} \max_a \hat{Q}(s_t, a) - \hat{Q}(s_t, a_t) + \eta \text{ if } a_t = \pi^H(s_t), \\ 0 & \text{if otherwise,} \end{cases}$$

Mapping the reinforcement learning algorithm to e-commerce domain, the solution process used was through MDP. Intuitively, a process is Markovian if and only if the state transitions depend only on the current state of the system and are independent of all preceding states (Narayanan and Jennings, 2005). Formally, in Markov Process, the conditional probability density function, P , is defined as:

$$P(x_n | x_1, x_2, \dots, x_{n-1}) = P(x_n | x_{n-1})$$

The 4-tuple in e-commerce environment is defined as:

- S is set of pair of proposal and counter-proposal;
- A : finite set of strategies;
- $T : S \times A \rightarrow \Pi(s)$: state transition function represented for the probability value, signaling the better strategy to be chosen;
- $R : S \times A$: is described as a utility value, defined for the similarity of the attributes, mapped as a reward function.

5 EXPERIMENTAL ANALYSIS

Were realized learning negotiation experiments, due this was used a scenery of vehicles. In the experiments, only the seller has learning characteristics. Were used as negotiation attributes the product price, the delivery time and the payment time ($A' = Price$

Algorithm 3 The Heuristics Algorithm

Initialize $Q(s, a)$

Repeat:

 Visit the s state

 Select a strategy using the choice rule

 Receive the reinforcement $r(s, a)$ and observe the next state s' .

 Update the values of $H_t(s, a)$.

 Update the values of $Q_t(s, a)$ according to:

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

 Update the $s \leftarrow s'$ state

Until some stop criteria is reached,

where $s = s_t, s' = s_{t+1}, a = a_t, a' = a_{t+1}$

$= [\$17000, \$90000]; Delivery_Time = \{1 \text{ day}, 2 \text{ days}, 3 \text{ days}, \dots\}; Payment_Time = \{1 \text{ month}, 2 \text{ months}, 3 \text{ month}, \dots\}$).

As characteristics C' , were defined the follow attributes: potency (hp), consumption, luggage space and quantity of doors ($Potency = [67hp, 400hp]; Consumption = [8Km/liter, 160km/liter]; Luggage_Space = [180 \text{ liters}, 300 \text{ liters}]; Quantity_of_doors = \{3 \text{ doors}, 5 \text{ doors}\}$).

The set D' , viewed in subsection 2.2.1, is represented for the attributes name, civil state, gender, finance, profession, age ($Civil_State = \{Single, Married, widower, Divorced\}; Gender = \{Male, Female\}; Finance = [\$6000, \$90000]; Profession = \{Bricklayer, Programmer, Teacher, Journalist, Engineering, Architect, Doctor, Lawyer, Judge, Politician\}; Age = [18, 80]$).

Given this domain specification, were simulated, firstly, 1000 negotiation. Then, were created 1000 clients with different characteristics, but coherent with their preferences. For instance, a client with characteristics: $D' = \{Gender=Male, Civil_State=single, Age=35, Finance=\$60000, Profissao=Lawyer\}$, his preferences are adjusted to: $Weight = \{Price = 0.6, Delivery_Time = 0.05, Payment_Time = 0.05, Potency = 0.1, Consumption = 0.1, Luggage_Space = 0.05, Quantity_of_Luggage = 0.05\}$.

The outcomes of this first stage of the experiment can be viewed in the Figure 3. The average time, spent to the seller in a negotiation was 432ms.

Using the techniques showed in Section 4, and using the historical of 1000 negotiations of the first experiment, a second experiment with more 1000 clients created randomly was realized. The outcomes is represented in the Figure 4. In this experiment, the seller knew decrease the average of interactions with their clients and, consequently, decrease the average time per negotiation to 207ms.

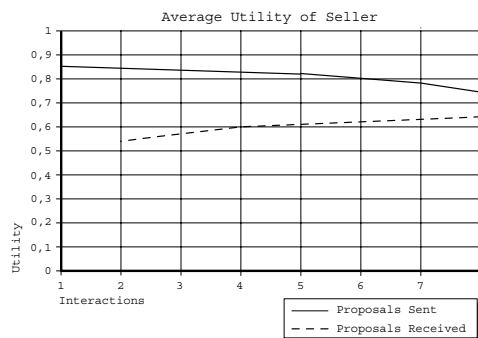


Figure 3: Before, without machine learning.

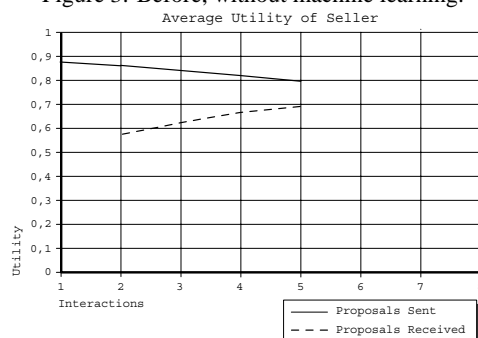


Figure 4: After, with machine learning.

6 FINAL REMARKS

This paper has presented a novel approach by proposing a software framework for supporting adaptive bilateral negotiation. The major contribution of this paper is the proposed decision making apparatus. To model opponent players, an algorithm that uses self-organizing maps (neural networks) and a Q-Learning algorithm used basically to predicting the next important actions were proposed. Therefore, some improvements over current existing approaches include the combination of neural networks and Q-Learning, used in an appropriated software framework.

For future work, is aimed to develop new components that implement other artificial intelligence techniques in order to be compared with the existing ones. For example, case-based learning to be compared with neural networks with regard to players modelling.

REFERENCES

- Bartolini, C., Preist, C., and Jennings, N. R. (2004). A Software Framework for Automated Negotiation. In Choren, R., Garcia, A., Lucena, C., and Romanovsky, A., editors, *Software Engineering for Large-Scale Multi-Agent Systems*, volume 3390 of *Lecture Notes in Computer Science*, pages 213–235. Springer-Verlag.
- Bianchi, R. A. C., Ribeiro, C. H. C., and Costa, A. H. R. (2004). Heuristically accelerated q-learning: A new approach to speed up reinforcement learning. In Bazzan, A. L. C. and Labidi, S., editors, *17th Brazilian Symposium on Artificial Intelligence*, volume 3171 of *Lecture Notes in Computer Science*, pages 245–254. Springer.
- Debenham, J. K. (2003). An eNegotiation Framework. In *Australian Conference on Artificial Intelligence*, pages 833–846.
- Faratin, P., Sierra, C., and Jennings, N. R. (2002). Using Similarity Criteria to Make Issue Trade-offs in Automated Negotiations. *Artificial Intelligence*, 142(2):205–237.
- Fatima, S. S., Wooldridge, M., and Jennings, N. R. (2004). An Agenda Based Framework for Multi-issues Negotiation. *Artificial Intelligence*, 152(1):1–45.
- Filho, R. R. G. N. and Costa, E. B. (2003). A Decision-Making Model to Support Negotiation in Electronic Commerce. In *3rd International Interdisciplinary Conference on Electronic Commerce (ECOM-03)*, pages 16–18, Gdansk, Poland.
- Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M., and Sierra, C. (2001). Automated Negotiation: Prospects Methods and Challenges. *Group Decision and Negotiation*, 10(2):199–215.
- Klein, M., Faratin, P., Sayama, H., and Bar-Yam, Y. (2003). Protocols for Negotiating Complex Contracts. *IEEE Intelligent Systems*, 18(6):32–38.
- Narayanan, V. and Jennings, N. R. (2005). An adaptive bilateral negotiation model for e-commerce settings. In *Seventh IEEE International Conference on E-Commerce Technology*.
- Sardinha, J. A. R. P., Milidiú, R. L., Paranhos, P. M., Cunha, P. M., and de Lucena, C. J. P. (2005). An Agent Based Architecture for Highly Competitive Electronic Markets. In Russell, I. and Markov, Z., editors, *FLAIRS Conference*, pages 326–332. AAAI Press.
- Wong, W. Y., Zhang, D. M., and Kara-Ali, M. (2000). Negotiating With Experience. In *CSIRO Mathematical and Information Sciences*.
- Zeng, D. and Sycara, K. (1997). Benefits of Learning in Negotiation. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI-97/IAAI-97)*, pages 36–42, Menlo Park. AAAI Press.
- Zhang, S., Ye, S., Makedon, F., and Ford, J. (2004). A Hybrid Negotiation Strategy Mechanism in an Automated Negotiation System. In Breese, J. S., Feigenbaum, J., and Seltzer, M. I., editors, *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 256–257, New York, NY, USA. ACM Press.