



Universidad
de Alcalá

BIBLIOTECA

Document downloaded from the institutional repository of the University of Alcalá: <http://dspace.uah.es/dspace/>

This is a postprint version of the following published document:

Lopez-Carmona, M. A., Marsa-Maestre, I., Velasco, J. R., Hoz, E. de la, Alarcos, B., 2010, "Strategies for Offer Generation and Relaxation in Fuzzy Constraint-Based Negotiation Models", Multiagent and Grid Systems, Vol. 6(5-6), p.503-525.

Available at <http://dx.doi.org/10.3233/MGS-2010-0163>

Copyright 2010 IOS Press



(Article begins on next page)

This work is licensed under a
Creative Commons Attribution-NonCommercial-NoDerivatives
4.0 International License.

Strategies for Offer Generation and Relaxation in Fuzzy Constraint-Based Negotiation Models

Miguel A. Lopez-Carmona*, Ivan Marsa-Maestre,
Juan R. Velasco, Enrique de la Hoz and Bernardo Alarcos

July 19, 2010

Department of Computer Engineering, Escuela Politecnica Superior,
Universidad de Alcala, Alcala de Henares (Madrid), 28871, Spain

Abstract

Fuzzy constraints have been used in several automated multi-attribute negotiation models. It is recognized that fuzzy constraints represent an efficient way of capturing requirements and preferences, and an useful mechanism for representing trade-offs. Most approaches are mainly focused on using constraints as a framework for describing preferences, and few works focus on using them as an element in the communication process itself in order to elaborate efficient negotiation strategies. This paper proposes and evaluates a set of strategies for offer generation and for the construction of offer relaxation requests in constraint-based negotiation models. To deploy the strategies, a fuzzy constraint based model for non-mediated bilateral automated purchase negotiations has been used. Fuzzy constraints are used both to represent preferences and to express offers. A set of locutions and decision mechanisms which trigger them are fully specified, where each agent may decide its degree of cooperation and its degree of expressiveness. Expressiveness is based on the propagation of constraints and relaxation requests. The paper analyzes which combination of different agents' attitudes allow to improve the negotiation processes. Experimental evaluation confirms the advantage of an expressive approach based on the propagation of constraints. In addition, this paper studies how applying a clustering algorithm to the seller's catalogue of products instead of managing single products may contribute to an improvement in the duration of the negotiation dialogues, and to a significant improvement on the utility of the deals that are achieved. These results support the usefulness of the proposed negotiation strategies.

*Corresponding author. Tel.: +34 91 8856673; Fax: +34 91 8856641 e-mail: miguelangel.lopez@uah.es

Keywords: Multi-attribute negotiation, automated purchase negotiation, fuzzy constraints, multi-agent systems

1 Introduction

An automated bilateral negotiation may be seen as an automated interaction between two agents with the goal of reaching an agreement over a given range of issues or attributes. Most research in automated negotiation to date has focused on the competitive aspect [21]. On the other hand, work by dispute resolution theorists in the social sciences has also focussed substantially on how to achieve negotiated agreements that are of a high value to all parties [4]. This approach is known as *integrative* or *interest-based negotiation*, and it has been recognised as the more successful approach to the negotiation problem. Scenarios where such approach may arise are: business process management involving agents within the same organization, e-commerce negotiations where the seller is interested in having a satisfied buyer (e.g. long-term commercial relationships), or e-commerce scenarios where risk averse agents avoid the conflict in the negotiation processes. Different techniques may be used to implement an integrative solution. For example, in multi-issue scenarios, issue trade-offs are used to find win-win solutions. In order to implement issue trade-off mechanisms, fuzzy constraints have been used in several multi-attribute negotiation models, specially in e-commerce [9, 16, 12]. Fuzzy constraints represent an efficient way of capturing requirements, being capable of representing trade-offs between the different issue values. Basically, a fuzzy constraint maps different ranges of issue values into different utility levels. Most fuzzy constraint based negotiation models have focused on using fuzzy constraints as a mean to capture and represent agent preferences [1], or to implement reasoning mechanisms based on fuzzy logic [12]. However, constraints can also be used to express offers. When expressing offers with constraints, a subspace from the solution domain can be explored in a given exchange, and then the search for agreements can be more efficient.

In [16], Xudong Luo et al. present in one of the most relevant works in the field a fuzzy constraint based model for bilateral multi-issue negotiations. In this model, two agents, a buyer and a seller, are able to automatically negotiate the purchase of a product. A buyer agent expresses offers by means of hard constraints extracted from a set of fuzzy constraints which represent her preferences. The seller agent simply rejects or accepts the offer, but never acts strategically rejecting acceptable offers waiting for future gains. The rejection by the seller agent of a buyer's offer implies that the buyer agent relaxes one of the constraints to prepare a new proposal (the relaxation is assumed to minimize

the loss of utility). As it can be seen, the negotiation protocol is simple, with a very limited agents' expressive capability, where the agents are assumed to be risk averse and strategic behaviour is not considered. The main contributions of their work are in the development of mechanisms and operators used to assess the acceptability of the offers, and in the inclusion of rewards in the negotiation process. However, in real negotiation environments the strategic aspects play an important role, and these are not taken into account in this work. In real life, some buyers or sellers could be risk seeking or risk neutral, and so, it is important to consider the strategic perspective in the negotiation model. In addition, it would be desirable to enhance the expressive capabilities of the agents in order to improve the efficiency of the negotiation processes.

This paper proposes and evaluates a set of strategies for offer generation and for the construction of offer relaxation requests, which operates in a fuzzy constraint-based negotiation framework. Taking as a basis the Luo's et al. negotiation model, we extend their negotiation framework by enhancing the expressiveness of the agents and by considering the strategic behaviour of the agents. More specifically, we propose an expressive communication model which lets a buyer agent to value the degree of importance that each submitted constraint has, and lets a seller agent to inform which is the preference for a specific constraint to be relaxed. The proposed negotiation framework is formalized as a set of locutions and a set of decision mechanisms which trigger them. The aim of this approach is to provide the basis for constructing integrative negotiation mechanisms, where agents with different strategic objectives can be considered, and the negotiation processes can be made more efficient. In addition, this paper analyzes which combination of agents' attitudes allow to improve the negotiation processes. Finally, we study how applying a clustering algorithm to the seller's catalogue of products instead of managing single products may contribute to an improvement in the duration of the negotiation dialogues, and to a significant improvement on the utility of the deals that are achieved. Experimental results show that the proposal outperforms existing approaches.

The rest of the paper is organized as follows. Section 2 presents the negotiation model, which consists of a description of the *agent preference model*, the *dialogue model*, the *decision mechanisms*, and the *transition rules* that connect the dialogue model to the decision mechanisms. Next section presents the dynamics of the *negotiation process* and the *clustering algorithm*. Section 4 analyzes the agent's strategies under the model presented and selects the valid joint strategies, and Section 5 presents the experimental scenario and the results obtained. Finally, Section 6 compares the model with the existing work, and last section summarizes the conclusions and sheds light on some future research.

2 A negotiation model based on fuzzy constraints

We have used a *formal dialogue game* to structure the negotiation framework. The framework of formal dialogue games is increasingly used as a base for structuring the interactions of agent communication protocols [19], adopted from the theory of argumentation field. Formal dialogue games are those in which two or more players pronounce or transmit locutions in accordance with certain predetermined rules. In the proposed negotiation model all dialogues are confined to two agents, buyer and seller, so that the dialogues are exclusively bilateral. The model consist of an *agent preference model*, a *set of locutions*, a *set of decision mechanisms*, and a *set of transition rules* which links the locutions to the decision mechanisms. It is worth noting that the only significant similarity with the Luo's et al. work is in the definition of the agent preference model. However, in the proposed preference model several new concepts are introduced, like the purchase requirement valuation, the relaxation requirement, and the negotiation profiles which define the strategic behaviour of the agents. The notation used in the description of the agent preference model is the same that the one used in the Luo's et al. work. For clarity purposes, in Appendix A, an UML state diagram which summarizes the negotiation protocol is shown.

2.1 Agent preference model

Buyer agent's preferences over the attributes of a product are described by means of a *fuzzy constraint satisfaction problem (FCSP)*, which is a 3-tuple (X, D, C^f) where $X = \{x_i | i = 1, \dots, n\}$ is a finite set of attributes or issues, $D = \{d_i | i = 1, \dots, n\}$ is the set of finite domains of the attributes, and $C^f = \{R_j^f | j = 1, \dots, m\}$ is a set of m fuzzy constraints over the attributes. A fuzzy constraint corresponds to the membership function of a fuzzy set, and the function that numerically indicates how well a given constraint is satisfied is the *satisfaction degree function* $\mu_{R_j^f} : X \rightarrow [0, 1]$, where 1 indicates completely satisfied and 0 indicates not satisfied at all. Given the *cut level* $\sigma \in [0, 1]$, the induced crisp constraint of a fuzzy constraint R^f is defined as R^c . It means that if R^c is satisfied, the satisfaction degree for the corresponding fuzzy constraint will be at least σ . Therefore, the *overall satisfaction degree (osd)* of a given solution $x' = (x'_1, \dots, x'_n)$ is:

$$\alpha(x') = \min\{\mu_{R^f}(x') | R^f \in C^f\}$$

On the other hand, a **seller agent** owns a private catalogue of products $S = \{s_k | s_k = (p_k, u_k)\}$, where p_k is the vector of attributes and u_k is the profit the seller agent obtains if the product is sold. We assume that the profit u_k may

depend not only on the negotiated attributes but also on non-negotiated ones (stock period for instance).

Let A_b and A_s represent a buyer and a seller agent respectively. A **negotiation process** is a finite sequence of alternate proposals from one agent to the other. During the negotiation stage, A_b utters **purchase requirements**:

$$\pi = \bigcap \left\{ R_j^{c(\sigma_j)} \mid j \in [1, m] \right\},$$

where $R_j^{c(\sigma_j)}$ is a crisp constraint induced from R_j^f at a cut level σ_j . Therefore, a purchase requirement is a purchase proposal that is formed by a set of crisp constraints induced from the set of fuzzy constraints that describes the buyer's preferences regarding the attributes of the products. Each crisp constraint in the purchase requirement can be induced at a different cut level.

Complementing the *osd* definition, the **potential or expected overall satisfaction degree (posd)** is the *osd* that a buyer agent would get if the corresponding purchase requirement is satisfied, and it is defined as:

$$\alpha^\pi = \min\{\sigma_i \mid i = 1, \dots, m\}$$

In addition, a buyer agent may add to the purchase requirement a **purchase requirement valuation**:

$$v = \{v_j \mid j = 1, \dots, m; v_j \in [0, 1]\},$$

where v_j describes the degree of importance that the constraint $R_j^{c(\sigma_j)}$ has for the buyer agent.

A seller agent may respond in three different ways: *rejecting the proposal*, *offering a product* that satisfies the purchase requirement, or *suggesting the relaxation* of the purchase requirement. So, a **relaxation request** is defined as a vector:

$$\rho = \{r_j \mid j = 1, \dots, m; r_j \in [0, 1]\},$$

where r_j represents the preference for constraint $R_j^{c(\sigma_j)}$ to be relaxed.

The negotiation process and the agreements achieved will mainly vary depending on the strategies followed by the agents when generating purchase requirements and when requesting its relaxation. All these aspects are covered modeling the **agents' attitudes**. Agents' attitudes are related to the agents' strategic behaviour in the negotiation process, where strategic behaviour is described in terms of expressiveness and receptiveness. A negotiation profile $Pr_s = \{\psi, \beta\}$ describes the **seller agent's attitude**: $\psi \in \{0, 1\}$ controls whether seller agent uses relaxation requests, and $\beta \in \{0, 0.5, 1\}$ modulates its

attitude regarding a received purchase requirement. Finally, $Pr_b = \{\eta, \xi\}$ describes the *buyer agent's attitude*: $\eta \in \{0, 1\}$ defines whether buyer agent attends relaxation requests, and $\xi \in \{0, 1\}$ controls whether purchase requirement valuations will be used.

2.2 Negotiation dialogue

The dialogue is structured in accordance to a set of locutions grouped in four stages, **open dialogue** (L1-2), **negotiate** (L3-8), **confirm** (L9-10) and **close dialogue** (L11):

L1: open_dialogue(A_b, A_s, θ) A_b suggests the opening of a purchase dialogue to a seller participant A_s on product category θ . A_s wishing to participate must respond with *enter_dialogue*(.).

L2: enter_dialogue(A_s, A_b, θ) A_s indicates a willingness to join a purchase dialogue with participant A_b . Within the dialogue, a participant A_b must have uttered the locution *open_dialogue*(.).

L3: willing_to_sell(A_s, A_b, p_j) A_s indicates a willingness to sell a product. A_b must have uttered a *desire_to_buy*(.) or a *prefer_to_buy*(.) locution.

L4: desire_to_buy(A_b, A_s, π) A_b requests to purchase a product that satisfies the purchase requirement π .

L5: prefer_to_sell(A_s, A_b, π, ρ) A_s requests to relax the purchase requirement π , and expresses which constraints are preferred to be relaxed by means of the relaxation request ρ .

L6: prefer_to_buy(A_b, A_s, π, v) A_b requests to purchase a product which satisfies the purchase requirement π , and expresses its preferences for the different constraints by means of the purchase requirement valuation v .

L7: refuse_to_buy(A_b, A_s, p_j) A_b expresses a refusal to purchase a product. This locution cannot be uttered following a valid utterance of *agree_to_buy*(.).

L8: refuse_to_sell($A_s, A_b, p_j | \pi$) A_s expresses a refusal to sell a product, or it expresses a refusal to sell products that satisfy the purchase requirement π . This locution cannot be uttered following a valid utterance of *agree_to_sell*(.).

L9: agree_to_buy(A_b, A_s, p_j) A_b commits to buy a product. A locution of the form *willing_to_sell*(.) must have been uttered.

L10: agree_to_sell(A_s, A_b, p_j) A_s commits to sell a product. A locution of the form *agree_to_buy*(.) must have been uttered.

L11: withdraw_dialogue(A_x, A_y, θ) For A_x and A_y participants with different roles (i.e. sellers and buyers), A_x announces agent A_y the withdrawal from the dialogue.

2.3 Decision mechanisms

Syntactic rules are not enough to ensure that the dialogues are generated automatically. It is essential to equip each participant with mechanisms (*semantic decision mechanisms*) that allow it to invoke the correct locution at the right time, as a response to previous locutions or in anticipation of future ones. The mechanisms are grouped together depending on the role of the participant, Buyer (B) or Seller (S).

2.3.1 Buyer agent mechanisms

B1: Recognize Need allows to recognize the need to *acquire a product*. It may be based on a explicit initiative of the user, or in an automated response based on external stimulus.

Outputs: *wait*, *have_need*(θ), *have_no_need*(θ), where (θ) defines a *product category*.

B2: Generate Purchase Requirement defines two possible outputs, one that states that it is impossible to generate a purchase requirement and another one that specifies a requirement.

Outputs: *empty_set*(\emptyset), π

B3: Generate Purchase Requirement Valuation generates a *purchase requirement valuation* v when $\xi = 1$ (i.e. when buyer is expressive). Otherwise, the mechanism returns an *empty_set*.

Outputs: *empty_set*(\emptyset), v

B4: Consider Offers works *accepting* or *rejecting* a *sale offer* proposed by the seller agent, or *detecting* the need to *generate a new purchase requirement*. Given π^t sent at instant t , a buyer agent accepts a sale offer p_j when $\alpha(p_j) \geq \alpha^{\pi^{t+1}}$. The acceptance of an offer opens the offer confirmation stage of the dialogue. If a sale offer is not accepted and does not match π^t , the mechanism returns *reject_offer*(p_j). Otherwise, if the sale offer is not accepted, but it matches the constraints in π^t , the mechanism returns *gen._p._req*(p_j), indicating that a new purchase requirement must be generated. This last case may appear if π^t does not contain information for all the constraints in C^f .

Outputs: *accept_offer*(p_j), *reject_offer*(p_j), *gen._p._req*(p_j)

B5: Consider Withdrawal decides if it should *terminate* a dialogue with the seller agent. Outputs: *wait*, *withdraw*(θ)

2.3.2 Seller agent mechanisms

S1: Recognize Category assures that the seller agent has *available products* of the *category* (θ) in its catalogue.

Outputs: *wait*, *wish_to_enter*(θ), *wish_not_to_enter*(θ)

S2: Assess Purchase Requirement *evaluates* a received purchase requirement π^t . If exists, the mechanism returns the *sale_offer*(p_j) which matches π^t and maximizes the seller’s utility (risk aversion is assumed). Otherwise, it returns \emptyset or π^t depending on the seller’s expressive profile ($\psi = 0$ and $\psi = 1$ respectively). The π^t output indicates that another mechanism should generate a relaxation request based on the information contained in π^t .

Outputs: *sale_offer*(p_j), *empty_set* \emptyset , *purchase_requirement*(π^t)

S3: Generate Potential Sale-Offers *evaluates* which products in the catalogue can be considered as *good sale offers*.

Outputs: S_p , set of products considered as good offers.

S4: Generate Relaxation Request *generates* a relaxation request ρ . Given the set of potential sale offers S_p , it generates the relaxation request with the aim of leading the buyer agent towards the products contained in S_p .

Outputs: ρ

S5: Accept or Reject Offer decides whether an offer to purchase a product should be *accepted*. The mechanism returns *accept*(p_j) when p_j exists, and *reject*(p_j) otherwise.

Outputs: *accept*(p_j), *reject*(p_j)

S6: Consider Withdrawal decides whether a dialogue with a buyer agent should *finish*.

Outputs: *wait*, *withdraw*(θ)

Given the set of locutions and the corresponding internal decision mechanisms, the next stage is to link these elements to finally shape the complete negotiation framework.

2.4 Operational semantics

Operational semantics in a dialogue game indicates how the state of the dialogue changes after locutions have been sent. The locutions sent throughout the course of the dialogue generate transitions between the different states, so that the sent locutions are inputs of one or more decision mechanisms, which in turn generate new outputs in the form of locutions. Therefore, the operational semantics is a formalization of the connection between the locutions available in the dialogue model and the defined decision mechanisms. To express the operational semantics, the tuple $\langle A_x, K, s \rangle$ expresses that the decision mechanism K of agent A_x generates an output s . When the transitions are between the mechanisms of different agents, they are defined by the locutions that are sent, and when they are between the mechanisms of the same agent, they are defined without locutions. In the first case, an arrow and the denomination of the pertinent locution indicates the transition. In the second case only an arrow

appears. In the following, the set of transition rules is presented, and a brief description is given for the first four rules:

TR1 $\langle A_b, B1, have_need(\theta) \rangle \xrightarrow{L1} \langle A_s, S1, . \rangle$ indicates that a buyer agent that wishes to acquire a product from category θ , is trying to start a purchase negotiation dialogue using the locution $L1: open_dialogue(.)$. Said locution activates the mechanism $S1: Recognize Category$ of the seller agent with which it wants to establish the dialogue.

TR2 $\langle A_b, B1, have_no_need(\theta) \rangle \rightarrow \langle A_b, B1, wait \rangle$ indicates that a buyer agent that does not wish to acquire a product from category θ , will not start a purchase negotiation dialogue and will review the situation later on.

TR3 $\langle A_s, S1, wish_not_to_enter(\theta) \rangle \rightarrow \langle A_s, S1, wait \rangle$ states that a seller agent that does not wish to start a trading dialogue with a buyer agent will review the situation later.

TR4 $\langle A_s, S1, wish_to_enter(\theta) \rangle \xrightarrow{L2} \langle A_b, B2, . \rangle$ states that seller agent that wishes to participate in a purchase negotiation dialogue will do so by sending the locution $L2: enter_dialogue(.)$. This transmission makes the buyer agent to execute mechanism $B2: Generate Purchase Requirement$ with the objective of generating the first purchase requirement.

TR5 $\langle A_b, B2, \emptyset \rangle \rightarrow \langle A_b, B5, . \rangle$

TR6 $\langle A_b, B5, withdraw(\theta) \rangle \xrightarrow{L11} \langle A_s, S6, . \rangle$

TR7 $\langle A_s, S6, withdraw(\theta) \rangle \xrightarrow{L11} \langle A_b, B5, . \rangle$

TR8 $\langle A_b, B2, \pi \rangle \rightarrow \langle A_b, B3, . \rangle$

TR9 $\langle A_b, B3, \emptyset \rangle \xrightarrow{L4} \langle A_s, S2, . \rangle$

TR10 $\langle A_b, B3, v \rangle \xrightarrow{L6} \langle A_s, S2, . \rangle$

TR11 $\langle A_s, S2, \emptyset \rangle \xrightarrow{L8} \langle A_b, B2, . \rangle$

TR12 $\langle A_s, S2, sale_offer(p_j) \rangle \xrightarrow{L3} \langle A_b, B4, . \rangle$

TR13 $\langle A_s, S2, purchase_requirement(\pi^t) \rangle \rightarrow \langle A_s, S3, . \rangle$

TR14 $\langle A_s, S3, S_P \rangle \rightarrow \langle A_s, S4, . \rangle$

TR15 $\langle A_s, S4, \rho \rangle \xrightarrow{L5} \langle A_b, B2, . \rangle$

TR16 $\langle A_b, B4, generate_purchase_requirement(p_j) \rangle \rightarrow \langle A_b, B2, . \rangle$

TR17 $\langle A_b, B4, accept_offer(p_j) \rangle \xrightarrow{L9} \langle A_s, S5, . \rangle$

TR18 $\langle A_b, B4, reject_offer(p_j) \rangle \xrightarrow{L7} \langle A_s, S2, . \rangle$

TR19 $\langle A_s, S5, accept(p_j) \rangle \xrightarrow{L10} \langle A_b, B5, . \rangle$

TR20 $\langle A_s, S5, reject(p_j) \rangle \xrightarrow{L8} \langle A_b, B2, . \rangle$

TR21 $\langle A_x, K, wait \rangle \rightarrow \langle A_x, K, . \rangle$

One of the fundamental aims of this work is to develop an automated negotiation system. Therefore, the first thing we must demonstrate is that the dialogue model, the decision mechanisms, and the operational semantics, that is to say, the dialogue game framework for automated purchase negotiation is

able to generate dialogues automatically. This demonstration is presented in Appendix C.

3 The negotiation process

The negotiation process can be summarized as follows. Buyer agent performs the communicative act *desire_to_buy* which includes a purchase requirement. In order to construct the purchase requirement buyer agent applies the mechanism *B2: Generate Purchase Requirement* described in Section 2.3. Mechanism B2 guarantees that the new purchase requirement implies a bounded loss of *posd*. Depending on the buyer's attitude ($\eta = 1$), the seller's relaxation requests are evaluated in order to build the new purchase requirement which is expected to be the most valued by the seller agent. With this strategy the buyer agent limits the loss of *posd* and at the same time cooperates with the seller agent. In addition, an expressive ($\xi = 1$) buyer agent may value the new purchase requirement. A valuation implies that for each constraint in the purchase requirement, a numerical value is included which weights its importance.

From the perspective of the seller agent and given a received purchase requirement, there are two alternatives: offer a product which satisfies the purchase requirement, or send a relaxation request. Assuming that the seller agent is risk averse, seller agent will always offer a product if it satisfies the purchase requirement. Otherwise, the seller agent will respond with the *prefer_to_sell* communicative act which includes a relaxation request. The relaxation request is built in mechanisms *S3: Generate Potential Sale Offers* and *S4: Generate Relaxation Request*.

In order to help the understanding of the negotiation model, in the following, the mechanisms to generate purchase requirements and relaxation requests are described in detail.

3.1 Buyer generation of purchase requirements

Given a purchase requirement π^t which has been sent at instant $t \in \mathbb{N}$, this mechanism applies a *general concession strategy* which generates π^{t+1} so that $\alpha^{\pi^t} \geq \alpha^{\pi^{t+1}} \geq \alpha^{\pi^t} - \varepsilon$. $\varepsilon \in [0, 1]$ is an arbitrary value that fixes the maximum accepted loss of *posd*. It determines the agent's attitude with respect to how rapidly it is willing to make concessions.

The generation of a new purchase requirement implies to extract a new set of crisp constraints. Algorithm 1 implements the required functionality. In Step 1, a set $S^{\pi^{t+1}}$ is formed which contains the potential purchase requirements. Each potential requirement is obtained relaxing only one of the constraints in

Algorithm 1: Generate Purchase Requirements

Input: $\pi^t, \rho^t, \eta, \varepsilon$ **Output:** π^{t+1}

```
1 forall constraint  $i$  in  $\pi^t$  do
    relax constraint  $i$  to obtain  $\pi_i^{t+1}$ ;
    compute  $\alpha^{\pi_i^{t+1}}$ ;
    if  $\alpha^{\pi_i^{t+1}} \geq \alpha^{\pi^t} - \varepsilon$  then
         $S^{\pi^{t+1}} \leftarrow \pi_i^{t+1}$ ;
    end
end
2 if  $\eta == 1$  then
     $\pi^{t+1} = \arg \max_{S^{\pi^{t+1}}} (\rho^t \cdot (\pi^t - \pi_j^{t+1}))$ ;
else
     $\pi^{t+1} = \arg \max_{S^{\pi^{t+1}}} (\alpha^{\pi_j^{t+1}})$ ;
end
return  $\pi^{t+1}$ 
```

π^t , and must satisfy that its *posd* falls within the concession limits fixed by ε . In Step 2, buyer agent makes the selection of a purchase requirement from $S^{\pi_i^{t+1}}$. Selection depends on parameter η . For $\eta = 1$, buyer agent selects the purchase requirement which maximizes the scalar product of the relaxation request and the difference $\pi^t - \pi_i^{t+1}$. This difference is a measure of the relaxation made if π_i^{t+1} is selected. It is computed by subtracting the cut levels σ applied to each constraint in π^t and π_j^{t+1} . In short, the candidate which maximizes the seller's preferences is selected. For $\eta = 0$, the purchase requirement which maximizes the *posd* is selected. It means that buyer agent does not attend seller's relaxation requests.

A buyer's *purchase requirement valuation* is an expression of how important is the satisfaction of each constraint in a purchase requirement. Algorithm 2 implements the required functionality.

Algorithm 2: Generate Purchase Requirement Valuation

Input: π^{t+1}, ξ **Output:** v

```
if  $\xi == 1$  then
1    $v = \{1 - \alpha^{\pi_1^{(t+2)}}, \dots, 1 - \alpha^{\pi_m^{(t+2)}}\}$ ;
else
    $v = \emptyset$ ;
end
return  $v$ 
```

Given π^{t+1} (i.e. the requirement that is going to be submitted), a vector is obtained with the *posd* for all the possible purchase requirements that result from relaxing only one of the constraints. To sum up, what is being computed is the *posd* that would be obtained if constraint R_j is relaxed in a future negotiation round. The operation $1 - \alpha^{\pi_j^{(t+2)}}$ reflects that the valuation increases for low values of *posd*.

3.2 Seller generation of relaxation requests

The generation of relaxation requests is a two step process. First, the seller agent evaluates which products in its catalogue can be considered as *good sale offers*. The aim of this mechanism is not to generate specific sale offers to be sent to the buyer agent. Its main purpose is to serve as an input to the mechanism which finally builds the relaxation request expression. The idea is that the relaxation requests will be built in order to lead the buyer agent to make offers for products in this set. It must be noted that this mechanism only works when there are no products in the catalogue which satisfy the purchase requirement received.

Two aspects for carrying out the selection process of good sale offers have been considered: **Utility and Viability**. *Utility* is a local criterion which refers to the utility u_k of a sale offer s_k . *Viability* depends on two aspects: the degree of similarity between the product p_k and the purchase requirement π , and the buyer's purchase requirement valuation v . Both parameters are related by the function $prefer(s_j)$:

$$prefer(s_j) = \beta * u_j + (1 - \beta) * \widehat{viability}(p_j, \pi^t, v),$$

where $\widehat{viability} = 1 - \widehat{dist}(p_j, \pi^t, v)$. The distance term $\widehat{dist}(p_j, \pi^t, v)$ is computed as the euclidean distance between the product attributes and the limits specified in the constraints of the purchase requirement, weighted by the purchase requirement valuation. For high valuation levels, distance estimate increases. The agent's receptive profile parameter β modulates the weight of utility and viability. For $\beta = 1$ only utility is considered, while for $\beta = 0$ the product selection criterion is based on the expected viability of the sale offer. For $\beta = 0.5$ both criteria are equally considered.

Once the preference values for all the products in the catalogue have been obtained, a preference threshold is applied to generate the list of potential sale offers S_p . The threshold value influences the relaxation request generation process. Low threshold values imply a higher number of products in S_p , whereas a more selective threshold means a smaller set S_p . So, preference threshold modulates the impatience of the seller agent for constructing selective relaxation

requests.

Finally, given the set of potential sale offers S_p , the relaxation request is built with the aim of leading the buyer agent towards those sale offers. The basic principle of this mechanism is to get the buyer agent to relax those constraints that are not satisfied by the products contained in S_p . So, the relaxation request is defined as a vector $\rho = r_1, \dots, r_m$, where $r_k = 0$ indicates that constraint R_k is not satisfied by any product, and $r_k = 1$ that constraint R_k is satisfied by at least one product.

3.3 Clustering the seller catalogue of products

The seller agent performs calculations for every product in the catalogue and for each negotiation round in order to estimate the distance from each product to the constraints received. So, in addition to this approach, a modification to the algorithm is proposed in order to reduce the number of operations. The hypothesis is that applying clustering to the seller's product catalogue, the performance of the mechanism can be improved. The proposed algorithm works as follows.

The *fuzzy c-means* algorithm (see Appendix B) is applied over the p_k elements in the product catalogue S . When the process finishes a set of representatives $Rep^S = \{Rep_i^S | i = 1, \dots, c\}$ is obtained, where c is a predefined number of partitions. Now, for each product p_k , the different membership degrees to the different partitions $\mu_{1k}, \dots, \mu_{ck}$ are computed. Before entering a negotiation dialogue it is assumed that the seller agent has applied the clustering algorithm to the product catalogue S . It generates the set of product representatives Rep^S , one for each of the partitions made, which may be considerably smaller than the product catalogue. In order to compute the *prefer* value of a product a set of partial similarity estimates $sim^{Rep} = \{sim_i^{Rep} | sim_i^{Rep} = sim(Rep_i^S, \pi^t); i = 1, \dots, c\}$ are computed between the purchase requirement received and the representatives. Finally, the *prefer* value is computed for each product p_k as follows: 1) The partial similarity estimates are weighted by the corresponding membership degrees. The average of the partial estimates provides the global similarity estimate for p_k (it must be noted that with this approach we do not need to make similarity calculations for all the products in the catalogue but only for the representatives). Moreover, the variations of the similarity estimates will be smaller because the references will be the representatives, not the products. 2) The local utility u_k is used, not the utilities of the representatives. Summarizing, the *prefer* function is redefined as:

$$prefer(s_k) = \beta * u_k + (1 - \beta) * viability(sim^{Rep}, (\mu_{1k}, \dots, \mu_{ck}))$$

where

$$viability = \sum_{i=1}^n sim_i^{Rep} * \mu_{ik}.$$

4 Analysis of strategies

The presented negotiation framework allows for testing negotiation scenarios where agents behave according to different negotiation profiles. An expressive buyer agent ($\xi = 1$) will use purchase valuations, whilst a non-expressive one ($\xi = 0$) will not. A non-receptive ($\eta = 0$) buyer agent will not attend the seller's relaxation requests to generate new purchase requirements, whilst a receptive ($\eta = 1$) agent will consider both the relaxation request and the potential overall satisfaction degree of the purchase requirement to be generated. On the other hand, an expressive seller agent ($\psi = 1$) will use relaxation request, whilst a non-expressive one ($\psi = 0$) will not. Finally, the seller's receptive profile β may indicate no receptiveness ($\beta = 1$), intermediate receptiveness ($\beta = 0.5$), or maximum receptiveness ($\beta = 0$). The receptive profile determines how the relaxation request is formed.

In the following, the validity of expressiveness vs receptiveness relationships for each agent are evaluated, and then, for the valid relationships, an analysis of joint strategies is presented.

4.1 Analysis of strategies at agent level

To denominate the different strategies the following convention is used. **BA** refers to a Buyer agent Attitude or behaviour, and **SA** refers to a Seller agent Attitude or behaviour. Next, first the types of expressive behaviour **ne** and **e** appear, to define non-expressiveness and expressiveness respectively. Finally the types of receptive behaviour **nr** and **r** appear, to define non-receptive and receptive respectively. Only for a seller agent and for a receptive behaviour (**r**) a numerical suffix is added to consider the level of receptiveness. We look first at agent level, beginning with the *buyer agent*.

(BAer) expressive and receptive, **(BANer)** non-expressive and receptive, and **(BANenr)** non-expressive and non-receptive are coherent strategies. However, **(BAenr)** expressive and non-receptive strategies makes no sense, as the purpose of a purchase requirement valuation is to redirect the negotiation in such a way that the seller agent sends useful relaxation request. If the agent does not consider relaxation requests, the valuation is of no use whatsoever. To sum up, the buyer agent can behave in three different ways: **BAer**, **BANer** y **BANenr**.

For the *seller agent*, **(SAer1)** expressive and receptive ($\beta = 0$), **(SAer0.5)**

expressive and receptive ($\beta = 0.5$), **(SAenr) expressive and non-receptive** ($\beta = 1$), and **(SANenr) non-expressive and non-receptive** are valid strategies. However, **(SAner1 or SAner0.5) non-expressive and any receptive strategy** makes no sense, as a non-expressive seller agent does not send relaxation requests, and the main purpose of a receptive strategy is to direct the relaxation request generation. To sum up, the seller agent can behave in accordance with four different strategies: **SAer1**, **SAer0.5**, **SAenr** and **SAnenr**.

4.2 Analysis of combination of strategies

There are 12 different combinations of strategies. However, some of these combinations are not coherent. In **BAer vs SAnenr** the buyer agent's valuations are not taken into account by the seller agent, which furthermore is not expressive. This aspect is detectable by the buyer agent, given that it does not receive relaxation request. A rational agent will not send valuations if it knows they are of no use. The best strategy for a buyer agent under these circumstances is to change to a non-expressive and non-receptive strategy **BAnenr**. In **BAner vs SAnenr** neither of the agents is expressive, so for the buyer agent to be receptive makes no sense, and furthermore this fact is detectable by the buyer agent. A rational buyer agent would change to a **BAnenr** strategy. After this analysis, there are 10 pairs of balanced strategies : **BAer vs SAer1**, **BAer vs SAer0.5**, **BAer vs SAenr**, **BAner vs SAer1**, **BAner vs SAer0.5**, **BAner vs SAenr**, **BAnenr vs SAer1**, **BAnenr vs SAer0.5**, **BAnenr vs SAenr**, y **BAnenr vs SAnenr**. To simplify this repertoire the following groupings are built:

BAer vs SAe_r_ has in common the fact that the buyer agent is simultaneously expressive and receptive, and the seller agent is expressive. Furthermore it seems obvious that the seller agents' different receptive profiles will affect the results of the negotiation, because the generation of relaxation request varies depending on this profile. Therefore, a priori we need to test with the three combinations that make up the group.

BAner vs SAe_r_ has in common the fact that the buyer agent is not expressive, it is receptive, and that the seller agent is expressive. When the seller agent is receptive, intuitively it can be stated that the results of the negotiations are different to those of the previous group. This is so because the buyer agent's valuations are not available to the seller agent. However, when the seller agent is not receptive the scenario is identical to the previous group. In other words, if the seller agent is not receptive it makes no difference whether the buyer agent sends valuations or not. In conclusion, the **BAner vs SAenr** pair is identical to **BAer vs SAenr**, as far as the results of the negotiation are concerned. To

speed up the tests of this type, we restrict the tests to **BAner vs SAer**.

BAner vs SA_e_r_ is characterised by the non-expressiveness and non-receptiveness of the buyer agent. So, it makes no difference whether the seller agent is expressive or receptive as the buyer agent will be unable to take it into account. To speed up the execution of the tests in this group, we have opted to define as representative the **BAner vs SAner** pair.

Summarizing, there exist 6 pairs of strategies that can resolve negotiations with disparate results: 1) **BAer vs SAer1**, 2) **BAer vs SAer0.5**, 3) **BAner vs SAer**, 4) **BAner vs SAer0.5**, 5) **BAner vs SAer1** and 6) **BAner vs SAner**.

5 Experiments

5.1 Settings

The **buyer agent's preferences** are described as a set of 5 fuzzy constraints $R_{1...5}^f$ over 5 issues $a_{1...5}$, where each fuzzy constraint restricts one issue. The **seller agent's preferences** are described as a catalogue of products. The products are defined taking into account the fuzzy constraints. By restricting the range of values that each attribute may take, products with predefined utilities for the buyer agent can be defined.

First, a set of products which maximizes the buyer agent's utility is generated. This set is called the *solution set* $S_{sol} \subseteq S$, and it has been fixed to provide the buyer agent an overall satisfaction degree $\alpha(p_k) = 0.7 \forall p_k \in S_{sol}$. Finally, the *noise set* $S_{noise} \subseteq S | S_{sol} \cup S_{noise} = S$ is built to provide an $\alpha(p_k) \leq 0.3 \forall p_k \in S_{noise}$.

The next step is to assign utility values from the seller's side (i.e. assign u_k to each product). In the case of S_{noise} , utilities are randomly generated using a uniform allocation between 0.9 and 1, while for S_{sol} a uniform allocation between 0 and 0.69 is used. Also randomly, the utility of one of the products from the set S_{sol} is assigned 0.7. The aim is to see if this optimal solution is reached after a negotiation. With these utilities, the seller agent's preferred sale offers correspond to the noise set products. However, a smart seller agent will come to the conclusion that these products are not viable sale offers.

Settings defined above have been used in the experiments without clustering (i.e. without grouping the products in the seller's catalogue). In the clustering approach the experimental settings are similar. However, only the **BAner vs SAer0.5** strategies are considered, which correspond to the best results obtained with the non-clustering approach. In addition, an experiment implies to generate one or more solution sets, and one or more noise sets. The different

sets are generated restricting the range of values that each attribute may take. In this way we could, for instance, generate 2 solution sets and 3 noise sets with the restrictions shown in Table 1. Once the ranges of values for the different sets are established, the products in the different sets are randomly generated within these ranges of values. Different sizes of catalogues, varying the number of products in each set from 16 to 256 have been generated. For a given experiment, i.e., a solution and noise sets configuration and a given size, 300 negotiation dialogues are launched varying the product catalogues. Moreover, the same experiment is performed using and without using clustering in order to test the validity of the hypothesis. It must be noted that in the experiments, the number of clusters is known in advance.¹ The following distributions of noise and solution sets have been tested: 1 solution + 1 noise, 1 solution + 4 noise, 2 solution + 1 noise, 2 solution + 3 noise. The aim of these distributions is to have scenarios where the probability to find good solutions at random is lower or higher.

5.2 Results without clustering

For each of the 6 pairs of strategies under analysis, and for different sizes of catalogues, 300 negotiations are carried out. The median of the joint utility and the success rate has been analyzed, where the success rate estimates the number of times that the pareto-optimal solution is obtained, that is to say, the solution in which $u_k = 0.7$. In Fig. 1 the results obtained (with the 95% confidence intervals) from the tests of the **BAner vs SAner**, and **BAner vs SAer05** scenarios are summarised. In the upper graphic the medians are shown, and in the bottom the success rates. The success rates follow the same trend for all the catalogues, with an improvement in the case of the BAner vs SAer05 scenario. Regarding the medians, for catalogues with up to 64 products, the results are optimal. For catalogues with more than 256 products the strategies tend to converge. It should be recalled that a heavily populated catalogue means the seller agent will have high utility sale offers with a higher probability. For the other four strategies the results are identical to the results obtained in the BAner vs SAner scenario. These results for the **BAer/BAner vs SAer1** scenarios were foreseeable, taking into account that the seller agent only considers the purchase requirements, and so, the relaxation requests have no effect. In the **BAer vs SAer0.5** scenario the valuation of the purchase requirements negatively impacts the viability estimation, so it unfavourably modifies the preferences for the different sale offers. The results obtained for

¹There exist techniques which estimate the optimal number of clusters in terms of two values, the partition and entropy coefficients. These techniques could be used in real negotiation scenarios.

the **BAner vs SAer** scenario show how when a seller agent limits itself to looking out for its best interest, the agent worse off.

In more detail, in the **BAner vs SAner** scenario the success rate stabilizes around 10%, although for 4 and 8 products the rate is higher, which is logical, as the number of relaxation combinations is higher than the number of products. There is a dip in the median value with 16 products, however, the median grows again as the number of products increases. This effect was foreseeable, as when the number of products increases, the probability that the seller agent has products with a high utility increases. Finally, in the **BAner vs SAer0.5** scenario it can be seen how the results are significantly better in every case. This test shows that the expressiveness of the seller agent is key to obtain satisfactory solutions.

In Fig. 2 two graphs depict the percentage improvement in the success rates and the comparative percentage improvement in utility. The improvement in the success rates portrays the comparative between the percentage of success rates obtained in the BAner vs SAer05 scenario and those obtained in the BAner vs SAner scenario (similar to the scenario described in [16]). This graph shows an improvement trend of improvement in the success rates. For catalogues with a small solution set the improvement is of approximately 200%, with an increase of around 325% for catalogues of 16 and 32 products. It should be taken into account that when there are very few products, the possibility of a good solution being found at random, is higher than when the catalogue is large, which is why the improvement is smaller for 4 and 8 products. Although for 64 products the success rate decreases to 250%, in general, as the size of the catalogues increases there is a trend for the rates to improve. As the catalogues become very large, the probability of obtaining an optimal solution without expressiveness decreases down to zero, whereas with expressiveness the optimal solution is explicitly searched for.

The relative improvement in utility is a comparative measure that shows the improvements obtained with the BAner vs SAer05 strategies. It can be observed that the reference catalogue is the one with 16 products, which is the scenario with which the maximum utility is obtained. So, the graph shows a percentage of relative improvement of 100% for this catalogue. For large catalogues the percentage of relative improvement decreases below 10%. The minimum percentage improvement for smaller catalogues is 10% with an average value of around 60%.

5.3 Results with clustering

All the experiments show an improvement in the duration of the negotiation dialogues when using the clustering approach. In Fig. 3 the percentage improvement is presented. It increases as the number of product increases, with around 35% of improvement for noise and solutions sets with 256 products.

Fig. 4 shows a set of boxplots representing the utilities the seller agent obtains, and Fig. 5 shows the pareto-optimality rate (success rate). Using clustering, negotiations perform similar or better than when clustering is not used. For catalogues with 1 solution set, clustering strictly performs better for heavily populated catalogues. For catalogues with 2 solution sets, clustering always performs better. Last result was foreseeable, taking into account that the number of solution representatives is higher with respect to the number of noise representatives. Regarding the pareto-optimality rate, clustering is always useful, and usefulness increases for large product catalogues. Moreover, in scenarios where the probability of reaching good agreements (2 solution + 1 noise and 2 solution + 3 noise scenarios) is high, this proposal performs even better.

6 Related work

There are vastly different research directions regarding automated bilateral negotiations covering different areas such as game theory, evolutionary computation or distributed artificial intelligence, many of them involving integrative negotiation mechanisms [3, 7, 14, 15, 20]. In this paper, a non-mediated fuzzy constraint based negotiation framework for competitive e-marketplaces has been proposed and analyzed. In competitive markets [11, 16, 5], there is an inherent need to restrict the amount of private information the agent reveals. However, this restriction can have a detrimental effect on the search for a solution. As it is stated above, especially in the case of multi-attribute negotiations, it is possible to reach a more satisfactory agreement by means of an adequate combination of attributes or constraints. However, most solutions put forward to tackle this problem are mediated and iterative mechanisms, which are applicable to preference models based on linear-additive or quasi-concave utility functions [2, 3, 13]. As an alternative to these solutions, the negotiation framework proposed is based on a dialogue of constraint based offers in which preferences or satisfaction degrees are partially disclosed.

There are several works which use fuzzy constraints to model preferences, however, most of them use fuzzy constraints simply to model preferences, while interaction is described as a positional bargaining. The FeNAs platform [10]

uses fuzzy constraints and permits correlated multiple bilateral negotiations, however, the agent's interactions are positional. In [12] a general framework for multi-attribute and multilateral negotiation based on fuzzy constraints is proposed. The negotiation model is based on fuzzy constraints, which when applied to a distributed domain of agents are organized as a network of distributed fuzzy constraints. This work makes an interesting contribution to the regularization of the mechanisms for calculating the satisfaction degrees. The negotiation model is also based on single-point offers. In [18] a dialogue game framework for agent purchase negotiations is presented. This paper focuses on defining the dialogue rules, but does not describe specific decision mechanisms.

The most closely related work to ours is presented in Luo et. al. [16]. They propose a fuzzy constraint based model for bilateral multi-issue negotiations in semi-competitive environments, which uses constraints to express offers, and includes the idea of rewards and restrictions. The most noticeable aspects are related to the acceptability function defined and with the operators used to apply the prioritization of the fuzzy constraints. Under the assumption of non-strategic behaviour, the model exhibits communicative asymmetry. A buyer agent expresses offers by means of fuzzy constraints, while a seller agent simply rejects or accepts a proposal. This limitation in turn means that the joint decision is not balanced and the search for a good agreement for both parties is somewhat random. In order to cope with the drawback of this asymmetry, in our work a more expressive communication model lets the buyer agent to value the degree of importance that each submitted constraint has, and lets the seller agent to inform which is the preference for a specific constraint to be relaxed. The aim of this approach is to provide the basis for constructing integrative negotiation mechanisms. Finally, in [6, 17] a set of auction-based negotiation protocols among agents with nonlinear utility functions are proposed. However, the main difference to our work is that we employ a non-mediated solution specially adapted to purchase negotiation scenarios.

7 Conclusions

This paper presented a fuzzy constraint based model for automated purchase negotiations. The model uses fuzzy constraints to model preferences and to express proposals. In addition, agents can add meta-information to offers and counter offers to enhance the convergence of the negotiation processes. Formally, the negotiation framework is defined as a dialogue game, with a set of locutions, a set of decision mechanisms and a set of transition rules which link the locutions to the decision mechanisms. The analysis of the model includes a study of the agents' strategic behaviour and an experimental evaluation. The results show

that the best joint strategies correspond to a buyer agent using non-valued constraint based proposals, and a seller agent submitting relaxation requests attending to utility and viability of potential sale offers. We can conclude that a limited disclosure of preferences (expressiveness) of the seller agent is essential to obtain an improvement in the negotiations. However, the expressiveness of the buyer which consist of attaching a valuation to a purchase requirement makes the results come close to those achieved with the reference strategies: non-expressive and non-receptive. However, it must be pointed out that an 'inexpressive' buyer agent is more expressive than an 'inexpressive' seller agent because a buyer agent covers an offer subspace with a given purchase proposal, while a seller agent expresses offers as concrete products or rejections to purchase requirements.

In addition, a clustering algorithm applied to the catalogue of products of the seller agent has been proposed. By means of the clustering of products, the seller agent optimizes the generation of relaxation requests in terms of computation time, also getting to increase the joint utility of the agreements reached.

In terms of future work, the main research task that should be considered is the inclusion of different constraint based preference models. In the fuzzy constraint based model, a buyer agent uses the *min* operator to compute the overall satisfaction degree of an offer, and applies a concession algorithm to generate new purchase requirements which relaxes one constraint per negotiation round. Moreover, once a constraint has been relaxed, the proposed algorithm does not consider to return to a previous relaxation level. However, in a more complex setting such as a weighted constraint based framework, the overall satisfaction levels strongly depend on the different constraint relaxation levels. This makes necessary the development of new concession algorithms which incorporate new techniques to efficiently use the relaxation requests.

Acknowledgements

This work has been supported by the Spanish Ministry of Education and Science grant TIN2008-06739-C04-04.

References

- [1] Jose J. Castro-Sanchez, Nicholas R. Jennings, Xudong Luo, and Nigel R. Shadbolt. Acquiring domain knowledge for negotiating agents: a case of study. *International Journal of Human-Computer Studies*, 61(1):3–31, 2004.

- [2] H. Ehtamo, R. P Hamalainen, P. Heiskanen, J. Teich, M. Verkama, and S. Zionts. Generating pareto solutions in a two-party setting: constraint proposal methods. *Management Science*, 45(12):1697–1709, 1999.
- [3] P. Faratin, C. Sierra, and N. R Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205–237, 2002.
- [4] F. Fisher and W. Ury. *Getting to yes: Negotiating an agreement without giving in*. Random House Business Books, 1981.
- [5] Minghua He, N. R Jennings, and Ho-Fung Leung. On agent-mediated electronic commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):985–1003, 2003.
- [6] Takayuki Ito, Mark Klein, and Hiromitsu Hattori. A multi-issue negotiation protocol among agents with nonlinear utility functions. *Journal of Multiagent and Grid Systems*, 4(1):67–83, 2008.
- [7] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Protocols for negotiating complex contracts. *IEEE Intelligent Systems*, 18(6):32–38, 2003.
- [8] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice-Hall, 1995.
- [9] R. Kowalczyk. Fuzzy e-negotiation agents. *Soft Computing*, 6(5):337–347, 2002.
- [10] R. Kowalczyk and V. Bui. Fenax: a fuzzy e-negotiation agents system. In *Proceedings of the IEEE/IAFE/INFORMS 2000 Conference on Computational Intelligence for Financial Engineering (CIFEr)*, pages 26–29, New York, USA, 2002.
- [11] K. Kurbel and I. Loutchko. Towards multi-agent electronic marketplaces: What is there and what is missing? *The Knowledge Engineering Review*, 18(1):33–46, 2003.
- [12] R. Lai and M. W. Lin. Modeling agent negotiation via fuzzy constraints in e-business. *Computational Intelligence*, 20(4):624–642, 2004.
- [13] Raymond Y.K. Lau, Maolin Tang, and On Wong. Towards genetically optimised responsive negotiation agents. In IEEE Computer Society, editor, *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)*, pages 295–301, Beijing, China, September 20-24 2004. IEEE Computer Society.

- [14] Miguel A. Lopez-Carmona and Juan R. Velasco. An expressive approach to fuzzy constraint based agent purchase negotiation. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2006)*, pages 429–431, Hakodate, Japan, 2006.
- [15] Miguel A. Lopez-Carmona, Juan R. Velasco, and Ivan Marsa-Maestre. The agents’ attitudes in fuzzy constraint based automated purchase negotiations. In *Multi-Agent Systems and Applications V*, volume 4696 of *Lecture Notes in Artificial Intelligence*, pages 246–255, Berlin, Germany, 2007. Springer Verlag.
- [16] X. Luo, N. R Jennings, N. Shadbolt, Ho-Fung-Leung, and J. H M Lee. A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. *Artificial Intelligence*, 148(1-2):53–102, 2003.
- [17] Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco, and Enrique de la Hoz. Effective bidding and deal identification for negotiations in highly nonlinear scenarios. In Decker, Sichman, Sierra, and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1057–1064, Budapest, Hungary, May, 10-15 2009.
- [18] P. McBurney, R. M. Van Euk, S. Parsons, and L. Amgoud. A dialogue game protocol for agent purchase negotiations. *Journal of Autonomous Agents and Multi-Agent Systems*, 7(3):235–273, 2003.
- [19] P. McBurney and S. Parsons. Games that agents play: A formal framework for dialogues between autonomous agents. *Logic, Language and Information*, 11(3):315–334, 2002.
- [20] Quoc Bao Vo and Lin Padgham. Searching for joint gains in automated negotiations based on multi-criteria decision making theory. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2007)*, pages 496–503, Honolulu, Hawaii, USA, 2007.
- [21] Quoc Bao Vo, Lin Padgham, and Lawrence Cavedon. Negotiating flexible agreements by combining distributive and integrative negotiation. *Intelligent Decision Technologies*, 1(1-2):33–47, 2007.

Author’s Bios

Miguel Angel Lopez-Carmona received his B.E. in Electronics Engineering from the Universidad de Alcala (Madrid, Spain), M.E. in Telecommunication

Engineering from the Polytechnic University of Madrid, and PhD degree from the Universidad de Alcala in 2006. From 1995 to 2000, he worked in Logytel and Alcatel as project manager and research scientist in the areas of railways and defense. From 1999 to 2006, he was assistant professor at the Universidad de Alcala, and from 2006 he is associate professor at the Department of Computer Engineering at the Universidad de Alcala. Currently, his research interests include automated negotiation, multi-agent systems, and distributed optimization.

Ivan Marsa-Maestre is an assistant lecturer at the Computer Engineering Department at the University of Alcala. His research interests include agent-based systems, automated negotiation, smart environments and mobile agent security. He received from the Universidad de Alcala his M.E. in Telecommunication Engineering in 2003, and his PhD degree in 2009.

Juan Ramon Velasco Perez received his M.E. and the PhD from the Polytechnic University of Madrid in 1991 and 1997 respectively. Currently he is full professor at the Department of Computer Engineering at the Universidad de Alcala, and the coordinator of the Telematic Services Engineering Research Group. His research interests include multi-agent systems, smart environments, automated negotiation and evolutionary algorithms.

Bernardo Alarcos Alcazar is associated professor at the Department of Computer Engineering at the Universidad de Alcala. He received the M.E. degree in Telecommunication Engineering in 1997 from the Polytechnic University of Madrid, and in 2004 the PhD from the Universidad de Alcala. His research interests are autonomic computing, security networks and ad-hoc networking.

Enrique de la Hoz de la Hoz received his M.E. in Telecommunication Engineering degree from the Universidad de Alcala in 2001. He is assistant professor at the Universidad de Alcala since 2003, and a PhD candidate. His research activities are mainly focused on security in ad-hoc networks, distributed optimization and open source software. Currently, he is involved in research activities with RedIRIS (Spain) and UNINETT (Norway).

Appendix A: UML state diagram of the negotiation protocol

Fig. 6 shows the UML state diagram of the negotiation protocol. Each state represents a decision mechanism, and each line connecting the state boxes shows the output which activates the transition to another state. If the transition links the states of different agents, then the line connecting the state boxes also shows the invoked locution.

Appendix B: Fuzzy c-means clustering algorithm

The aim of the clustering is to carry out automatic grouping of the products in the seller's catalogue. The fuzzy c-means algorithm has been used in order to compute the partitions. This grouping algorithm is widely used in different fields such as pattern recognition, data mining or image processing [8]. Each partition will be formed by a subset of products and a representative product. In general, when using fuzzy c-means a set of partitions is generated where each partition has a representative, and every element belongs to the different partitions simultaneously at different membership degrees. Let $X = \{x_1, x_2, x_3, \dots, x_n\}$ be a set of n objects where $x_i \in \mathbb{R}^S$ is an object described as a set of S real values which are measures of its characteristics. A fuzzy c-partition of X is a class of c fuzzy sets V_1, V_2, \dots, V_c where c is an integer in the range $[2, n]$. So, a fuzzy c-partition for X is defined as $M_{fcn} = (U \in \mathbb{R}^{c \times n})$. The membership degree of an object k to a partition i is defined as $\mu_{ik} \in [0, 1]$, where $\sum_{i=1}^c \mu_{ik} = 1, \forall k$. Now, the main goal is to find the best U matrix partition in M_{fcn} , which is achieved when the following function is minimized:

$$J_m(U, V) = \sum_{k=1}^n \sum_{i=1}^c \mu_{ik}^m \cdot d_{ik}^2(v_i, x_k), U \in M_{fcn}, 1 < m < \infty.$$

In this function v_i defines the representative (prototype or centroid) of each class, m expresses the fuzziness of the different sets, and d is the euclidean distance. The representatives are computed using the following formula:

$$v_i = \left(\frac{\sum_{k=1}^n \mu_{ik}^m \cdot x_k}{\sum_{k=1}^n \mu_{ik}^m} \right),$$

and the fuzzy membership using:

$$\mu_{ik} = \left[\frac{\left(\frac{1}{d_{ik}^2(v_i, x_k)} \right)^{1/(m-1)}}{\sum_{j=1}^c \left(\frac{1}{d_{ik}^2(v_j, x_k)} \right)^{1/(m-1)}} \right].$$

The fuzzy c-means algorithm iterates recalculating v_i and μ_{ik} in order to minimize $J_m(U, V)$. It is established that this algorithm converge for any $m \in [1, \infty)$, but fuzziness of the partitions increases as m increases [8]. So, m must be chosen depending on the specific problem considered. In the negotiation scenario, hyperspheric sets are assumed, and an *a priori* number of fuzzy sets is defined. If needed, there exist techniques which estimate the optimal number of fuzzy sets in terms of two values, the partition and entropy coefficients.

Appendix C: Demonstration of the automatic generation of dialogues

This appendix provides the demonstration that the negotiation framework proposed generates dialogues automatically (i.e. the negotiation protocol is complete).

We need to demonstrate: (a) that all the locutions can be activated by one or more of the decision mechanisms, and (b) that every time one of these mechanisms is executed it ultimately activates a locution. To support these propositions we first present for (a), a list of the locutions, together with the mechanisms that activate them, and the transition rule in which the activation is featured.

L1: Mechanism B1 (Rule TR1).

L2: Mechanism S1 (Rule TR4).

L3: Mechanism S2 (Rule TR12).

L4: Mechanism B3 (Rule TR9).

L5: Mechanism S4 (Rule TR15).

L6: Mechanism B3 (Rule TR10).

L7: Mechanism B4 (Rule TR18).

L8: Mechanism S2 (Rule TR11); Mechanism S5 (Rule TR20).

L9: Mechanism B4 (Rule TR17).

L10: Mechanism S5 (Rule TR19).

L11: Mechanism B5 (Rule TR6); Mechanism S6 (Rule TR7).

For (b), we show for each mechanism and their possible states: whether they activate a locution, or whether they indirectly activate a mechanism that in turn activates a locution. We also present the transition rules where these connections are established.

B1: Output `have_need` activates L1 (Rule TR1).

B1: Output `have_no_need` activates the mechanism B1 (Rule TR2).

B2: Output `empty_set` activates the mechanism B5 (Rule TR5).

B2: Output π activates the mechanism B3 (Rule TR8).

B3: Output `empty_set` activates the locution L4 (Rule TR9)

B3: Output `v` activates the locution L6 (Rule TR10).

B4: Output `generate_purchase_requirement` invokes the mechanism B2 (Rule TR16).

B4: Output `accept_offer` invokes the locution L9 (Rule TR17).

B4: Output `reject_offer` invokes L7 (Rule TR18).

B5: Output `withdraw_dialogue` invokes L11 (Rule TR6).

S1: Output `wish_not_to_enter` activates the mechanism S1 (Rule TR3).

S1: Output `wish_to_enter` activates the locution L2 (Rule TR4).

- S2*: Output *empty_set* invokes L8 (Rule TR11).
- S2*: Output *sale_offer* invokes L3 (Rule TR12).
- S2*: Output *purchase_requirement* invokes the mechanism S3 (Rule TR13).
- S3*: Output S_p activates the mechanism S4 (Rule TR14).
- S4*: Output ρ invokes the locution L5 (Rule TR15).
- S5*: Output *accept* invokes L10 (Rule TR19).
- S5*: Output *reject* invokes L8 (Rule TR20).
- S6*: Output *withdraw* invokes the locution L11 (Rule TR7).

It can easily be proven that all the mechanisms generate a locution or activate a mechanism that then generates a locution, or activate a mechanism that then generates another mechanism that finally generates a locution.

	R_1^f	R_2^f	R_3^f	R_4^f	R_5^f
Sol1	[61,100]	[61,100]	[91,100]	[91,100]	[61,70]
Sol2	[61,100]	[61,100]	[61,70]	[61,70]	[91,100]
Noise1	[40,60]	[40,60]	[1,20]	[1,20]	[1,20]
Noise2	[1,20]	[1,20]	[1,20]	[40,60]	[40,60]
Noise3	[1,20]	[40,60]	[40,60]	[1,20]	[1,20]

Table 1: An example of ranges of attributes for 2 solution and 5 noise sets.

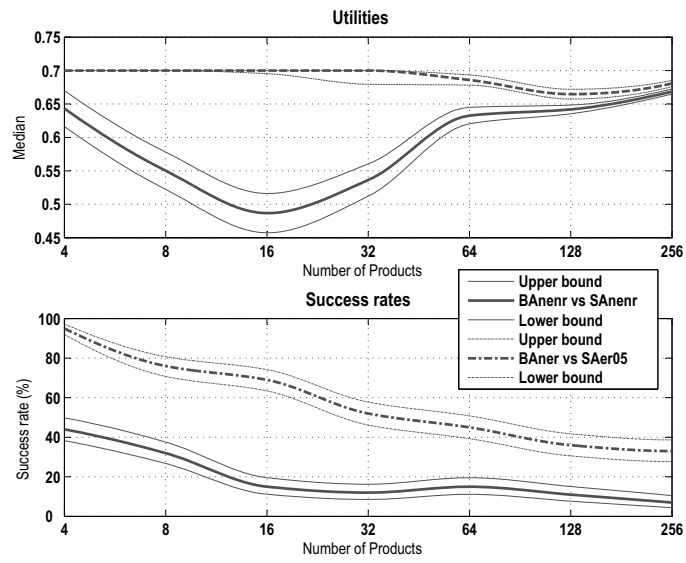


Figure 1: Comparative of the BAner vs SAner and the BAner vs SAer05 strategies.

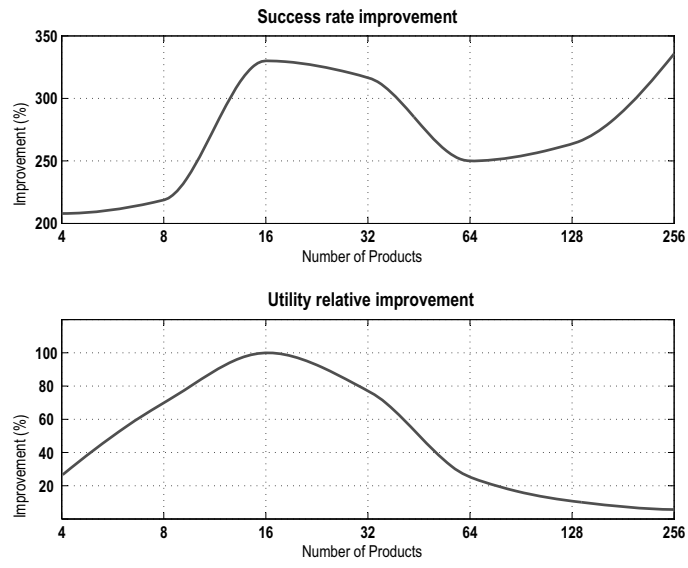


Figure 2: Improvement in the success rate and relative improvement of utility.

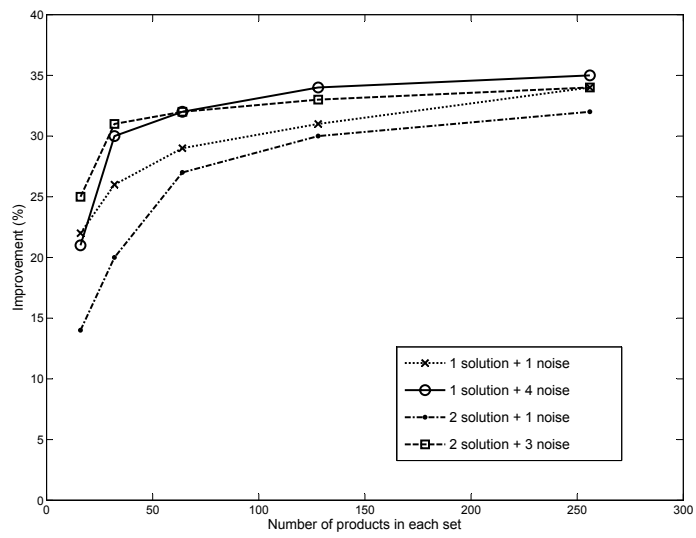


Figure 3: Improvement in the duration of the negotiation dialogues using fuzzy c-means.

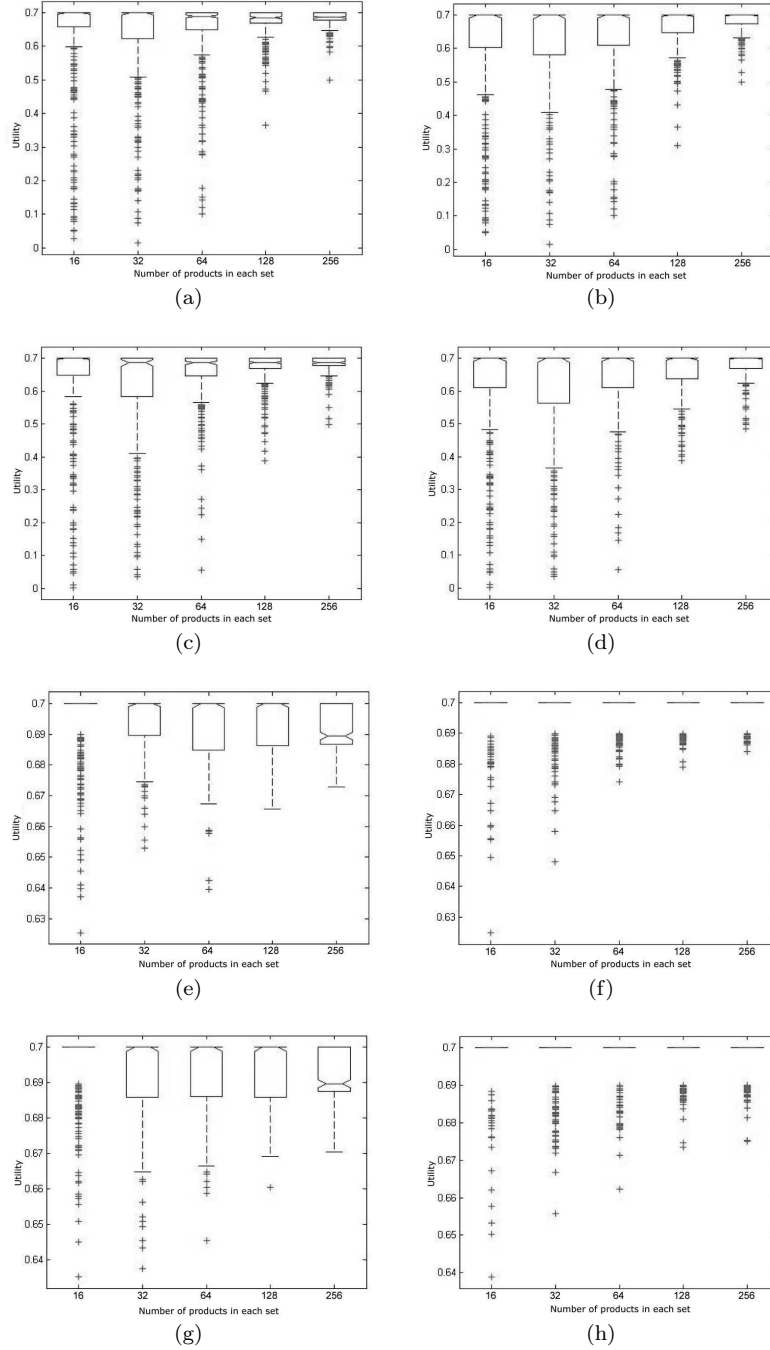
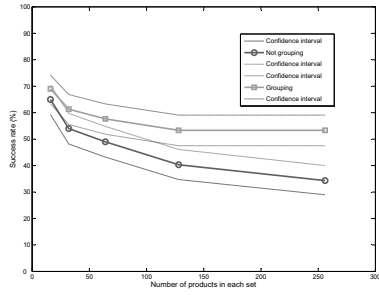
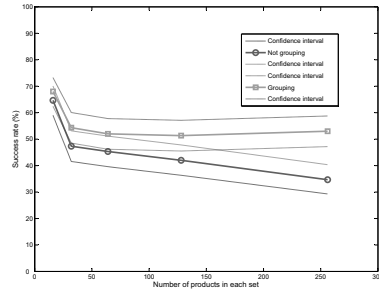


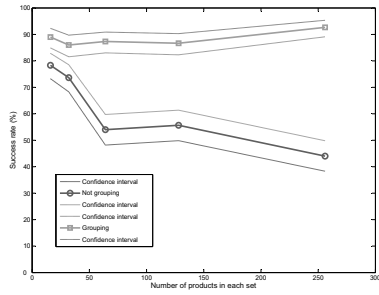
Figure 4: Boxplot of utilities achieved by the seller agent. Without clustering: (a) 1 solution+1 noise (c) 1 solution + 4 noise (e) 2 solution+1 noise (g) 2 solution+3 noise. With clustering: (b) 1 solution+1 noise (d) 1 solution + 4 noise (f) 2 solution+1 noise (h) 2 solution+3 noise.



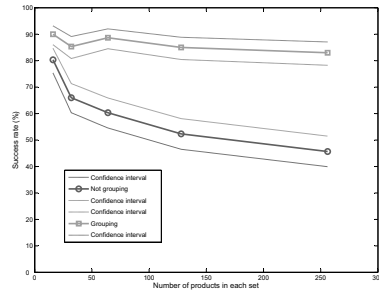
(a)



(b)



(c)



(d)

Figure 5: Pareto optimality rate % (success rate %) vs Number of products per noise and solution set: (a) 1 solution+1 noise (b) 1 solution + 4 noise (c) 2 solution+1 noise (d) 2 solution+3 noise.

Figure 6: Negotiation protocol UML state diagram.

