

An Expressive Approach to Fuzzy Constraint Based Agent Purchase Negotiation

Miguel A. López-Carmona
Department of Automática, University of Alcalá
Escuela Politécnica, Ctra. NII, km. 33.600
28871 Alcalá de Henares (Madrid), Spain
miguellop@aut.uah.es

Juan R. Velasco
Department of Automática, University of Alcalá
Escuela Politécnica, Ctra. NII, km. 33.600
28871 Alcalá de Henares (Madrid), Spain
juanra@aut.uah.es

ABSTRACT

This paper presents a fuzzy constraint based model for bilateral multi-attribute agent purchase negotiations in competitive trading environments. Argumentation is used as a key mechanism to improve agreements, in contrast to other fuzzy constraint based models which are limited to quantitative offers and counter-offers. A set of locutions and decision mechanisms which fire them are specified, so that each agent may decide its degree of cooperation and expressiveness.

Categories and Subject Descriptors

I.2 [Artificial Intelligence]: Distributed Artificial Intelligence, Problem Solving

General Terms

Algorithms, Theory, Experimentation

Keywords

Negotiation, Fuzzy Constraints, Strategies

1. INTRODUCTION

The use of fuzzy constraints is an efficient way of capturing requirements, and a useful mechanism for representing trade-offs between the possible values of attributes [1, 3]. Moreover, the solution space can be explored in a given exchange and so means that the search for an agreement is more efficient. The problem with previous approaches is that the ability of the seller agent to express its needs is limited [3]. As a consequence the joint decision process is not balanced. We develop a more expressive approach which lets the buyer agent value the degree of importance that each submitted constraint has, and lets the seller agent inform how constraints should be relaxed. We propose a general framework to implement these mechanisms, so that each agent may decide its degree of cooperation and expres-

siveness, which in turn may have effects on the speed of convergence and the quality of the agreement.

2. DOMAIN KNOWLEDGE

Let $\mathcal{B}_{req} = (X, D, C^f, N_b)$ be the requirement model of the buyer agent, where $X = \{x_i | i = 1, \dots, n\}$ is the set of attributes, $D = \{d_i | i = 1, \dots, n\}$ is the set of domains of the attributes, and $C^f = \{R_i^f | i = 1, \dots, m\}$ is the set of fuzzy constraints which expresses the requirements on the attributes. $N_b = \{\xi, \eta\}$ is the negotiation profile, where $\xi \in [0, 1]$ controls the use of purchase requirement valuations, and $\eta \in [0, 1]$ modulates the buyer's attitude regarding a relax requirement. The function that indicates how well a given constraint R_i^f is satisfied is the *satisfaction degree function*: $\mu_{R_i^f}(X) \rightarrow [0, 1], i = 1, \dots, m$. The *overall (or global) satisfaction degree* of a potential solution p_X is defined as: $\alpha(p_X) = \oplus \{\mu_{R_i^f}(p_X) | R_i^f \in C^f\}$, where $\oplus$ is an aggregation from $[0, 1]^m$ to $[0, 1]$.

DEFINITION 1. A purchase requirement is defined as $\lambda_{\mathcal{B}_{req}} = \bigcap \left\{ R_i^{c(\sigma_j^i)} \right\}$, where $R_i^{c(\sigma_j^i)}$ is the induced crisp constraint at cut level $\sigma_j^i \in [0, 1]$. A purchase requirement may be formed by a subset of fuzzy constraints, where $\text{cons}(\lambda_{\mathcal{B}_{req}})$ denotes the set of constraints associated with each $\lambda_{\mathcal{B}_{req}}$.

DEFINITION 2. Given a $\lambda_{\mathcal{B}_{req}}$, the potential overall satisfaction degree (posd) the buyer gets if this requirement is met is $\alpha^{\lambda_{\mathcal{B}_{req}}} = \oplus \{\sigma_j^i\}$, where the cut level applied to constraints which are not included in the $\lambda_{\mathcal{B}_{req}}$ takes value 1.

DEFINITION 3. Given a $\lambda_{\mathcal{B}_{req}}^k$ defined at instant k , a concession strategy may be defined as a mechanism which generates a new $\lambda_{\mathcal{B}_{req}}^{k+1}$ such that $\alpha^{\lambda_{\mathcal{B}_{req}}^{k+1}} < \alpha^{\lambda_{\mathcal{B}_{req}}^k}$, where $\alpha^{\lambda_{\mathcal{B}_{req}}^k} - \alpha^{\lambda_{\mathcal{B}_{req}}^{k+1}}$ is minimized.

DEFINITION 4. Given a $\lambda_{\mathcal{B}_{req}}^k$, a trade-off strategy may be defined as a mechanism which generates a new $\lambda_{\mathcal{B}_{req}}^{k+1}$ such that $\alpha^{\lambda_{\mathcal{B}_{req}}^{k+1}} \geq \alpha^{\lambda_{\mathcal{B}_{req}}^k}$ and $\lambda_{\mathcal{B}_{req}}^{k+1} \neq \lambda_{\mathcal{B}_{req}}^k | \forall j < k + 1$.

DEFINITION 5. Given a $\lambda_{\mathcal{B}_{req}}^k$, a purchase requirement valuation is defined as a set $v_{\mathcal{B}_{req}} = \{v_i | v_i \in [0, 1], i \in \text{cons}(\lambda_{\mathcal{B}_{req}}^k)\}$, where v_i represents the degree of importance that the constraint i has for the buyer agent.

Let $\mathcal{S}_{req} = (S, N_s)$ be the requirement model of the seller agent, which is defined by the set of products the seller holds and the profit that the seller agent gets if the product is sold, $S = \{s_j | s_j = (p_j, u_j), p_j = (a_{j1}, \dots, a_{jn}), 0 \leq j \leq k\}$, where k is the total number of products; and a negotiation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

Copyright 2006 ACM 1-59593-303-4/06/0005 ...\$5.00.

profile $N_s = \{\psi, \beta\}$, where $\psi \in \{0, 1\}$ controls if the seller agent expresses its preferences for a specific relaxation of the buyer's demands, and $\beta \in [0, 1]$ modulates the seller's attitude regarding a purchase requirement valuation.

DEFINITION 6. A sale offer is defined as a product p_j .

DEFINITION 7. A relax requirement is defined as a set $\rho_{\mathcal{B}_{req}} = \{r_i | r_i \in [0, 1], i \in \text{cons}(\lambda_{\mathcal{B}_{req}})\}$, where r_i represents the preference for constraint i to be relaxed.

3. THE NEGOTIATION DIALOGUE

Our purchase negotiation dialogue is a four stage sequence [2]: *open dialogue* (L1-2), *negotiate* (L3-8), *confirm* (L9-10) and *close dialogue* (L11). A negotiation will involve a seller participant P_s who joins any dialogue initiated by a potential buyer P_b . These are the main locutions:

L3: willing_to_sell(P_s, P_b, p_j) indicates to the buyer P_b a willingness to sell a product p_j .

L4: desire_to_buy($P_b, P_s, \lambda_{\mathcal{B}_{req}}^k$) requests to purchase a product which satisfies $\lambda_{\mathcal{B}_{req}}^k$.

L5: prefer_to_sell($P_s, P_b, \lambda_{\mathcal{B}_{req}}^k, \rho_{\mathcal{B}_{req}}$) requests to relax $\lambda_{\mathcal{B}_{req}}^k$, and expresses which constraints are preferred to be relaxed by means of $\rho_{\mathcal{B}_{req}}$.

L6: prefer_to_buy($P_b, P_s, \lambda_{\mathcal{B}_{req}}^k, v_{\mathcal{B}_{req}}$) requests to purchase a product which satisfies $\lambda_{\mathcal{B}_{req}}^k$. A purchase requirement valuation $v_{\mathcal{B}_{req}}$ is used.

L7: refuse_to_buy(P_b, P_s, p_j) expresses a refusal to purchase a product p_j .

L8: refuse_to_sell($P_s, P_b, p_j | \lambda_{\mathcal{B}_{req}}$) expresses a refusal to sell a product p_j or products which satisfy the purchase requirement $\lambda_{\mathcal{B}_{req}}$.

L9: agree_to_buy(P_b, P_s, p_j) and

L10: agree_to_sell(P_s, P_b, p_j) express a commitment to buy or sell a product p_j .

4. THE DECISION MECHANISMS

According to the participants (Buyers **B** or Sellers **S**):

B1: Recognize Need initiates a purchase dialogue by means of a user interface. **Outputs:** $\text{have_need}(\theta)$, where θ is a category of products.

B2: Generate Purchase Requirement generates a new purchase requirement. **Outputs:** $\emptyset, \lambda_{\mathcal{B}_{req}}^k$.

There are two basic strategies: (a) add a new constraint when the first $\lambda_{\mathcal{B}_{req}}$ is generated or when a sale offer cannot be accepted, or (b) relax $\lambda_{\mathcal{B}_{req}}^k$. An empty set $\emptyset$ is generated as output if none of these strategies can be used. In case (b), if the received locution is a $\text{refuse_to_sell}()$, we define a meta strategy which consists in applying when possible the trade-off strategy, and otherwise the concession one, so the loss of posd is minimized. If the received locution is a $\text{prefer_to_sell}()$, a cooperative strategy may be applied:

DEFINITION 8. Given $\rho_{\mathcal{B}_{req}}^k$ as a response to a $\lambda_{\mathcal{B}_{req}}^k$, a cooperative strategy is a mechanism which generates a new $\lambda_{\mathcal{B}_{req}}^{k+1}$ as a function of $\rho_{\mathcal{B}_{req}}^k$ and $\lambda_{\mathcal{B}_{req}}^k$, which must integrate the demands of the seller agent.

This strategy is useful when a buyer agent has several alternatives to generate a new purchase requirement. Taking into account the demands of the seller agent, the speed of convergence and the joint utility may be increased. Obviously, both parameters must be requirements for the buyer agent. We propose the following algorithm to relax a pur-

chase requirement.

Algorithm 1 (1) Obtain the set $\{\alpha_{R_i^f}^{\lambda_{\mathcal{B}_{req}}^{k+1}} | \forall i \in \text{cons}(\lambda_{\mathcal{B}_{req}}^{k+1})\}$,

where $\alpha_{R_i^f}^{\lambda_{\mathcal{B}_{req}}^{k+1}}$ is the *posd* when R_i^f is relaxed; (2) Compute

$\alpha_{\mathcal{B}_{req} \text{ cooperative}}^{\lambda_{\mathcal{B}_{req}}^{k+1}} = \odot(\{\alpha_{R_i^f}^{\lambda_{\mathcal{B}_{req}}^{k+1}} | \forall i \in \text{cons}(\lambda_{\mathcal{B}_{req}}^{k+1})\})$, where $\odot$

is an aggregation operator; (3) Obtain $\text{feasible}\{\lambda_{\mathcal{B}_{req}}^{k+1}\}$ as the

set of elements which satisfy $\alpha_{R_i^f}^{\lambda_{\mathcal{B}_{req}}^{k+1}} \geq \alpha_{\mathcal{B}_{req} \text{ cooperative}}^{\lambda_{\mathcal{B}_{req}}^{k+1}}$; (4)

Relax the constraint obtained by means of the *csf* function:

$$\arg(\max_{\text{feasible}\{\lambda_{\mathcal{B}_{req}}^{k+1}\}} \alpha_{R_i^f}^{\lambda_{\mathcal{B}_{req}}^{k+1}} + r_i * \eta)$$

B3: Generate Purchase Requirement Valuation obtains the valuation of a $\lambda_{\mathcal{B}_{req}}$. **Outputs:** $\emptyset, v_{\mathcal{B}_{req}}$.

If $\xi = 0$ this mechanism returns an empty set $\emptyset$. Otherwise, a valuation of the purchase requirement must be obtained. A high valuation implies that the buyer agent prefers not to relax the constraint.

Algorithm 2 (1) Given a $\lambda_{\mathcal{B}_{req}}^{k+1}$ then obtain a vector with the *posd* for all the possible future purchase requirements: $\{\alpha_{R_i^f}^{\lambda_{\mathcal{B}_{req}}^{k+2}} | \forall i \in \text{cons}(\lambda_{\mathcal{B}_{req}}^{k+2})\}$; (2) Then apply the pur-

chase requirement valuation function $\text{prvf} : \{\alpha_{R_i^f}^{\lambda_{\mathcal{B}_{req}}^{k+2}} | \forall i \in \text{cons}(\lambda_{\mathcal{B}_{req}}^{k+2})\} \rightarrow v_{\mathcal{B}_{req}}$. It represents a general framework to obtain the valuation of $\lambda_{\mathcal{B}_{req}}$.

B4: Consider Offers decides at a particular whether to:

(a) accept an offer, when $\alpha(p_j) \geq \alpha_{\mathcal{B}_{req}}^k$; (b) reject an offer, when for some reason it does not fall within a previously submitted $\lambda_{\mathcal{B}_{req}}$; or (c) generate a new $\lambda_{\mathcal{B}_{req}}$ when it does not exceed the actual *posd*. **Outputs:** $\text{accept_offer}(p_j)$, $\text{reject_offer}(p_j)$, $\text{generate_purchase_requirement}(p_j)$.

B5: Consider Withdrawal generates the $\text{withdraw}(\theta)$ output. **Outputs:** $\text{withdraw}(\theta)$.

S1: Recognize Category assesses whether the seller has products to sell. **Outputs:** $\text{enter}(\theta)$, $\text{not_to_enter}(\theta)$.

S2: Assess Purchase Requirement assesses whether a proposed $\lambda_{\mathcal{B}_{req}}$ can be satisfied. **Outputs:** $\emptyset, \lambda_{\mathcal{B}_{req}}^k, p_j$.

Algorithm 3 (1) The seller agent searches for products which satisfy the purchase requirement; (2) If it cannot be satisfied and parameter $\psi = 0$, an empty set $\emptyset$ is returned; (3) If it cannot be satisfied and parameter $\psi = 1$ then $\lambda_{\mathcal{B}_{req}}^k$ is returned; (4) From the products which satisfy the buyer's requirement, choose those with the highest utility and which has been a sale offer a minimum number of times.

S3: Generate Potential Sale Offers selects products which the seller agent considers as good candidates for a future sale offer. **Outputs:** set S_p of products.

If a seller agent cannot satisfy a $\lambda_{\mathcal{B}_{req}}$, it may encourage the buyer agent to change its proposals. A good alternative to promote a convergence in the negotiation is to express how $\lambda_{\mathcal{B}_{req}}$ should be relaxed. This mechanism treats the first task in this process: the selection of products which the seller agent would prefer to sell. We have identified two main aspects when making this selection: *local utility*, which depends on the u_j parameter; and *viability*, which depends on the similarity between the product p_j and $\lambda_{\mathcal{B}_{req}}^k$, as well as on an estimate of the sale offer p_j buyer's valuation. The β parameter modulates both aspects. So we define the *prefer*

function which measures the goodness of a sale offer in terms of utility and viability as:

$$prefer(p_j) = \beta * u_j + (1 - \beta) * viability(p_j, \lambda_{B_{req}}^k, v_{B_{req}})$$

Algorithm 4 (1) Take all the products in the catalogue and compute the function *prefer*; (2) Fix a threshold and select those products with a *prefer* value exceeding this threshold. This is the set S_p .

S4: Generate Relax Requirement obtains the relax-requirement $\rho_{B_{req}}$. **Outputs:** $\rho_{B_{req}}$.

Given a set of potential sale-offers S_p , it generates a $\rho_{B_{req}}$ in order to encourage the buyer agent to buy one of the products contained in S_p .

Algorithm 5 (1) For each potential sale-offer see which constraints from $\lambda_{B_{req}}$ are not satisfied; (2) Generate $\rho_{B_{req}}$, where $r_i = 1$ when constraint R_i^f has not been satisfied for any product, and otherwise $r_i = 0$.

S5: Accept or Reject Offer returns *accept*(p_j) if the sale-offer accepted by the buyer agent still exists and otherwise *reject*(p_j). **Outputs:** *accept*(p_j), *reject*(p_j).

S6: Consider Withdrawal generates the *withdraw*(θ) output when invoked. **Outputs:** *withdraw*(θ).

5. OPERATIONAL SEMANTICS

Operational semantics provides a formal linkage between the dialogue locutions and the semantic mechanisms previously defined. We use the notation in [2] which defines $\langle P_x, K, s \rangle$ to denote the mechanism with number K and with output s of agent P_x . The transition rules are:

- TR1: $\langle P_b, B1, have_need(\theta) \rangle \xrightarrow{L1} \langle P_s, S1, . \rangle$
- TR2: $\langle P_s, S1, not_to_enter(\theta) \rangle \rightarrow \langle P_s, S1, . \rangle$
- TR3: $\langle P_s, S1, enter(\theta) \rangle \xrightarrow{L2} \langle P_b, B2, . \rangle$
- TR4: $\langle P_b, B2, \emptyset \rangle \rightarrow \langle P_b, B5, . \rangle$
- TR5: $\langle P_b, B5, withdraw(\theta) \rangle \xrightarrow{L11} \langle P_s, S6, . \rangle$
- TR6: $\langle P_s, S6, withdraw(\theta) \rangle \xrightarrow{L11} \langle P_b, B5, . \rangle$
- TR7: $\langle P_b, B2, \lambda_{B_{req}}^k \rangle \rightarrow \langle P_b, B3, . \rangle$
- TR8: $\langle P_b, B3, \emptyset \rangle \xrightarrow{L4} \langle P_s, S2, . \rangle$
- TR9: $\langle P_b, B3, v_{B_{req}} \rangle \xrightarrow{L6} \langle P_s, S2, . \rangle$
- TR10: $\langle P_s, S2, \emptyset \rangle \xrightarrow{L8} \langle P_b, B2, . \rangle$
- TR11: $\langle P_s, S2, saleoffer(p_j) \rangle \xrightarrow{L3} \langle P_b, B4, . \rangle$
- TR12: $\langle P_s, S2, purchasereq(\lambda_{B_{req}}^k) \rangle \rightarrow \langle P_s, S3, . \rangle$
- TR13: $\langle P_s, S3, S_p \rangle \rightarrow \langle P_s, S4, . \rangle$
- TR14: $\langle P_s, S4, \rho_{B_{req}} \rangle \xrightarrow{L5} \langle P_b, B2, . \rangle$
- TR15: $\langle P_b, B4, generatepurchasereq(p_j) \rangle \rightarrow \langle P_b, B2, . \rangle$
- TR16: $\langle P_b, B4, acceptoffer(p_j) \rangle \xrightarrow{L9} \langle P_s, S5, . \rangle$
- TR17: $\langle P_b, B4, rejectoffer(p_j) \rangle \xrightarrow{L7} \langle P_s, S2, . \rangle$
- TR18: $\langle P_s, S5, accept(p_j) \rangle \xrightarrow{L10} \langle P_b, B5, . \rangle$
- TR19: $\langle P_s, S5, reject(p_j) \rangle \xrightarrow{L8} \langle P_b, B2, . \rangle$

6. A PURCHASE SCENARIO

A buyer desires to buy a product θ such that: $X_\theta = \{price, brand\}$, $D_\theta = \{[0, 30000], \{A, \dots, J\}\}$, and $\oplus = min$. Figure 1 shows the buyer's preferences in a colored plane. The seller's catalogue is shown as a set of stars, circles and triangles representing products which give the seller agent a profit of 1.0, 0.5 and 0.1, respectively. The line describes the progress of the negotiation for both the inexpressive (dotted line) and expressive strategies. The progress of the negotiation in the first steps is the same in both strategies because all the potential sale offers are far from the buyer requirements, so the seller indicates the same preferences in the re-

lax requirement. However, when $\lambda_{B_{req}}$ is near to $(17000, E)$ and $(15000, D)$, the seller agent indicates that the brand constraint should be relaxed. For the buyer agent it is the same to relax the price or the brand constraint, so it relaxes the brand. The aggregated utility of the outcome is 1.4, while in the inexpressive strategy is 0.5.

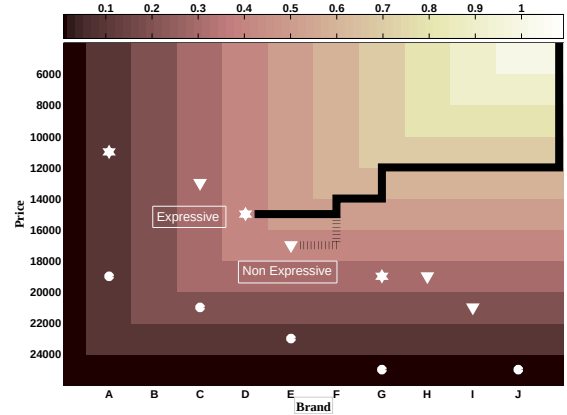


Figure 1: Expressive vs Inexpressive Strategies

7. CONCLUSIONS

This paper presents a model to support automated bilateral purchase negotiations using fuzzy constraints. All the internal decision mechanisms and the rules which govern interactions have been defined. In contrast to other fuzzy constraint models, in ours: (1) A buyer agent attends the seller's requirements in order to select the alternative from the set of trade-off proposals that is likely to benefit both agents. Typically this has been done employing an estimation of the similarity between an offer and the set of feasible counter-offers. The problem is that it is assumed that the potential offers are related among them, and this is not always the case. (2) Constraints can be valued in order to help the seller agent to make a more effective search. The purpose of this search is to select the more convenient potential sale offers in order to generate a balanced relax requirement. (3) Different attitudes for the seller and buyer agents can be easily modelled by means of the negotiation profiles.

8. ACKNOWLEDGEMENTS

This work has been supported by the Spanish Ministry of Science and Technology grant MCYT-TIC2003-09192-C11-05, and by the University of Alcalá grant UAH-PI-2003/001.

9. REFERENCES

- [1] K. R. Lai and M. W. Lin. Modeling agent negotiation via fuzzy constraints in e-business. *Computational Intelligence*, 20(4):624–642, 2004.
- [2] P. McBurney, R. M. V. Euk, S. Parsons, and L. Amgoud. A dialogue game protocol for agent purchase negotiations. *Autonomous Agents and Multi-Agent Systems*, 7:235–273, 2003.
- [3] Xudong-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.