Evolutionary Computing and Negotiating Agents

Noyda Matos, Carles Sierra

IIIA - Artificial Intelligence Research Institute
CSIC - Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia, Spain.
{noyda, sierra}@iiia.csic.es

Abstract. Automated negotiation has been of particular interest due to the relevant role that negotiation plays among trading agents. This paper presents two types of agent architecture: *Case-Based* and *Fuzzy*, to model an agent negotiation strategy. At each step of the negotiation process these architectures fix the weighted combination of tactics to employ and the parameter values related to these tactics. When an agent is provided with a Case-Based architecture, it uses previous knowledge and information of the environment state to change its negotiation behaviour. On the other hand when provided with a Fuzzy architecture it employs a set of fuzzy rules to determine the values of the parameters of the negotiation model. In this paper we propose an evolutionary approach, applying genetic algorithms over populations of agents provided with the same architecture, to determine which negotiation strategy is more successful.

1 Introduction

Negotiation in multi-agent systems is one of the main research lines in multi-agent systems [18] and has been studied from different points of view: economics [22], dialectics [27] or coordination [4]. Recent growing interest in autonomous agents and their application in areas such as electronic commerce has given more importance to the problem of automated negotiation. Agents negotiate to coordinate their activities and come to a mutual agreement. In many cases, this automated negotiation requires different behaviours for different negotiation situations.

We explore an existing model of negotiation [15] based on a set of mutually influencing two-parties, many-issues negotiation. This model relies on strategies and tactics to define the agent's negotiation behaviour (i.e. to determine which offers should be accepted and which rejected, what counter-offers should be generated and when). The model has been already applied upon a real-world business process management system [15] in which the agents need to agree as to who should perform a particular service under what terms and conditions. Since this model can operate in a wide range of environments and has a large number of parameters, Genetic Algorithms (GA) appear to be a good means of determining its performance through an empirical evaluation (see previous results in [19]).

In this paper we extend this previous work by exploring the combination of an evolutionary approach with two types of agent architecture: *Case-Based* and *Fuzzy*. Both architectures determine, at each step of the negotiation, which combination of tactics is more useful and which are the best tactic parameters. Over a family of agents with the same architecture we use an evolutionary approach to determine which individual instantiation of each type of architecture is preferred.

Case-Based Reasoning has received a lot of attention over the last years, and has been employed with good results in many areas, including negotiation [24]. We use this technique to determine the combination of tactics and the parameter values to use at each moment of the negotiation by looking at the similarity of the current negotiation to previous cases kept in a case base. The successful negotiations that an agent performs following a specific role (seller or buyer) are kept in the case base for later retrieval. The most similar case is adapted to the current situation by a set of adaptation fuzzy rules.

As an alternative route, we propose agents that use a set of fuzzy rules to model the strategy of the negotiation process for an agent. A subset of these rules determines the general behaviour of the agent, and the other adjusts the weights and the parameters of the tactics considering the information of the agent's mental state. With the combination of both kind of rules we obtain a weighted combination of tactics and their associated parameter values.

Over these architectures we make an evolutionary approach using GAs to analyse the behaviour of both kinds of agents architecture. This approach was chosen because GAs have been shown to find good solutions for problems of this nature [3]. The long term goal of this research is to study the dynamics of agents populations.

This paper presents a preliminary view of these two types of architectures and is structured as follows: Section 2 summarises the service-oriented negotiation model presented in [23]. Section 3 gives an example of scenarios in which we apply this model. Sections 4 and 5 describe the two agent architectures. Section 6 describes the general steps of the GA. Section 7 places our work in context and Section 8 outlines the avenues of further research.

2 The service-oriented negotiation model

This section outlines the main components of the service-oriented negotiation model presented in [23]. The majority of the justification for particular design choices and much of the detailed explanation of the negotiation behaviour has been omitted.

This multi-lateral negotiation model is based on a set of mutually influencing two-parties, many-issues negotiations. In service-oriented negotiations, agents can adopt two possible roles that are, in principle, in conflict. Let $Agents$ denote the set of agents in the system and let the conflicting roles be denoted by the set of agents: $Sellers$ and $Buyers$.

Negotiations can range over a number of quantitative (e.g. price, duration, and cost) and qualitative (e.g. type of reporting policy, and nature of the contract) issues. The set of issues used by each agent can change during the negotiation process, increasing or decreasing the number of issues. Quantitative issues in negotiation are defined over a continuous domain (i.e $x_j \in \mathcal{D}_j = [min_j, max_j]$). Qualitative issues are defined over an ordered domain (i.e $x_j \in \mathcal{D}_j = \langle q_1, \cdots, q_n \rangle$). When an agent receives an offer, it rates it by using a function that combines the scores of the different issues. For the purposes of this paper we can see it as a linear combination:

$$V^a(x) = \sum_{1 \leq j \leq n} w_j^a(t) V_j^a(x_j)$$

where $w_j^a(t)$ is the importance of issue j for agent a at time t[1].

Each agent has a scoring function $V_j^a : \mathcal{D}_j^a \to [0,1]$ that gives the score agent a assigns to a value of issue j in the set of its acceptable values $\mathcal{D}_j^a$. For convenience, scores are kept in the interval $[0,1]$, and scoring functions are either monotonically increasing or decreasing for quantitative issues. If the

[1] The weights associated to the issues are defined by each agent at the beginning of the negotiation, but can change later due to external causes or to the agent's interests. We assume the weights are normalised.

score of the received offer is greater than the score of the counter offer the agent would send at this point, then the offer is accepted. If the pre-established deadline (t^a_{max}) at which the negotiation must have been completed is reached, the offer is rejected. Otherwise, a counter offer is submitted.

Let $x^t_{a\to b}$ represent the vector of values proposed by agent a to agent b at time t, and $x^t_{a\to b}[j]$ be the value for issue j proposed from a to b at time t. A *Negotiation Thread* between agents $a, b \in Agents$, at time $t_n \in Time$, noted $X^{t_n}_{a\leftrightarrow b}$, is any finite sequence of length n of the form $(x^{t_1}_{a\to b}, x^{t_2}_{b\to a}, x^{t_3}_{a\to b}, \dots)$ with $t_1, t_2 \dots \le t_n$, where:

1. $t_{i+1} > t_i$, the sequence is ordered over time,
2. For each issue j, $x^i_{a\to b}[j] \in \mathcal{D}^a_j, x^{i+1}_{b\to a}[j] \in \mathcal{D}^b_j$ with $i = 1, 3, 5, \dots$, and optionally the last element of the sequence being one of the particles $\{accept, reject\}$.

$last(X^{t_n}_{a\leftrightarrow b})$ is a function returning the last element in a sequence.

Offers and counter-offers are generated by lineal combinations of functions called *tactics*. A tactic generates a value for a single negotiation issue based upon a single criterion: time remaining, resources remaining and/or the behaviour of the opponent. If multiple criteria are important in determining the value of a negotiation issue, then multiple tactics can be applied to the issues. In this case, the tactics are assigned a weight to indicate their relative importance. As the negotiation proceeds, new criteria may become relevant and the relative importance of existing criteria may vary. To reflect this fact, an agent has a *strategy* which varies the weights of the different tactics over time in response to various environmental and negotiation cues (section 2.3).

2.1 Quantitative Case

Time-dependent tactics These tactics model the fact that the agent is likely to concede quicker as the deadline for the negotiation approaches. We model the offer of agent a to agent b for issue j at time $0 \le t \le t^a_{max}$, by a function α^a depending on time.

$$x^t_{a\to b}[j] = \begin{cases} min^a_j + \alpha^a(t)(max^a_j - min^a_j) & \text{If } V^a_j \text{ is decreasing} \\ min^a_j + (1 - \alpha^a(t))(max^a_j - min^a_j) & \text{If } V^a_j \text{ is increasing} \end{cases}$$

A wide range of functions can be defined simply by varying the way in which $\alpha^a(t)$ is computed. Here we use a family of polynomial functions parameterised by a value $\beta \in \mathbb{R}^+$ that determines the convexity of the curve:

$$\alpha^a(t) = \left(\frac{min(t, t^a_{max})}{t^a_{max}}\right)^{\frac{1}{\beta}}$$

This expression represents an infinite number of possible tactics, one for each value of β. However, to better understand their behaviour, we have classified them into two sets which show clearly different patterns of behaviour: **Boulware** (don't start conceding until the deadline is nearly up) with $\beta \ll 1$, and **Conceder** (start giving ground fairly quickly) with $\beta \gg 1$.

Resource-dependent tactics These tactics generate counter-offers depending on how a particular resource is being consumed; they become progressively more conciliatory as the quantity of resource diminishes:

$$\alpha^a(t) = e^{-\frac{\mu^a}{|X^t_{a\leftrightarrow b}|}}$$

We use a measure of resource which includes both the time μ^a an agent a considers reasonable to be negotiating with another agent and the number of messages involved in the negotiation thread.

Behaviour-dependent or Imitative tactics These tactics base their actions on the behaviour of their negotiation opponent [2]. The tactics differ depending on which aspect of the opponent's behaviour they imitate and to what degree. That is:

1. **Relative Tit-For-Tat** (Relative-TFT)
   The agent reproduces, in percentage terms, the behaviour that its opponent has performed $\delta \geq 1$ steps ago:

$$x_{a\to b}^{t_{n+1}}[j] = min(max(\frac{x_{b\to a}^{t_{n-2\delta}}[j]}{x_{b\to a}^{t_{n-2\delta+2}}[j]} x_{a\to b}^{t_{n-1}}[j], min_j^a), max_j^a)$$

2. **Random Absolute Tit-For-Tat** (Random-TFT)
   The same as *Relative TFT* except that the behaviour is imitated in absolute rather than percentage terms.

$$x_{a\to b}^{t_{n+1}}[j] = min(max(x_{a\to b}^{t_{n-1}}[j] + (x_{b\to a}^{t_{n-2\delta}}[j] - x_{b\to a}^{t_{n-2\delta+2}}[j]) + (-1)^s R(M), min_j^a), max_j^a)$$

   where $s = 0$ if V_j^a is decreasing, $s = 1$ if V_j^a is increasing, M is the maximum amount by which the agent can change its imitative behaviour and $R(M)$ is a function that generates a random integer in the interval $[0, M]$.
3. **Averaged Tit-For-Tat** (Average-TFT).
   The agent uses the average of the percentage change in a window of size $\lambda \geq 1$ of its opponents history to determine its offer.

$$x_{a\to b}^{t_{n+1}}[j] = min(max(\frac{x_{b\to a}^{t_{n-2\lambda}}[j]}{x_{b\to a}^{t_n}[j]} x_{a\to b}^{t_{n-1}}[j], min_j^a), max_j^a)$$

2.2 Qualitative Case

In this work we admit only ordered qualitative variables $\mathcal{D}_j = \langle q_1, \cdots, q_n \rangle$. For this kind of variables the aforementioned tactics need some transformation. We must redefine values like min_j^a or max_j^a. When we need to use a qualitative issue's value in a tactic equation, we use instead of the qualitative value, the score related to the value. In the same way the min_j^a and max_j^a values are related to the maximum and minimum scoring of the variable. For instance for the *Relative Tit-For-Tat* tactics the transformation is the following:

$$x_{a\to b}^{t_{n+1}}[j] = V_j^{a^{(-1)}} \left(min(max(\frac{V_j^a(x_{b\to a}^{t_{n-2\delta}}[j])}{V_j^a(x_{b\to a}^{t_{n-2\delta+2}}[j])} V_j^a(x_{a\to b}^{t_{n-1}}[j]), V_j^a(min_j^a)), V_j^a(max_j^a)) \right)$$

where

$$min_j^a = \min_{q\in\mathcal{D}_j^a}\{V_j^a(q)\} \quad max_j^a = \max_{q\in\mathcal{D}_j^a}\{V_j^a(q)\}$$

We mentioned before that the scoring functions are monotonically increasing or decreasing. This restriction is necessary because we need scoring function to be invertible, that is $V_j^{a^{(-1)}}$ must be definable.

2.3 Strategies

The aim of a negotiation strategy is to determine the best course of action to reach an agreement. When agent a receives an offer from agent b, it becomes the last element in the current negotiation thread between the agents. If the offer is unsatisfactory to a, the agent a generates a counter-offer. In generating its counter-offer, a may use the information of mental state and different weighted combinations of tactics for each of the negotiation issues. More formally, an agent (a) has a representation of its mental state (MS_a^t) at time t, which contains information about its beliefs, its knowledge of the environment (time, resources, etc.), and any other attitudes (desires, goals, obligations, intentions, etc.).

Given a negotiation thread between agents a and b at time t_n, $X_{a\leftrightarrow b}^{t_n}$, over domain $\mathcal{D} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_p$, with $last(X_{a\leftrightarrow b}^{t_n}) = x_{b\to a}^{t_n}$, and a finite set of m tactics[2] $T^a = \{\tau_i | \tau_i : MS_a \to \mathcal{D}\}_{i\in[1,m]}$, a *weighted counter proposal*, $x_{a\to b}^{t_{n+1}}$, is a linear combination of the tactics given by a matrix of weights

$$\Gamma_{a\to b}^{t_{n+1}} = \begin{pmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1m} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{p1} & \gamma_{p2} & \cdots & \gamma_{pm} \end{pmatrix}$$

defined in the following way:

$$x_{a\to b}^{t_{n+1}}[j] = (\Gamma_{a\to b}^{t_{n+1}} * T^a(MS_a^{t_{n+1}}))[j,j]$$

With $(T^a(MS_a^{t_{n+1}}))[i,j] = (\tau_i(MS_a^{t_{n+1}}))[j]$, $\gamma_{ji} \in [0,1]$ and for all issues j, $\sum_{i=1}^{m} \gamma_{ji} = 1$. The weighted counter proposal extends the current negotiation thread as follows:

$$X_{a\leftrightarrow b}^{t_{n+1}} = X_{a\leftrightarrow b}^{t_n} \bullet x_{a\to b}^{t_{n+1}}$$

where $\bullet$ is the sequence concatenation operation.

Given a set of tactics, different types of negotiation behaviour can be obtained by weighting the tactics in a different way. That is, by changing the matrix Γ — particular to each negotiation thread.

A *Negotiation Strategy* for agent a is any function f such that, given a's mental state at time t_n, $MS_a^{t_n}$, and a matrix of weights at time t_n, $\Gamma_{a\to b}^{t_n}$, generates a new matrix of weights for time t_{n+1}, i.e.

$$\Gamma_{a\to b}^{t_{n+1}} = f(\Gamma_{a\to b}^{t_n}, MS_a^{t_n})$$

The main point in every step of the negotiation process is then to determine both, which weighted combination of tactics to employ and the value of the parameters associated with these tactics. Our goal is to make an evolutionary study of how two different agent architectures allow us to create a negotiation strategy and determine which strategy is more successful. In this paper we study two of these architectures: *Case-Based* and *Fuzzy*. In figure 1 their schemes are presented.

3 Negotiation in a Real Estate Agency

In this section we describe a real domain, a Real Estate Agency, that will be used to exemplify our negotiation model. In a Real Estate Agency there is a set of properties that need to be sold. In this

[2] This definition uses the natural extension of tactics to the multi-dimensional space of issues' values.

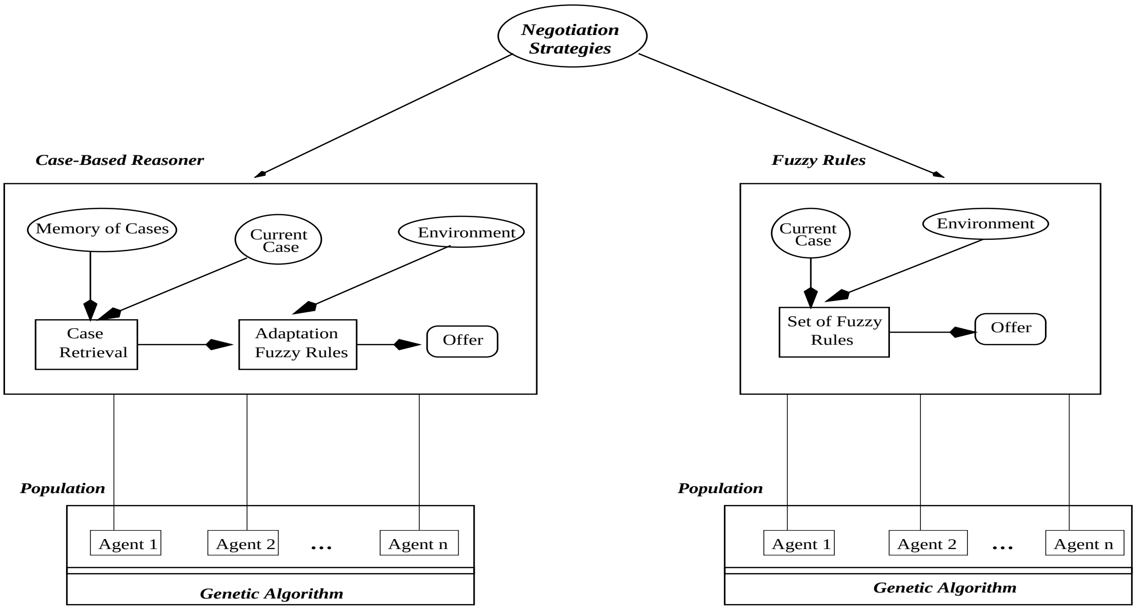

Fig. 1. Main components of the agent architecture.

domain there are two main agent roles, *seller* that represents the interests of a Real Estate Agency and *buyer* that represents the interests of a customer. Seller agents need to sell a house (preferable the one with higher profit for the agency) and buyers want to buy a house with specific features.

In this example, artificial agents represent the interests of a human buyer or seller. If a person needs to buy a house he wants to visit as few properties as possible; this is why the person might be willing to instruct an agent with a set of features that represent his desires in order to obtain a house that meets his needs. The coordination of different buyer agents negotiating at the same time with different state agencies is not treated here.

The agents so delegated negotiate over a set of issues that describe the characteristics of the house, for instance:

- *Surface*
- *District*
- *Number of rooms*
- *Floor number*
- *Garage*
- *Price*
- *Brightness*
- *Number of bathrooms*
- *Elevator*
- *Address*

The seller agent has a complete information of all the properties on sale at the Real Estate Agency. However, in some cases the buyer agent does not have a clear opinion on his preferences on the mentioned issues. The seller agent, during the negotiation, usually includes new issues to enrich the description of a house. Then, the buyer uses this new information to compare and discriminate better among the different offers made by the seller. That is, its utility function gets extended to consider the new issues introduced. In so doing, the buyer tries to obtain a complete description of the properties.

Usually the agents try to adjust either the issues related to the description of the house and later the price issue. They negotiate until they obtain an agreement, in this case a property that satisfies both sides, if any exists, or one of them withdraws.

4 Case-Based Negotiating Agents

Case-based reasoning solve a new problem by adapting a previous similar situation [1,25]. Using CBR terminology, a *case* in negotiation can be thought of as a negotiation process that has been stored in a specific way and that can be reused in solving new negotiation processes. The main tasks that a case-based negotiating agent has to deal with are the representation of the negotiation cases in its case base, the retrieval of a past case similar to the new one and the adaptation of its solution to the current negotiation process. We denote by *CB* the *Case Base*. In *CB* we only keep track of the negotiation processes that reach an agreement, that is, that have the following final thread: $X_c^{t_n} = \{x_c^{t_1}, \cdots, x_c^{t_{deal}}, accept\}$ where $t_n = t_{deal} + 1$. Now we analyse in detail the main parts of the *CBR* cycle.

4.1 Case Representation

The *Case Base* for any agent a, is composed of a set of cases that represent past negotiation processes. This *Case Base* is extended by the negotiations performed by agent a and the negotiations performed by other agents with the same role as a (seller or buyer). An agent can communicate successful cases to agents following the same role when meeting during the evolutionary computation (see section 6). The cases that belong to other agents will be considered like a negotiation performed by agent a and, to simplify, we incorporate then in the case base by substituting the name of the other agent by a. We represent a case as a vector that keeps values of the following components:

1. Problem Description
   - *Good*: the good which was negotiated.
   - t_{deal}: the time that negotiation takes.
   - t_{max}: the maximum time for the negotiation, $t_{max} \geq t_{deal}$.
   - *Init*: this value determines who began the negotiation, 1 if agent a started, 0 otherwise.
   - *Issues*: the vector of the subsequent sets of issues involved in each step of the negotiation process, i.e. $Issues = \big[[j_1, j_2, j_3], [j_1, j_3], \cdots, [j_1, j_4]\big]$. The vector is indexed by time instants, i.e. $Issues[t_1] = [j_1, j_2, j_3]$, $Issues[t_2] = [j_1, j_3]$.
   - *Weights*: the vector of the subsequent vectors of relative importance values assigned by the agent to each issue during the negotiation, i.e. $Weights = \big[[w_{j_1}, w_{j_2}, w_{j_3}], [w_{j_1}, w_{j_3}], \cdots, [w_{j_1}, w_{j_4}]\big]$.
   - *Thread*: the vector that keeps offers and counter-offers made during the negotiation.
   - *Valuation*: the relative importance of this case. This value quantifies the contributions that this case made in the resolution of other cases. The updating of this value is explained in section 4.4.
   - *Utility*: this quantity indicates how well the negotiation resulted by comparing the utility of final deal with the utility at the Nash equilibrium point [5].

$$Utility = \tanh\left(\frac{V^a(x_c^{t_{deal}})}{V^a(x_c^{Nash})}\right)$$

     where $x_c^{t_{deal}}$ is the final deal, and x_c^{Nash} represents the deal that would be made at the Nash equilibrium point. We use function tanh to keep the value of utility in the interval [0,1].

2. Solution
   - *Gammas*: sequence of matrices of weights Γ employed by agent a during the negotiation. $Gammas = (\Gamma_1, \cdots, \Gamma_p)$ where p is the number of times that agent a participates (makes offers) during the negotiation.
   - *Parameters*: the vector that keeps the value of the parameters of each one of the tactics used in each step of the negotiation.
   $Parameters = [(Par^1_{\tau_1}, Par^1_{\tau_2}, ...Par^1_{\tau_m}), \cdots, (Par^p_{\tau_1}, Par^p_{\tau_2}, ...Par^p_{\tau_m})]$
   where τ_i denote tactics from the families of tactics defined in Section 2.

Example:
Now let us see an example of case representation for a negotiation process in a Real Estate Agency. Two agents, a seller b and a buyer a negotiate during 5 units of time. In the first steps of the negotiation process the set of issues was $S_1 = \{Address,\ Surface,\ Number\_of\_Rooms,\ Brightness, Price\}$ and later the seller agent b added a new issue *Garage*, so $S_2 = \{Address, Surface,\ Number\_of\_Rooms,\ Brightness,\ Price, Garage\}$. Agent a began the negotiation process and used only 3 tactics, from the following families: *Boulware, Conceder* and *Relative TFT*. The offers and counter-offers are shown in the following negotiation thread: $X^{t_5}_{a \leftrightarrow b} = \{x^{t_1}_{a \to b}, x^{t_2}_{b \to a}, x^{t_3}_{a \to b}, x^{t_4}_{b \to a}, x^{t_5}_{a \to b}, accept\}$ where:

$$
\begin{aligned}
x^{t_1}_{a \to b} &= [?,\quad 140m^2, 4,\ Very\_Bright,\quad \pounds 400k]\\
x^{t_2}_{b \to a} &= [\#21, 60m^2,\ 4,\ Slightly\_Bright,\ \pounds 400k]\\
x^{t_3}_{a \to b} &= [?,\quad 120m^2, 4,\ Very\_Bright,\quad \pounds 400k]\\
x^{t_4}_{b \to a} &= [\#69, 120m^2, 3,\ Bright,\quad \pounds 600k, true]\\
x^{t_5}_{a \to b} &= [\#69, 120m^2, 3,\ Bright,\quad \pounds 500k, true]
\end{aligned}
$$

The representation of this negotiation process in the *Case Base* of agent a would be:

$$
\Big[\underset{Good,}{House,}\ \underset{t_{deal},}{5,}\ \underset{t_{max},}{7,}\ \underset{Init,}{1,}\ \underset{Issues,}{[S_1, S_1, S_1, S_2, S_2],}\ [\underset{Weight[t_1],}{[0.1, 0.23, 0.23, 0.13, 0.3],}\ \underset{Weight[t_2],}{=,}\ \underset{Weight[t_3],}{=,}\ \underset{Weight[t_4]}{=,}
$$

$$
\underset{Weight[t_5]}{[0.1, 0.2, 0.15, 0.1, 0.4, 0.05]}],\ \underset{Thread^*}{\left[x^{t_1}_{a \to b}, x^{t_2}_{b \to a}, x^{t_3}_{a \to b}, x^{t_4}_{b \to a}, x^{t_5}_{a \to b}, accept\right],}\ \underset{Valuation}{0.3,}\ \underset{Utility}{0.5}
$$

$$
\Big(\underset{\Gamma^1,}{[[0.4, 0.3, 0.3], [0.5, 0.2, 0.3], [0.5, 0.1, 0.4], [0.4, 0.4, 0.2], [0.7, 0.1, 0.2]],}\ \underset{\Gamma^2,}{=,}
$$

$$
\underset{\Gamma^3,}{[[0.9, 0.05, 0.05], [0.9, 0.05, 0.05], [0.9, 0.05, 0.05], [0.9, 0.05, 0.05], [0.7, 0.1, 0.2], [0.9, 0.05, 0.05]]}\Big),
$$

$$
[\underset{Par^1}{(0.02, 5, 2),}\ \underset{Par^2}{=,}\ \underset{Par^3}{=}]\ \Big]
$$

Where the sign = means that the value of the component at this position is the same as the previous one in the vector. The vector $Par^i = (0.02, 5, 2)$ contains the value of the parameters related with the tactics employed by agent a, in this case $(\beta_{boulware}, \beta_{conceder}, \delta_{relative})$. In the last step of the negotiation, agent a behaves in a more boulware way for all the issues for which it agrees on the value (agent a doesn't want to change these values) and only concentrates in adjusting the price of the house. That is shown in the value of the Γ matrix, $[0.9, 0.05, 0.05]$. The value ? for an issue j is handled like a *don't care* value which has scoring zero, $V_j^a(?) = 0$.

4.2 Case Retrieval

The case retrieval process is executed concurrently with the other activities of a case-based agent. When an agent sends an offer, it immediately begins to retrieve those cases that are more similar to the current negotiation from its *Case Base*. When it receives a counter-offer to its offer it is incorporated into the negotiation thread and used to finally select the most similar case from those that were obtained in the meantime. We denote by N the current negotiation represented in the same format used for the cases stored in the *Case Base CB*. The goal of this task is to find the case in *CB most similar* to N. We can see a graphical representation of this process in figure 2.

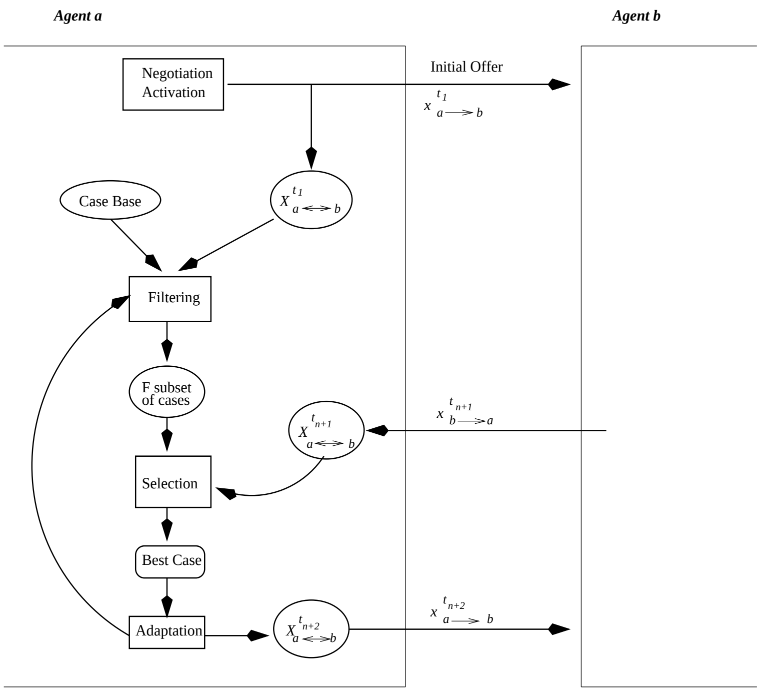

Fig. 2. Case Retrieval.

Filtering

Not all cases stored in *CB* can be compared with N for different reasons. For instance, because the set of issues considered in each step of the negotiation and the duration of the negotiation are different. This is why we build a subset F selecting from CB those cases that satisfy all the following criteria:

1. **Class of t_{deal}**. We divide CB into two classes that depend on the value of t_{deal} with respect to a cut value θ:
   - *Short* t_{deal}: $C \in F_1$ iff $C \in CB$ and $C[t_{deal}] \leq \theta$
   - *Large* t_{deal}: $C \in F_1$ iff $C \in CB$ and $C[t_{deal}] > \theta$

   The rationale here is that the negotiation strategies are radically different depending on how much time is available.
2. **Duration of negotiation.** With t being the time consumed so far in the current negotiation, we select the cases with a duration longer than the value of t, but not longer than the t_{max} defined in N. This is an important criterion, again because the negotiation strategies strongly depend on the time that the agent has to negotiate [10]. These conditions can be expressed by:
   $C \in F_2$ iff $C \in CB$ and $t < C[t_{deal}] \leq N[t_{max}]$
3. **Sets of Issues**. We select the cases from CB that contain at least the set of issues of N for each step of the negotiation, to make the comparison between them possible.
   $C \in F_3$ iff $C \in CB$ and $\forall k \leq t \quad S(N[Issues, k]) \subseteq S(C[Issues, k])$ where S is a function that transforms a vector into a set.
4. **Who began the negotiation?** Only the cases in which the agent that began the negotiation is the same as in N are considered. $C \in F_4$ iff $C \in CB$ and $C[Init] = N[Init]$.

Hence, we define F as the set of cases that satisfy all criteria, that is, $C \in F$ iff $C \in F_1 \cap F_2 \cap F_3 \cap F_4$

Final Selection

The cases that satisfied all the aforementioned criteria compose the set F from where we will select the most similar case with respect to the current negotiation. The similarity at time t between the case N and all the cases $C \in F$ is measured in the following way:

$$Sim(N, C, t) = \rho_1 * \sum_{k=1}^{t} \frac{2k}{t * (t+1)} * Sim\_Step(N, C, k) + \rho_2 * C[Valuation] + \rho_3 * C[Utility]$$

In this similarity measure we take into account three factors:

- The similarity between the offers of both cases for each step k of the negotiation ($Sim\_Step(N, C, k) \in [0, 1]$). With the factor $\frac{2k}{t*(t+1)}$ [3] we concede an increasing importance to the similarity of the offers in the last steps of the negotiation.
- The Valuation of the case, that shows how much this case has contributed to the solution of other cases ($C[Valuation] \in [0, 1]$).
- The utility obtained by this case during the negotiation ($C[Utility] \in [0, 1]$).

The factors $\rho_i \in [0, 1]$, $i = 1, 2, 3$, determine the relative relevance of the three elements mentioned in the similarity measure and satisfy: $\rho_1 + \rho_2 + \rho_3 = 1$. The similarity of cases will change by varying the value associated with each ρ_i, for example: if we give more importance to ρ_3, associated with the

[3] $\sum_{k=1}^{t} \frac{2k}{t*(t+1)} = 1$; so we employ this component as a weighting mechanism that produces a normalised output.

$C[Utility]$, we prefer cases which are better off in the negotiation, i.e. whose scoring was better or nearest to the scoring at the Nash equilibrium point.

The similarity degree between two offers takes into account the similarity between the issues of both cases and is defined as:

$$Sim\_Step(N,C,k) = \sum_{j=i_1}^{i_n} AE\left(\frac{V_j^a(N[Thread,k,j])}{V_j^a(N[Thread,k+1,j])} * \frac{V_j^a(C[Thread,k+1,j])}{V_j^a(C[Thread,k,j])}\right) *$$

$$* N[weights,k,j] * Sim\_Weight(N,C,k,j)$$

with $N[issues,k] = [i_1,\cdots,i_n]$

In this function we take into account:

- The similarity between the vector *Thread* for both cases. We consider a quotient between the scoring value for issue j in the step k and the scoring for the same issue in the step $k+1$. This quotient is computed for both cases, N and C, and then the quotient of both is the argument of the function $AE = Almost\_Equal$ that shows how close to 1 was the result. For example, $AE = \max(0, -x^2 + 2x)$.
- The weight for issue j in N at step k ($N[weights,k,j] \in [0,1]$). We give more importance to the similarity between those issues that agent a considers more relevant in the current step of negotiation.
- The similarity between the vectors of weights of the case C and N ($Sim\_Weight(N,C,k,j) \in [0,1]$). The relative importance that the agents gave to the issues under the negotiation determine in some way how similar the strategies to adopt in each step of this process can be. For this reason we consider the similarity between the vectors of weights as a factor in the overall step similarity. We prefer cases with a similar importance to the issues. The similarity between the vectors of weight is calculated in the following way:

$$Sim\_Weight(N,C,k,j) = \frac{\left| N[weights,k,j] - \frac{C[weights,k,j]}{\aleph(N,C,k)} \right|}{card(N[issues,k])}$$

$$\aleph(N,C,k) = \sum_{j=i_1}^{i_n} C[weights,k,j]$$

with $N[issues,k] = [i_1,\cdots,i_n]$

The sets of issues for both cases are not necessarily the same. We compute the similarity related with the set of issues involved in the case N. For that reason we must normalise the weights of that set of issues for the case C, that is the factor $\aleph(N,C,k)$.

Incremental Computation of Similarity

We use an incremental way of computing the similarity between cases in F and the current case N. This is done for efficiency reasons. Immediately after agent a sends an offer at time t, it begins to compute a provisional similarity ($Sim(N,C,t)$) to profit from the waiting time. When the counter part answers, the agent incorporates the new information, the offer just received, by actualising its thread and adjusting the similarity of the cases stored in F with the objective of selecting *the most similar*

case.
The similarity between N and all the cases $C \in F$ is then computed in the following way:

$$\begin{aligned} Similarity(N, C, t+1) &= \rho_1 * Sim(N, C, t+1) + \rho_2 * C[Valuation] + \rho_3 * C[Utility] \\ Sim(N, C, 0) &= 0 \\ Sim(N, C, t+1) &= \frac{t}{t+2} * Sim(N, C, t) + \frac{2}{t+2} * Sim\_Step(N, C, t+1) \end{aligned}$$

Where $Sim\_Step(N, C, t)$ is as mentioned before. The value $Sim(N, C, t)$ is kept by each agent and is used in the next step $t+1$ to avoid calculate the same quantity.

4.3 Adaptation

The case with the highest similarity contains the vector of $Parameters_{t+1}$ and the Γ_{t+1} matrix; these elements are to be used in the next step of the negotiation process. However, this solution can be improved by adapting the values of the $Parameters_{t+1}$ and the Γ_{t+1} to the current mental state (MS_a^{t+1}). This adaptation process is modelled by a set of *Adaptation Fuzzy Rules*. These rules represent conditions of the environment in which the agent acts and determine variations in the value of the parameters of the tactics and the γ_{ij}'s. In general these rules follow the following pattern:

$RULE_i$: **IF** x_1 is A_{i1} and ... and x_n is A_{in} **THEN** y is B_i

where $x_1, \cdots, x_n$ and y are the mental state variables and $A_{i1}, \cdots, A_{in}, B_i$ are linguistic labels of the variables $x_1, \cdots, x_n, y$ in the universe of discourse $U_1, \cdots, U_n, V$ of the variables. An example of linguistic labels could be: $\{high, medium, low\}$. These linguistic labels are characterised by their membership functions $A_{ij} : U_j \to [0, 1]$, $j = 1, \cdots, n$; $B_i : V \to [0, 1]$ and have a trapezoidal shape. For more details see [26].

In general the rule conditions express a state of the environment and the negotiation tactics. The rule consequents change the weighted combinations and the tactic parameters. A variation in a γ_{ij} generates automatically a normalisation of the rest of gammas.

Example: In the domain of the Real Estate Agency, we could have the following fuzzy rules:

IF *highway is near* and $\gamma_{price,boulware}$ *is medium* THEN $\triangle\beta_{boulware}$ *is negative_small*
IF *park is near* and $\gamma_{price,boulware}$ *is high* THEN $\triangle\gamma_{price,conceder}$ *is positive_high*

The linguistic labels *positive_high* represents a big positive number and *negative_small* a small negative number. If the first rule applies, the agent will behave in a more boulware fashion in all issues by decreasing its $\beta_{boulware}$ value. On the contrary (second rule) if a park is near the house and the agent's attitude to the price is mainly boulware, it will become more conceder, i.e. the $\gamma_{price,conceder}$ related to the conceder tactic increases the value and via the normalisation process $\gamma_{price,boulware}$ will decrease.

4.4 Retain

If the result of the negotiation is satisfactory, i.e. $last(X_N^{t_n}) = accept$, the new case is stored in CB. Moreover, the *Valuation* of all $C \in CB$ is updated by assigning a positive reinforcement to the cases that contributed to steps of the solution ($r = 0$) and negative reinforcement if not ($r = 1$):

$$C[\mathit{Valuation}] = C[\mathit{Valuation}] + \delta * (1 - C[\mathit{Valuation}])$$
$$\delta = k * (-1)^r * \tanh\left(\frac{V^a(x_N^{t_{deal}})}{V^a(x_N^{Nash})}\right)$$

where $\delta \in [0, 1]$, $x_N^{t_{deal}}$ is the final deal, and x_N^{Nash} represents the deal that would be made at the Nash equilibrium point and $k = 0.001$ is a reduction factor. Like explained before the function tanh keeps the value of the quotient in $[0, 1]$.

5 Fuzzy Rules

Systems based on fuzzy rules [8, 7] have proved to be an important tool to model complex systems. We had defined a type of agents that use a family of fuzzy rules to model the negotiation strategy. The set of fuzzy rules determines, at each step, the parameters of the tactics involved in the negotiation step and the weighted combination of these tactics.

Each agent interacts in a specific domain; this is why they must represent this domain in a way that permits the interaction with other agents in the same domain. In other words, they need to use an ontology [12] that permits the representation of objects, concepts that exist in that domain and also relations between them. We assume, for the purpose of using GA as will be seen later in section 6, that this ontology is common, and that the fuzzy rules are built on top of it using the same syntax for all agents.

In general we can divide the set of fuzzy rules in two classes:

- *Rules of general behaviour.* These rules define the general conduct of the agent. They determine how an agent reacts during a negotiation, i.e. with a more boulware, conceder or imitative behaviour.
- *Rules of the Mental State.* These rules analyse the environment to adjust the parameters and the weighted combination of tactics. The mental state rules have a higher priority because are more specific, that is, more informed.

Similar to the adaptation fuzzy rules of the case-based agents, the rules follow the following pattern:

$RULE_i$**:** **IF** x_1 is A_{i1} and ... and x_n is A_{in} **THEN** y is B_i

where $x_1, \cdots, x_n$ and y are the mental state variables and $A_{i1}, \cdots, A_{in}, B_i$ are linguistic labels of the variables $x_1, \cdots, x_n, y$ in the universe of discourse $U_1, \cdots, U_n, V$ of the variables. These linguistic labels are characterised by their membership functions $A_{ij} : U_j \to [0, 1]$, $j = 1, \cdots, n$; $B_i : V \to [0, 1]$ and have a trapezoidal shape.

Example: In the domain of the Real Estate Agency we could have the following rules:

1. *Rules of general behaviour.* Like in the adaptation fuzzy rules of the *Case Base* architecture, these rules change the value of the $\gamma_{i,j}$'s and the value of the parameters associated with each tactic. However, these rules take into account other aspects such as the general conduct of the agent. For instance, the following rules describe the behaviour (the dominant tactic) depending on the current of negotiation stance and the behaviour of the opponent.

– IF *remaining_time is medium* THEN $\triangle\gamma_{j,tft}$ *is positive_high*

If the remaining time that the agent has to negotiate is medium then the possibilities to interact with the contrary are bigger. Under these conditions the imitative tactics have more opportunities to assess their opponents' behaviour and to respond appropriately, hence we increment the value of the $\gamma_{j,tft}$ for all issues j.

– IF *average(last_opponent_offers) is high_conceder* and $\gamma_{j,conceder}$ *is high* THEN $\triangle\beta_{conceder}$ *is negative_small*

The agent analyses the last offer received and if the opponent, in average, has conceded, and if the conceder tactic is the domain tactic for all issues j, then the agent carries out a small decrease in its conceder behaviour to try to exploit the opponent.

2. *Rules of the Mental State.* These rules cause variations on the weighted combinations and the parameter values depending on the knowledge of the agent about the negotiation context.

– IF *agency is going_bankrupt* and $\gamma_{price,conceder}$ *is medium* THEN $\triangle\gamma_{price,boulware}$ *is positive_high*

If the Real Estate Agency is going bankrupt and we are not very tough in negotiating price, we change it by making the agent behave in a more boulware way, because it has, in such conditions, a higher opportunity of getting a lower price for a property.

– IF *time_on_sell is long* and $\gamma_{j,tft}$ *is high* THEN $\triangle\gamma_{j,boulware}$ *is positive_high*

If the Real Estate Agency has a property that wasn't sold for a long period we can obtain more profit by not conceding in the same way as the agency, because if the house has not been sold for a long time, means probably that its price is far beyond the reasonable one. In other words we should not imitate the conceding steps of the agency in the same amount, hence we increase our boulware behaviour for all issues.

– IF w_{bright} *is high* and *house is bright* and $\gamma_{surface,boulware}$ *is medium* THEN $\beta_{boulware}$=1

The agent behaves less boulware if it gives a lot of importance to brightness and it is a feature of the house being offered.

The handling of this set of rules is done by the following steps (see Figure 3):

1. *Fuzzification.* The membership functions of the input variables about the current negotiation thread and the weighted tactics are applied to their actual values to determine the truth degree of each condition.
2. *Rule Evaluation.* In this process the truth value of the premise of each rule is propagated to the conclusions. The result of this process is the assignment of a fuzzy subset to some output variables.
3. *Defuzzification.* Our goal is to determine a new matrix Γ and values for the parameters associated to each tactic, for this reason we need to obtain an exact value for these variables. There are several defuzzification methods to obtain a crisp value from a fuzzy subset and its associated truth degree, but we employ the centroid method, where the crisp value of the output variables corresponds to the value of the centre of gravity of the membership function of the fuzzy value [26].

4. *Normalisation.* Finally, a normalisation of the values obtained by defuzzification is done.

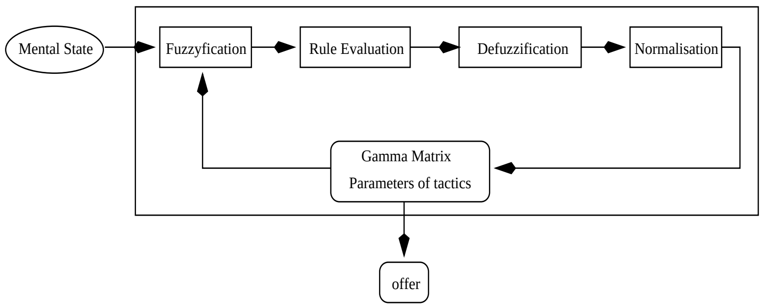

Fig. 3. Main Components of a Fuzzy Agent Architecture.

6 Evolving the negotiation strategies

GAs generate a sequence of ever improving ("fitter") populations as the outcome of a search method modelled by a *selection mechanism*, *crossover* (recombining existing genetic material in new ways) and *mutation* (introducing new genetic material into the population by random modifications) [11]. In our case we employ the GA to make an evolutionary approach of the *Fuzzy Rules*, of fuzzy agents, and the *Case Base* and the *Adaptation Fuzzy Rules* associated with Cased-Based agents. We form a population, where each agent becomes an individual, and model as genetic material the different sets of rules and/or the case base. The overall aim of the search in all cases is to find a set of rules and a base of cases which are optimally adapted for particular negotiation situations.

Even though we have two kinds of populations to analyse, *Fuzzy* and *Case-Based*, (see Figure 4) the general steps for the GA are the same. In this section we develop its basic schema —indicating how rules and cases are encoded as genes (section 6.1), how the fitness of the agents is computed (section 6.2), and how the search algorithm operates in detail (section 6.3).

6.1 Codification

Fuzzy.
Every fuzzy agent has a set of *Fuzzy Rules*. Each individual in a population represents an agent, and the genes that conform it are the codification of the linguistic labels involved in the set of rules (the ontology of the agent) followed by the codification of its fuzzy rules (see Figure 5). Linguistic labels have the form of trapezoidal-shaped functions, and for each one we represent the 4-tuple (a_j, b_j, c_j, d_j) that describes its shape, that is, $A_{ij}, B_i \in \{A_1, \cdots, A_p\}$ where p is the maximum number of linguistic labels on the ontology, and $A_j = (a_j, b_j, c_j, d_j)$ $1 \leq j \leq p$.

The dimension of the set of fuzzy rules differs from one agent to another. We consider the length of the individual's genetic code to be equal to the cardinality of the biggest set of rules in the initial

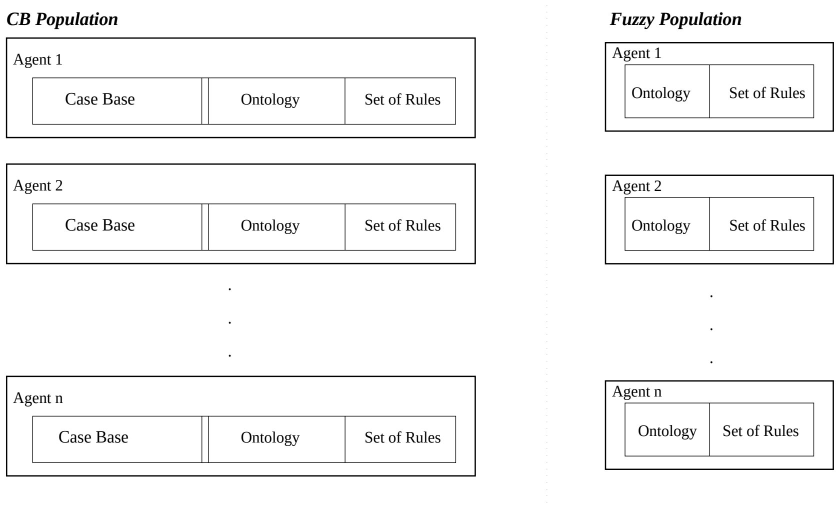

Fig. 4. Types of Populations

population and denote it by

$$r = \max_{agent_i \in P^0} \{card(FR_{agent_i})\}$$

If the number of rules of an agent is smaller than r the remaining positions of the vector codifying them are filled in with zeros.

$$
\begin{array}{c}
\quad\quad\quad\quad \mathit{Ontology} \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \mathit{Rules} \\
\begin{array}{|c|c|c|c||c|c|c|c||c||c|c|c|c||c|c|c|c|c|c|c||c||c|c|c|c|c|c|c|}
\hline
a_1 & b_1 & c_1 & d_1 & a_2 & b_2 & c_2 & d_2 & \cdots & a_p & b_p & c_p & d_p & x_1^1 & A_{11} & x_2^1 & A_{12} & \cdots & y^1 & B_1 & \cdots & x_1^r & A_{r1} & x_2^r & A_{r2} & \cdots & y^r & B_r \\
\hline
\end{array} \\
A_1 \quad\quad\quad A_2 \quad\quad\quad\quad A_p \quad\quad\quad\quad\quad\quad R_1 \quad\quad\quad\quad\quad\quad\quad R_r
\end{array}
$$

Fig. 5. Fuzzy Agent Codification.

Case-Based.

Case-based agents have a *Case Base* and a set of *Adaptation Fuzzy Rules*; both components evolve in the genetic process. Each individual in the population represents an agent and the genes are past cases stored in CB and the adaptation fuzzy rules codified in the same way as mentioned before.

For the *Case Base* we have fixed a maximum number of cases (M_Cases). We controlled the number of cases in CB selecting those cases with the best *Valuation*. Cases stored in the *Case Base* are genes of the individuals in the population. The dimension of the *Case Base* differs between agents. We consider the length of the individuals' genetic code to be equal to the cardinality of the biggest *Case Base* in the initial population

$$m_c = \min\left(M\_Cases, \max_{agent_i \in P^0}\{card(CB_{agent_i})\}\right)$$

If the number of cases of an agent is smaller than m_c the remaining positions of the vector codifying them are filled in with zeros.

$Cases \qquad\qquad\qquad Ontology \qquad\qquad\qquad\qquad\qquad Rules$

$$\boxed{c_1}\boxed{c_2}\boxed{\cdots}\boxed{c_{m_c}}\,\boxed{a_1}\boxed{b_1}\boxed{c_1}\boxed{d_1}\boxed{\cdots}\boxed{a_p}\boxed{b_p}\boxed{c_p}\boxed{d_p}\,\boxed{x_1^1}\boxed{A_{11}}\boxed{x_2^1}\boxed{A_{12}}\boxed{\cdots}\boxed{y^1}\boxed{B_1}\boxed{\cdots}\boxed{x_1^r}\boxed{A_{r1}}\boxed{x_2^r}\boxed{A_{r2}}\boxed{y^r}\boxed{B_r}$$

$A_1 \qquad\qquad A_p \qquad\qquad\qquad R_1 \qquad\qquad\qquad R_r$

Fig. 6. Case-Based Agent Codification.

6.2 Measuring a strategy's fitness

An agent's fitness value indicates how well it performs in comparison to others in the same population. Following basic evolutionary ideas, fitness also determines the agent's chance of surviving to the next population generation —the higher the value, the more likely the agent will be to reproduce. To compute an agent's fitness we play a round-robin tournament in which each buyer $b \in Buyers$ negotiates with each seller $s \in Sellers$. Then, we score each agent with a value that measures how well it performed and this score becomes the agent's fitness.

The fitness function compares the utility associated with the deal and the utility associated with the Nash equilibrium point (the point at which the sellers' and the buyers' scoring functions are equal [5]). The more positive the difference, the more successful the agent's behaviour. The cost involved in attaining a deal is taken into consideration by associating a charge with each message interchange —the more messages exchanged in coming to a deal, the higher the associated cost. With these two components in place, a non-subjective, cost-adjusted fitness function (f_a) can be defined:

$$f_a(X_c^{t_n}) = \begin{cases} V^a(x_c^{t_{deal}}) - V^a(x_c^{Nash}) - \tau(X_c^{t_n}) & \text{if } last(X_c^{t_n}) = accept \\ -V^a(x_c^{Nash}) - \tau(X_c^{t_n}) & \text{otherwise} \end{cases} \tag{1}$$

where $x_c^{t_{deal}}$ is the final deal, and x_c^{Nash} represents the deal that would be made at the Nash equilibrium point. Communication cost is modelled as: $\tau(X_c^{t_n}) = q * |X_c^{t_n}|$ where q is a constant. Varying the value of q allows us to change the relative importance of the communication cost in the fitness computation.

6.3 Algorithm steps

The GA 's basic operation can be characterised by the following schema:

P^0 population given by an expert;
While not(Stopping Criterion) **do**
  Make the population negotiate;

Calculate the fitness of all the individuals in the current P^t population;
Mating pool $MP = Tournament\_Selection(P^t)$;
Best strategies $Best_{MP} = Best\_Individuals(MP)$;
Remaining strategies $R = Crossover\&Mutation(MP - Best_{MP})$;
$P^{t+1} = Best_{MP} + R$
end while

1. **Generation of the first population**. The initial population represents the search's starting point and it is created by taking the set of fuzzy rules or adaptation fuzzy rules defined by the expert and a case base for each agent.
2. **Selection Process**. All GAs use some form of mechanism to chose which individuals from the current population should go into the mating pool that forms the basis of the next population generation. To be effective, the selection mechanism should ensure that as diverse a range of fit agents make it into the mating pool as possible (especially in the early stages). A selection mechanism which is known to work well in such circumstances is *Tournament Selection* [6]. For this reason, it is the mechanism we employ to select from P^t those individuals that will reproduce. Tournament selection works in the following way: t individuals are randomly chosen from the population. The individual with the highest fitness among the selected t is placed in the mating pool. This process is repeated N times, where N is the size of the population. t is called the tournament size and it determines the degree to which the best individuals are favoured [20]. Once the mating pool has been created, H individuals with the highest fitness in the pool are selected. These individuals will definitely form part of the new population. The rest of individuals in the next population, R, are created by applying crossover and mutation to the remaining individuals in the mating pool. Thus, the next generation (P^{t+1}) is composed of the H best individuals of the old population plus a number of newly created ones.
3. **Crossover Process**. This mechanism exchanges genetic material between individuals. We randomly select two individuals from the population. c crossover points are then randomly chosen and sorted in ascending order. Then, the genes between successive crossover points are alternately exchanged between the individuals. For the characteristics of our codification, for fuzzy agents we set two crossover regions, one where the ontology of the agent is codified and the other where the fuzzy rules are codified (in Figure 5 the two regions are separated by |||). We give a different crossover probability to each region, that is $P_{ontology}$ and P_{rules}. We made this distinction because we want to make less variations (exchange) in the ontology of the agents, $P_{ontology} < P_{rules}$. Similarly for the case-based agents we define P_{cases}, $P_{ontology}$ and P_{rules}. These regions are separated by ||| (see Figure 6).
4. **Mutation Process**. Mutation is another technique to create individuals in new generations. For the *Case Base* we do not apply mutation because any variation in a component of a case will most probably produce an undesirable strategy.
   In the case of fuzzy rules, the genes related with the linguistic variables of an individual, that is, the ontology, are given a chance P_m of undergoing mutation. When a gene is selected for mutation we change the scaling factor of one linguistic label involved in the set of rules. This kind of variation affects directly all the rules that contain that particular linguistic label as part of a condition or in the conclusion. Thereby we consider a low probability of mutation.
5. **Stopping Criterion**. The simulations stop when the population is stable or the number of iterations is bigger than a pre-determined maximum.

The concrete values for the probability of mutation and crossover are not mentioned in this paper because we are still at the experimental phase and we have not completely adjusted these values.

7 Related Work

In this paper we mention several topics from different research areas like negotiation, case-based reasoning, fuzzy logic and genetic algorithms. A complete review of all related literature is not possible here. Many works have been developed in each one of these areas, but up to our knowledge none combines these techniques to obtain a better performance for an agent during a negotiation process.

Research in negotiation models have been pursued in different fields like, game theory, social sciences and artificial intelligence, making assumptions that were pertinent for the objective of their study. Our interests lay in the study of the negotiation process among the agents and not on the outcome. In this work we employ a negotiation model and we concentrate on how can we find the best strategies in that model. For this propose we used case-based reasoning and fuzzy rules to model strategies.

In the area of case-based reasoning there are several works related with negotiation. For instance, Sycara [24] presented a model of negotiation that combines case-based reasoning and optimisation of the multi-attribute utilities of the agents. She provided a model of goal conflict resolution through negotiation implemented in the PERSUADER system, a program that resolves labour disputes, and tested her system using simulations of such domains. Our approach differs from this previous work in that we employ case-based reasoning to determine in each step of the negotiation the best performance of the agent by selecting the weighted proposal combinations and the parameters associated with a set of tactics. We also make to evolve the case base and the adaptation fuzzy rules trying to find the best performance. With this evolutionary process the agents interchange cases and rules to find an optimal set of cases and rules that allow to obtain a better scoring during the negotiations.

Many researchers have explored the use of genetic algorithms to tune fuzzy logic controllers. One of the pioneers on this was C. Karr [16], who used GAs to modify the membership function in the term sets of the variables used by a fuzzy controller. In the same line Herrera, Lozano and Verdegay [14], used a real encoding for a four-parameter characterisation of a trapezoidal membership value in each term set. Each rule used by the fuzzy control was represented by the concatenation of the membership values. The population was the concatenation of all rules so represented. In our case we use GAs to make evolve a set of fuzzy rules that model negotiation strategies in one case or that adapt the solution obtained by a case-based agent. We make genetic evolution over the linguistic labels (the ontology) of the agent and also, during the genetic process, the agents interchange rules to find an optimal performance.

Others researchers have also attempted to use GAs and co-evolutionary programming to find optimal interaction strategies. For example, Oliver [21] does consider negotiation strategies where each negotiating agent is a chromosome and the parameters of the negotiation model are genes in the chromosome. However, his negotiation model is much simpler. (Offers are accepted if they have a utility over a preset threshold and he encodes counter-offers as part of the genetic material which means they have limited sophistication.) The extra complexity required by our model means additional work was needed when designing the population evolution mechanism.

8 Conclusions and Further Work

In this paper we presented the main technical ideas of our current work on automated negotiation. We presented two agent architectures for negotiation, *Case-Based* and *Fuzzy*. Three basic techniques: *Case-Based Reasoning, Fuzzy Sets and Genetic Algorithms* are used as the basis to make an evolutionary analysis of negotiation strategies. Case-based agents, in each step of the negotiation, use it past information (case base) to retrieve a similar case to the current negotiation and adapt it by means of fuzzy rules. Fuzzy agents conduct their behaviour by means of a set of fuzzy rules. Over both architectures we make an evolutionary study applying GA, to determine the best strategy for the negotiation model mentioned in this paper.

Many aspects of these architectures need to be studied and adjusted during the experimental process that is currently ongoing. We think that this combination technique is a step forward the design of flexible and accountable trading agents.

9 Acknowledgements

The research reported in this paper is supported by the ESPRIT LTR 25500-COMRIS *Co-Habited Mixed-Reality Information Spaces* project and by the Agencia Española de Cooperación Internacional by means of Ph.D grant.

References

1. Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications. IOS Press.*, 7(1):39–59, 1994.
2. R. Axelrod. *The Evolution of Cooperation.* Basic Books, Inc., Publishers, New York, USA., 1984.
3. R. Axelrod. *The Complexity of Cooperation.* Princeton University Press., 1997.
4. Mihai Barbuceanu and Mark S. Fox. Cool: A language for describing coordination in multi agent systems. 1995.
5. Ken Binmore. *Fun and Games. A text on Game Theory.* Heath, 1992.
6. T. Blickle and L. Thiele. A comparison of selection schemes used in genetic algorithms. Technical Report 11, TIK, May 1995.
7. Piero P. Bonissone, Pratap S. Khedkar, and Yuto Chen. Genetic algorithms for automated tuning of fuzzy controllers: A transportation application. *IEEE*, pages 674–680, 1996.
8. Didier Dubois and Henri Prade. What are fuzzy rules and how to use them. *Fuzzy set and systems*, pages 169–185, 1996.
9. Frances Esteva, Pere Garcia, Lluis Godo, and Ramon Lopez de Mantaras. Fuzzy set modelling in case-based reasoning. *International Journal of Intelligent Systems*, 1997.
10. Peyman Faratin, Carles Sierra, and Nick R. Jennings. Strategic and tactical interactions between autonomous service-oriented agents. Technical Report Research Report IIIA-97/20, IIIA, 1997.
11. D. Goldberg. *Genetic Algorithms in Search, Optimisation and Matching Learning.* Addison-Wesley Publishing Co, 1989.
12. Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Formal Ontology in Conceptual Analysis and Knowledge Representation.*, 1993.
13. Thomas Haynes, Roger Wainwright, and Sandip Sen. Strongly typed genetic programming in evolving cooperation strategies.
14. F. Herrera, M. Lozano, and J.L. Verdegay. Tuning fuzzy logic controllers by genetic algorithms. Technical Report Technical Report DECSAI-93102, DECSAI, 1995.
15. N. R. Jennings, P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien, and M. E. Wiegand. Agent-based business process management. *Int Journal of Cooperative Information Systems*, 5(2–3):105–130, 1996.
16. C.L. Karr. Genetic algorithms for fuzzy controllers. *AI Expert*, pages 27–33, 1991.
17. John Koza. Evolution of emergent cooperative behaviour using genetic programming. *Computing with Biological Metaphors*, 1992.
18. S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, (94):79–97, 1997.

19. Noyda Matos, Carles Sierra, and Nick R. Jennings. Determining successful negotiation strategies: An evolutionary approach. In *(ICMAS-98)*, in press.
20. B. L. Miller and D. E. Golberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9(3):193–212, 1995.
21. Jim R. Oliver. On artificial agents for negotiation in electronic commerce. In *(http:// opim. wharton. upenn. edu/ oliver27/)*, 1994.
22. J. S. Rosenschein and G. Zlotkin. *Rules of Encounter.* MIT Press, Cambridge, ma edition, 1994.
23. Carles Sierra, Peyman Faratin, and Nick R. Jennings. A service-oriented negotiation model between autonomous agents. In *(MAAMAW-96)*, 1997.
24. Katia Sycara. Resolving goal conflicts via negotiation. In *In Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 245–250, St. Paul. MN., 1988. AAAI Press/The MIT Press.
25. Ian Watson. *Applying Case-Based Reasoning: Techniques for Enterprise Systems.* Morgan Kaufmann Publishers, inc., 1997.
26. Ronald R. Yager and Dimitar P. Filev. *Essentials of Fuzzy Modelling and Control.* Wiley-Inter-science, New York, USA., 1994.
27. D. Zeng and K. Sycara. How can an agent learn to negotiate. In M. Wooldridge J. Muller, editor, *Intelligent Agents III. Agents Theories, Architectures and Languages*, number 1193 in LNAI, pages 233–244. Spring Verlag, 1997.