Using Similarity Criteria to Make Negotiation Trade-Offs

P. Faratin*
Dept. Electronic Engineering,
Queen Mary & Westfield College,
London E1 4NS, UK.
P.Faratin@qmw.ac.uk

C. Sierra
IIIA-CSIC,
Campus UAB, Bellaterra,
08193 Barcelona, Spain.
sierra@iiia.csic.es

N. R. Jennings
Dept. Computer Science
University of Southampton,
Southampton SO17 1BJ, UK.
nrj@ecs.soton.ac.uk

Abstract

This paper addresses the issues involved in software agents making trade-offs during automated negotiations in which they have information uncertainty and resource limitations. In particular, the importance of being able to make trade-offs in real-world applications is highlighted and a novel algorithm for performing trade-offs for multi-dimensional goods is developed. The algorithm uses the notion of fuzzy similarity in order to find negotiation solutions that are beneficial to both parties. Empirical results indicate the benefits and effectiveness of the trade-off algorithm in a range of negotiation situations.

1. Introduction

Negotiation is a key form of interaction in multi-agent systems. It is important precisely because the agents are autonomous; that is, they decide for themselves what actions they should perform, at what time, and under what terms and conditions. Since such agents have no direct control over one another, they must negotiate in order to manage their interdependencies. Thus, we view negotiation as *a process by which a joint decision is made by two or more parties. The parties first verbalise contradictory demands and then move towards agreements* [9].

Automated negotiation exists in many shapes and forms: ranging from simple auctions in which agents merely have to bid truthfully [17], to complex strategic models in which agents argue for positions and aim to persuade their opponents of the value of a particular course of action [8]. In this work, however, we are interested in a particular class of negotiation: *service-oriented negotiation* [14]. In such negotiations, a producer and a consumer have to come to a mutually acceptable agreement over the terms and conditions under which the producer will execute some problem solving activity for the consumer. Specific issues that need to be agreed include the price of the service, the time at which it is required, the quality of the delivered service and the penalty to be paid for reneging upon the agreement.

In previous work [4], we investigated the design of reasoning mechanisms that enable agents to act competitively (obtain deals that are good for themselves) in service-oriented negotiations in which they have limited knowledge and computational resources. However, based on our experiences in the domains of business process management [6] and telecommunications network management [3], we find that in some cases there is also a need for agents to act in a more socially responsible manner. Thus, some of the agents in the business process management application are part of the same overarching organisation and some of the agents in the telecommunications application are from the same network operator. In such cases, the agents are concerned both with the outcome of the negotiation for themselves and for their negotiation opponent. In short, they care about equity and social welfare [2], as well as their individual utility. This requirement leads us to consider designing models that can uncover *win-win* negotiation solutions [10], again in the presence of limited knowledge and computational boundedness. Win-win negotiation refers to bargaining situations where both parties search for solutions that "squeeze out" more gains (either mutually or individually) than the currently agreed deal.

The particular mechanism for win-win negotiation that we explore here is that of agents making *trade-offs*. Intuitively, a trade-off is where one party lowers its scores on some negotiation issues and simultaneously demands more on others. Thus, an agent may accept a service of lower quality if it is cheaper or a shorter deadline if it receives a higher price. Such movements are intended to generate an offer that, although of the same value to the proposer, may benefit the negotiation opponent and hence increase the overall *joint gains* [10] between the two agents.

The contribution of this work is twofold. Firstly, extant work on automated negotiation has largely ignored the issue

*On leave at IIIA-CSIC.

of making trade-offs. Even when it has been dealt with, the advocated approach is based on assumptions that are unrealistic for real-world settings (section 4). Secondly, we present a novel algorithm for making trade-offs, in the presence of information uncertainty and resource boundedness, for multi-dimensional goods based upon the notion of fuzzy similarity [18]. Moreover, this algorithm is analysed theoretically (to determine its complexity) and evaluated empirically (to ascertain its operational performance).

The remainder of the paper is structured in the following manner. Section 2 presents our algorithm for making trade-offs in service-oriented negotiations (including the complexity analysis). Section 3 provides an empirical evaluation of our trade-off mechanism. Section 4 compares our approach to previous work on automated negotiation and section 5 outlines our conclusions and future work.

2. Making Trade-Offs

In our previous mechanism [4], agents proposed a series of contracts that had diminishing value to themselves (here we term such mechanisms *responsive*). However, in choosing to make a trade-off negotiation action, an agent is seeking to find a contract that has the same value to itself as its previous proposal, but which is more acceptable to (has higher value for) its negotiation opponent. When doing this, the agent would like to know its opponent's utility function in order to find the counter proposal that maximises the opponent's return. However, in our scenarios, this function is private and so a similarity function is used as an approximation.

2.1. Basics of Service-Oriented Negotiation

This sub-section outlines the basics of our service-oriented model (refer to [4] for more details). Let i ($i \in \{a, b\}$) represent the negotiating agents and j ($j \in \{1, ..., n\}$) be the issues under negotiation (eg price, delivery time, quality of service and penalty). Further, let $x_j^i \in [min_j^i, max_j^i]$ be a value for issue j that is acceptable to agent i. We limit ourselves to considering issues for which negotiation amounts to determining a value between an agent's defined delimited range. Each agent has a scoring function $V_j^i : [min_j^i, max_j^i] \rightarrow [0, 1]$ that gives the score agent i assigns to a value of issue j in the range of its acceptable values. For convenience, scores are kept in the interval $[0, 1]$. The relative importance that an agent assigns to each issue under negotiation is modelled as a weight, w_j^i, that gives the importance of issue j for agent i. We assume the weights of both agents are normalized, i.e. $\sum_{1 \leq j \leq n} w_j^i = 1$, for all i in $\{a, b\}$. An agent's scoring function for a *contract*—that is, for a value $x = (x_1, ..., x_n)$ in the multi-dimensional space defined by the issues' value ranges is then defined as: $V^i(x) = \sum_{1 \leq j \leq n} w_j^i V_j^i(x_j)$

For analytical purposes we restrict ourselves to an additive and monotonically increasing or decreasing value scoring system.

2.2. Formulating Trade-Offs

An agent will decide to make a trade-off action when it does not wish to decrease its aspirational level (denoted θ) for a given service-oriented negotiation. Thus, the agent first needs to generate some/all of the potential contracts for which it receives the score of θ. Technically, it needs to generate contracts that lie on the iso-value (or indifference) curve for θ [10]. As all these potential contracts have the same value for the agent, it is indifferent amongst them. Given this fact, the aim of the trade-off mechanism is to find the contract on the iso-curve that is most preferable to the negotiation opponent (since this maximises the joint gain). More formally, an iso-curve is defined as:

Definition 1 *Given an aspirational scoring value θ, the iso-curve set at level θ for agent a is defined as:*

$$iso_a(\theta) = \{x \mid V^a(x) = \theta\} \tag{1}$$

From this set, the agent needs to select the contract that maximises the joint gain. However, since an agent does not know its opponent's utility function some form of approximation is needed. The heuristic we employ is to select the contract that is most "similar" to the opponent's last proposal (since this may be more acceptable to the opponent). To compute similarity we use the concept of *fuzzy similarity* [18]. Asuming we have a formula like *"if p then q"*, fuzzy similarity is an approximation heuristic that supports reasoning of the kind *"if aproximately p then aproximately q"*. This technique was chosen because it allows an agent to approximately model the closeness of two contracts in decision making.

A trade-off can now be defined as:

Definition 2 *Given an offer, x, from agent a to b, and a subsequent counter offer, y, from agent b to a, with $\theta = V^a(x)$, a trade-off for agent a with respect to y is defined as:*

$$trade-off_a(x, y) = \arg \max_{z \in iso_a(\theta)} \{Sim(z, y)\} \tag{2}$$

where the similarity, Sim, between two contracts is defined as a weighted combination of the similarity of the issues:

Definition 3 *The similarity between two contracts x and y over the set of issues J is defined as:*

$$Sim(x, y) = \sum_{j \in J} w_j Sim_j(x_j, y_j) \tag{3}$$

with $\sum_{j \in J} w_j = 1$ and Sim_j being the similarity function for issue j. These weights represent the level of importance the agent believes its opponent places on the various issues. For example when reasoning about what deal to offer, an oil company negotiator, when interacting with an ecologist, may safely assume that the pollution risks are given greater weight by the ecologist than the oil production costs.

Following the results from [16], a similarity function that satisfies the axioms of reflexivity, symmetry, and t-norm transitivity can always be defined as a conjunction (modelled, for instance, as the minimum) of appropriate fuzzy equivalence relations induced by a set of criteria functions h_i. A criteria function is a function that maps values from a given domain into $[0, 1]$. Correspondingly, the similarity between two values for issue j, $Sim_j(x_j, y_j)$, is defined as:

Definition 4 *Given a domain of values D_j, the similarity between two values $x_j, y_j \in D_j$ is:*

$$Sim_j(x_j, y_j) = \bigwedge_{1 \le i \le m} (h_i(x_j) \leftrightarrow h_i(y_j)) \tag{4}$$

where $\{h_1, \ldots, h_m\}$ is a set of comparison criteria with $h_i : D_j \to [0, 1]$ and $\leftrightarrow$ is an equivalence operator. In our case, the criteria functions are given in section 3.1 and $1- \mid h(x_j) - h(y_j) \mid$ is used as the equivalence operator (since this is a straightforward measure of the absolute Euclidean distance between two points). The conjunction can be any t-norm function.

To illustrate the modelling of similarity in a given domain, consider the example of colours. $D_{colours} = \{yellow, violet, magenta, green, cyan, red, \ldots\}$. In order to model how similar two given colours are, we can consider different perceptive criteria. For instance, there are 'warm' colours and 'cold' colours. With respect to this criterion, *yellow* and *orange* are more similar that *yellow* and *violet*. We could also consider the criterion of visibility (as well as many others). Green is the colour with the worst visibility and yellow and cyan are those with the best. We can use these two criteria to model our example as (we present functions extensively as sets of pairs (input, output)):

$$\begin{aligned} h_t = \{&(yellow, 0.9), (violet, 0.1), (magenta, 0.1),\\ &(green, 0.3), (cyan, 0.2), (red, 0.7), \ldots\}\\ h_v = \{&(yellow, 1), (violet, 0.5), (magenta, 0.4),\\ &(green, 0.1), (cyan, 1), (red, 0.2), \ldots\} \end{aligned}$$

where h_t and h_v are, respectively, the comparison functions corresponding to temperature (warm is 1, cold is 0) and visibility (maximum is 1, minimum 0).

With these functions, and using min as conjunction, we can obtain by simple arithmetic that:

$$\begin{aligned} &Sim_{colour}(yellow, green) =\\ &\quad min(1- \mid h_t(yellow) - h_t(green) \mid,\\ &\quad\quad 1- \mid h_v(yellow) - h_v(green) \mid) =\\ &\quad = min(0.4, 0.1) = 0.1 \end{aligned}$$

or, $Sim_{colour}(cyan, violet) = min(0.9, 0.5) = 0.5$

2.3. The Trade-Off Algorithm

The trade-off algorithm performs an iterated hill-climbing search in a landscape of possible contracts. The search proceeds by successively generating contracts that lie closer to the iso-curve (representing the agent's aspiration level), followed by the selection of the contract that maximises the similarity to the opponent's last offering. The algorithm terminates when the last selected contract lies on the iso-curve.

The algorithm starts at y, the oponent's last offer, and moves towards the iso-curve associated with the agent's last offer, x, in S steps. Each step starts by randomly generating N new contracts that have a utility E greater than the contract selected in the last step y^j (or $y^0 = y$ if it is the first step). N is referred to as the number of children. Each new contract y^{j+1} so generated satisfies $v(y^{j+1}) = v(y^j) + E$. From the generated children contracts, the one that maximises the similarity with respect to the oponent's contract y is selected. E is computed as the overall difference between the value of x and y divided by the number of steps. That is, $E = \frac{v(x)-v(y)}{S}$. Below we present the algorithm responsible for generating a new random contract. This algorithm will thus be invoked N times at each step in order to compute the best trade-off contract (giving SN calls in total). The algorithm generates children by spliting the gain in utility, E, randomly among the set of issues under negotiation.

inputs: y^j; /* last step best contract. $y^0 = y$ */
E; /* step utility increase */
$v()$; /* value scoring function */
output: y^{j+1}; /* child of y^j */
begin
(1) $\overline{E}_i := 1 - v(y_i)$;
(2) $E_{max} := \sum w_i \overline{E}_i$;
(3) $\delta = 0.01 E_{max}$
if $(E_{max} > E + \delta)$ **then**
(4) $k := 0$; $E_n := 0$;
while $(E_n < E)$ **do**
$k := k + 1$;
(5) $r_i^k := random(0, \overline{E}_i)$;
(6) $E_n := E_n + \sum_i w_i r_i^k$;
(7) $\overline{E}_i := \overline{E}_i - r_i^k$;
endwhile
(8) $E_i := \left(\sum_{j=1}^{k} r_i^j\right) \frac{E}{E_n}$;
(9) $y_i^{j+1} := v_i^{-1}\left(v_i(y_i^j) + E_i\right)$;
else $raise\ error$
end

In more detail: (1) maximum utility gain per issue, (2) total maximum utility gain, (3) setting of the average number of iterations, (4) initialization of steps and of gained utility, (5) generation of a random value for utility gain for each issue, (6) update the utility gained in iteration k, (7) fix the utility potential gain for next iteration, (8) normalization, and (9) compute the value for each issue in the new contract.

2.4. Algorithmic Complexity

When analysing the complexity of our algorithm the first thing to note is that it includes a call to a random number generator inside the main loop (step 5). This has a direct impact on the number of iterations, and hence on the time the algorithm will take. Assuming the random number generator is probabilistic in nature, we cannot make a 'big-O' analysis of the complexity [1]. However, what we can compute is an "average case" assuming that the random generator is perfect.

Let n be the number of negotiation issues. Steps 1, 5, 6, 7, 8, and 9 all need a time which is $O(n)$ $(1 \leq i \leq n)$. The time used by the algorithm will be proportional then to the number of iterations, k, of the while loop, multiplied by the cost of each iteration (which, as said, is $O(n)$). That is, it will be proportional to kn. Let us derive how large k can be. The while loop will terminate when E_n becomes bigger than E. We know that before entering the loop for the first time $E_{max} = \sum_i \omega_i \overline{E_i}$ and $E_{max} > E + \delta$. E_n is the weighted addition of the portions r_i^k generated by each iteration. On average, and assuming perfect random number generation, at every iteration we will increment E_n by half of each issue's maximum potential utility gain given to the random generator, that is, $\sum_i \frac{\overline{E_i}}{2}$. Thus, in the first iteration, the algorithm will consume a half of E_{max}, i.e. $E_n = 0 + \sum_i \omega_i \frac{\overline{E_i}}{2}$ which is $\frac{E_{max}}{2}$. In the second, a half of the remaining amount, that is a half of $\frac{E_{max}}{2}$, i.e. $\frac{E_{max}}{4}$. In general, we'll consume $\frac{E_{max}}{2^k}$ at step k and leave $\frac{E_{max}}{2^k}$ for the next step. That is, E_n at step k is $E_n = E_{max} - \frac{E_{max}}{2^k}$. We can then compute the average value for k as a function of the difference between E_{max} and E. Given that we stop when $E_n > E$, we have $E_{max} - \frac{E_{max}}{2^k} > E$, that is, $E_{max} - E > \frac{E_{max}}{2^k}$. The step before we had $\frac{E_{max}}{2^{k-1}} > E_{max} - E$. Taking this latter inequality, it is easy to see that $k < 1 + log\frac{E_{max}}{E_{max}-E}$. As we consider that $E_{max} - E > \delta$ is true, we have $k < 1 + log\frac{E_{max}}{\delta}$. A policy to decide which value to assign to δ could be to fix its value as a percentage of E_{max}. For instance, making δ a 1% of E_{max} would mean that $k < 1 + log\frac{E_{max}}{0.01E_{max}}$, that is $k < 1 + log100 < 8$; eight iterations on average. Summarising, if we fix δ as a percentage c of E_{max}, we can see that the average number of iterations is $k = 1 + log\frac{1}{c}$. Thus, on average the total time of the algorithm is proportional to $(1 + log\frac{1}{c})n$.

3. Experimental Analysis

Having developed and analysed our trade-off algorithm, the next step is to evaluate its operational performance. To this end, we wish to obtain two types of empirical information. One set of experiments seek to investigate the *parameters* of the trade-off algorithm in generation of a *single* offer, while the other set seeks to investigate the *process* of negotiation when agents use trade-off and/or responsive mechanisms. The former, referred to as single-offer experiments, aim to evaluate the kernel of the trade-off algorithm. The latter, refered to as meta strategy experiments, deal with the dynamics of the algorithm when interacting with other mechanisms.

3.1. Experimental Procedures

Both types of experiments involve an offer/offers from one negotiator, a $player$, to another, the $opponent$. Furthermore, both experiments involve negotiation over four quantitative issues $[price, quality, time, penalty]$. The reservation values of each issue for both agents are considered to be the same. The importance weight vectors of the agents (section 2.1) are fixed throughout the negotiation: $W^{player} = [0.1, 0.5, 0.25, 0.15]$ and $W^{opponent} = [0.5, 0.1, 0.05, 0.35]$[1]. The value function V_i^a used by agent a for issue i is a linear scoring function of the following type:

$$V_i^a(x_i) = \begin{cases} \frac{max_i^a - x_i}{max_i^a - min_i^a} & \text{if decreasing} \\ \frac{x_i - min_i^a}{max_i^a - min_i^a} & \text{if increasing} \end{cases}$$

where $increasing$ and $decreasing$ refer to the direction of change in score as the value of that issue increases. For example, increasing the $price$ of the service decreases the score for a client, but increases it for a seller.

Other input variables of the trade-off algorithm were set in the following way. The discriminatory power—the magnitude of the difference between the input and output—of the criteria function (equation 4) was set so that it exhibited two properties. Firstly, that it has more discrimination within the issues' reservation values (as compared to values outside this range), since most of the negotiation will take place in this region. Thus, maximal discrimination should be between an issue's min and max values (section 2.1). We parameterised this reservation value requirement by the independent variable ϵ. When ϵ is low, the function should be maximally discriminative for values within the issue's

[1] Generally speaking, the differences in these weights are one of the key elements that provide the opportunity for joint improvements (the other being the different shapes of the scoring functions). For example, an increase in $price$ may have little effect in value for the $player$, but relatively more for the $opponent$.

reservation limits (*mutatis mutandis* when ϵ is high). Secondly, we also want to experiment with different discriminatory power *within* the reservation range (to support different similarity measures for different issues). For example, for one issue it may be desirable to have maximal discrimination at the centre of the reservation values, whereas for another issue maximal discrimination may be desired at the extremes of the reservation values. We parameterise this requirement using the variable α. When α is high, more discrimination is placed towards the maximum of the reservation values (*mutatis mutandis* when it is low). The following function satisfies these two requirements:

$$h(x) = \frac{1}{\pi} atan \left[\left(\frac{2 \mid x - min \mid}{x - min} \left| \frac{x - min}{max - min} \right|^{\alpha} - 1 \right) tan(\pi(\frac{1}{2} - \epsilon)) \right] + \frac{\pi}{2} \quad (5)$$

In this case, in order to be quite discriminatory, ϵ was fixed at 0.1 for all issues. For all issues, we fixed the different αs to be equal, $\alpha^{price} = \alpha^{quality} = \alpha^{time} = \alpha^{penalty} = 1$, to have linear criteria functions $(h_i's)$ that have equal discrimination power across the issue's reservation values. We chose to make ϵ and α constant to reduce the number of free variables in the experiments. However, normally the setting of values for ϵ and α reflects the agent's domain knowledge.

3.1.1 Single-Offer Experiment Variables

In these experiments the independent variables were: i) the number of children generated at each step in hillclimbing to the iso-curve, ii) the number of steps taken to reach the iso-curve and iii) the information that is available to an agent regarding the importance (or weight) the opponent places on each issue in computing the contract's value (equation 3). Values for the first and second variables control the amount of search performed by the trade-off algorithm. Experiments were run where the number of children was selected from the set $\{5, 100, 200\}$. The number of steps to the iso-curve was selected from the set $\{1, 40\}$. In the third set of dependent variables, an agent can have perfect, partial, imperfect or uncertain information on how the other agent weights the issues that are input into its similarity function (equation 3). In experiments with perfect information, the algorithm, in computing similarity, is given the other agent's weights for different issues (cardinally correct information). Partial information games are where the algorithm is given the correct order of importance but not the actual issue weights (ordinally correct information). Imperfect games represent the situation where the algorithm is given no information about the other's weights. Finally, uncertain information games represent cases where the algorithm is given undifferentiated weights for each issue, in this case $[0.25, 0.25, 0.25, 0.25]$. The dependent variable in all our experiments is the generated contract for both agents.

The experimental procedure consisted of inputting two contracts, representing x and y, into the algorithm under each of the dependent variable environments and observing the execution trace of the algorithm for *an* offer from the $player$ to the $opponent$. All input contracts (x and y) were subject to the general constraint that $v^{player}(y) < v^{player}(x)$ and $v^{opponent}(x) < v^{opponent}(y)$. This ensured trade-offs are possible by ruling out all those contracts that are already of a higher value to either party. The control set was generated by chosing the preferred child randomly at each step approaching the iso-curve (as opposed to using the similarity criteria).

3.1.2 Meta Strategy Experiment Variables

The aim of these experiments was to empirically evaluate the outcome and dynamics of negotiation when agents used either a trade-off mechanism or a responsive mechanism or a combination of the two in the course of negotiation (that is, a meta strategy of which mechanism to select in order to generate a series of counter-proposals). The first offer of both agents was generated using responsive mechanisms, since the trade-off mechanism requires at least one offer from the opponent. After that, an agent is faced a choice of which mechanism to select. Since there can be an infinite number of meta strategies (as many as potential sequences of chosings between responsive and trade-off types of counterproposals), the meta strategies considered in these experiments were limited to the set $\{responsive, smart, serial, random\}$. Responsive simply selected the responsive mechanism for generating an offer throughout negotiation. This was included to compare the trade-off mechanism against an agent that always concedes utility. A smart strategy consisted of deploying a trade-off mechanism until the agent observed a deadlock in the average closeness of offers between both agents as measured by the similarity function. That is, the distance between the offers was not reducing. Under these circumstances, the value of the previously offered contract, $V^a(x)$, was reduced by a predetermined amount, here 0.05, thereby lowering the input value of θ into the trade-off mechanism. A serial strategy involves alternating between the trade-off and responsive mechanisms. Finally, the random meta strategy randomly selected between the two mechanisms. The parameters of the responsive mechanism (see [4]) were set to produce concession behaviours, since being responsive often involves concessions in the light of environmental needs (e.g. time, resources etc.). For the trade-off algorithm, the number of children and number of steps were set to 100 and 40 respectively and the similarity weights were set at uncertain settings of $[0.25, 0.25, 0.25, 0.25]$. Both negotiators were given a deadline of twenty offers.

3.2. Results

Figure 1 and the top row of figure 2 show the results of varying, under different information inputs, the number of children generated in single-offer experiments when the number of steps to the iso-curve was set to 40. The bottom row of figure 2 represents the case where the number of children was set to 100, but the trade-off algorithm computed the iso-contract in a single step. The dot-dash line represents the execution trace of the random control, the solid line emanating from y the similarity based trade-off execution trace, and the line joining $(0, 1)$ to $(1, 0)$ the pareto-optimal line. The results show four major patterns. Firstly, when moving to the iso-curve if the space of possible contracts is not explored sufficiently, 5 children (figure 1 top row) or 1 step (figure 2 bottom row), then the gains of the $opponent$ are small. More specifically, only when the $player$ has perfect information about the $opponent$'s evaluations and the trade-off mechanism operates in 1 step with 100 children will the mechanism improve the offer (from the $opponent$'s perspective) (figure 2 E). The next best contract for the $opponent$ is when the $player$ has the same value as x (figure1 A). All other contracts generated by the $player$ when not fully exploring the search space (figures 1 B,C,D and 2 F) have lower value to the $opponent$ than x.

Secondly, in nearly all cases, the similarity based trade-off out performs the policy of randomly selecting a child for the next step towards the iso-curve. However this pattern does not hold for the cases of reaching the iso-curve in one step under partial and uncertain information environments (figure 1 G and H). This is the result of chance, rather than randomness being a better strategy in this type of environment.

Thirdly, the $opponent$'s benefit increases as the algorithm performs more search (5 to 200 children). Furthermore, there is no significant difference between perfect and partial information outcomes within the 100 and 200 result categories. This indicates that our algorithm requires only partial ordering information, rather than perfectly cardinal orderings, in order to compute outcomes that are better for the $opponent$.

Finally, for all environments and variable combinations, imperfect information results in significantly poorer outcomes for the $opponent$ than the other information classes. This is only to be expected since search is directed towards erroneous directions.

Figure 3 presents the data for the meta-strategy experiments. Individual offers between the $player$ and the $opponent$ are depicted as circles and squares respectively. The sequences of offers are joined by a solid line for the $player$ and a dotted line for the $opponent$. The final agreement is depicted as the offer where the circle and square meet.

The observed data exhibits two patterns. Firstly, there is a clear rank ordering across meta-strategy pairings over the summed joint value gained for the final outcome. The highest joint gain is achieved in negotiations between two $smart$ meta-strategists. In this case the final outcome is close to the pareto-optimal line, implying that such a pairing of meta-strategies results in outcomes that are most beneficial to both parties. The remaining rankings for $player, opponent$ pairings of meta-strategies are then [smart,serial], [serial,serial], [smart,random], [smart,responsive], [serial,responsive], [random,responsive], [random,random] with respective joint gains of 1.27, 1.18, 1.146, 1.11, 1.076, 1.06, 0.99. In general, the higher joint utilities occur when at least one of the agents is $smart$. The $random$ meta strategists, as expected, perform worst.

The other observable pattern relates to the number of messages exchanged between agents using different meta-strategies. This indirectly measures the communication load a meta-strategy places on the agents. The observed pattern is almost the reverse for the joint value outcomes above, with a [smart,smart] pairing incurring the highest communication cost (reaching a deal after 20 rounds), followed by [random,random], [smart,responsive], [smart,random], [smart,serial] (14 rounds), [serial,serial] (13 rounds), and [serial,responsive], [smart,smart] (12 rounds). This observation supports our intuition that higher joint utilities are gained through greater search, which, in turn, involves more communication between the agents.

In summary, these results indicate that unless agents know, at least partially, the importance the other agent attaches to an issue, then the best policy for computing trade-offs is to assign uncertain weightings to all issues. These weightings can then be updated by some learning rule towards partial or perfect information models, since a) information models are private and b) erroneous predictions can result in poorer outcomes. Furthermore, engaging in trade-off negotiation, particularly with a high search factor by both parties, results in higher joint gains. However, this improvement is achieved at the expense of an increased communication load.

4. Related Work

A number of approaches have been advocated for the process of making decisions during the course of negotiation. Chief amongst these is work on game theory. This strand of work has produced a large number of sophisticated and specialised models [11], which, although analytically well formed, are generally inappropriate for our purposes

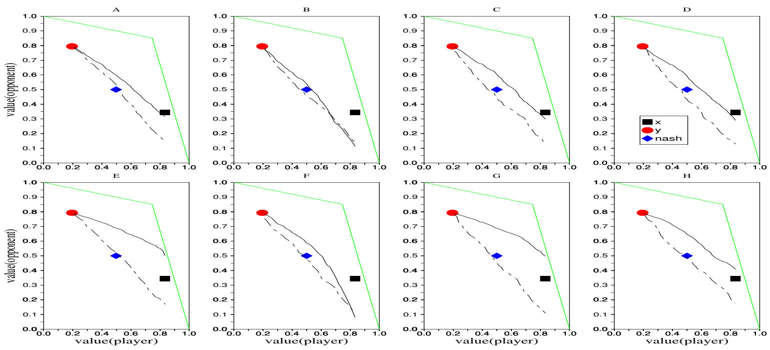

Figure 1. Data for 5 children in 40 steps (first row) and 100 children in 40 steps (second row). A) & E) Perfect information, B) & F) Imperfect information, C) & G) Partial Information, D) & H) Uncertain information.

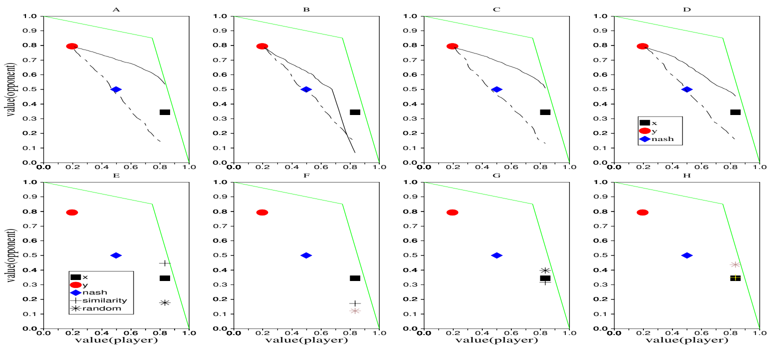

Figure 2. Data for 200 children in 40 steps (first row), and 100 children in 1 step (second row). A) & E) Perfect information B) & F) Imperfect information, C) & G) Partial Information, D) & H) Uncertain information.

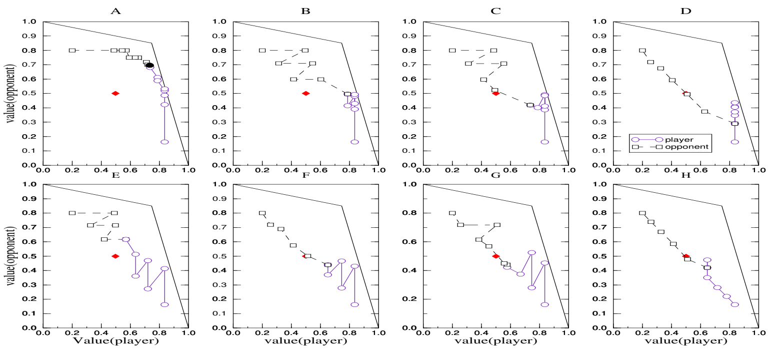

Figure 3. Dynamics of Negotiation Process for Meta Strategies: A) smart v. smart, B) smart v.serial, C) smart v. random D) smart v. responsive, E) serial v. serial, F) serial v. responsive, G) random v. random, H) random v. responsive.

(because they specify the solution properties and leave the process of how to reach these points unspecified and because they generally violate our privacy of information requirements). Strategic game theoretic models [12, 7], on the other hand, do model the process of negotiation. However they often make unrealistic information assumptions (eg that agents know each other's type [5]) and they do not model negotiation for joint gains. Overall, in contrast to game theoretic models, our work is targeted more to open systems, where information is sparse and computational resources are limited. In such environments, a satisficing solution is the best that can be hoped for.

Uncertainty in negotiation was also addressed by using decision theoretic models in the *Persuader* system [15] where multi-attribute utility theory was combined with case-based reasoning in contexts where the agent had no previous cases to reason with. This dual approach is similar to our work in that agents use both utility and similarity for decision making. However, we use similarity rather than utility to address the inherent uncertainties involved and, as we have shown in section 3.2, this appears to be a better choice in uncertain environments.

The process of negotiation has also been modeled as a distributed constraint satisfaction problem [13]. In such cases, an agent's objectives are represented as constraints together with their associated utilities. Strategies (e.g. composition, reconfiguration and relaxation operators) are then used to modify these constraints, or the current solution, until a final solution is reached. The relaxation of constraints is similar to our previous work on concession mechanism for negotiation, and the modification of the current solution closely resembles the trade-off mechanism reported here. However, in our work there is only one objective, namely reaching a contract which maximises value. Therefore, our approach is to develop reasoning mechanisms that deliberate over raw values rather than objectives.

5. Conclusions and Future Work

This paper presented a formal model and a related algorithm for carrying out trade-offs in automated negotiations. The algorithm is designed to work in a distributed setting in which agents have limited information about the preferences of their negotiation opponent and limited computational resources to devote to the negotiation process. Analytical and empirical evaluation showed our algorithm to be effective in such cases. Moreover, even when comparatively little information is known about the opponent's preferences, our algorithm still finds reasonable trade-offs and does so in an acceptable number of negotiation cycles.

For the future, we aim to use similarity measures to manipulate the set of negotiation issues at run-time, as well as using fuzzy techniques to model an agent's preferences and its ratings of the importance of the negotiation issues.

References

[1] A. Aho, J. Hopcroft, and J. Ullman. *Data Structures and Algorithms*. Addison-Wesley, Reading, Massachusetts, USA, 1985.

[2] K. P. Corfman and S. Gupta. Mathematical models of group choice and negotiations. *Handbooks in Operation Research and Management Sciences*, 5:83–142, 1993.

[3] P. Faratin, N. R. Jennings, P. Buckle, and C. Sierra. Automated negotiation for provisioning virtual private networks using fipa-compliant agents. In *Fifth International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology*, 2000.

[4] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3–4):159–182, 1998.

[5] R. Gibbons. *A Primer in Game Theory*. Harvester Wheatsheaf, New York, 1992.

[6] N. R. Jennings, P. Faratin, T. J. Norman, P. O'Brien, and B. Odgers. Autonomous agents for business process management. *Int. Journal of Applied Artificial Intelligence.*, 14(2):145–189, 2000.

[7] S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence Journal*, 75(2):297–345, 1995.

[8] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[9] D. G. Pruitt. *Negotiation Behavior*. Academic Press, 1981.

[10] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, Cambridge, USA, 1982.

[11] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. The MIT Press, Cambridge, USA, 1994.

[12] A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–109, 1982.

[13] A. Sathi and M. Fox. Constraint-directed negotiation of resource reallocation. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence Volume II*, pages 163–195, San Mateo, California, 1989. Morgan Kaufmann.

[14] C. Sierra, P. Faratin, and N. R. Jennings. A service-oriented negotiation model between autonomous agents. In M. Boman and W. V. de Velde, editors, *Proceedings of 8th European Workshop on Modelling Autonomous Agents in Multi-Agent World*, number 1237 in LNAI, pages 17–35. Springer-Verlag, 1997.

[15] K. Sycara. Multi-agent compromise via negotiation. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence Volume II*, pages 119–139, San Mateo, California, 1989. Morgan Kaufmann.

[16] L. Valverde. On the structure of F-indistinguishability. *Fuzzy Sets and Systems*, 17:313–328, 1985.

[17] N. Vulkan and N. R. Jennings. Efficient mechanisms for the supply of services in multi-agent environments. *Int Journal of Decision Support Systems*, 28(1–2):5–19, 2000.

[18] L. A. Zadeh. Similarity relations and fuzzy orederings. *Information Sciences*, 3:177–200, 1971.