Opponent Modelling in Automated Multi-Issue Negotiation Using Bayesian Learning

Koen Hindriks
Man-Machine Interaction Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
+31.15.2781315
k.v.hindriks@tudelft.nl

Dmytro Tykhonov
Man-Machine Interaction Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
+31.15.2783737
d.tykhonov@tudelft.nl

ABSTRACT
The efficiency of automated multi-issue negotiation depends on the availability and quality of knowledge about an opponent. We present a generic framework based on Bayesian learning to learn an opponent model, i.e. the issue preferences as well as the issue priorities of an opponent. The algorithm proposed is able to effectively learn opponent preferences from bid exchanges by making some assumptions about the preference structure and rationality of the bidding process. The assumptions used are general and consist among others of assumptions about the independency of issue preferences and the topology of functions that are used to model such preferences. Additionally, a rationality assumption is introduced that assumes that agents use a concession-based strategy. It thus extends and generalizes previous work on learning in negotiation by introducing a technique to learn an opponent model for multi-issue negotiations. We present experimental results demonstrating the effectiveness of our approach and discuss an approximation algorithm to ensure scalability of the learning algorithm.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *intelligent agents, multi-agent systems.*

General Terms
Algorithms, Performance, Economics, Experimentation, Theory.

Keywords
Automated Multi-Issue Negotiation, Opponent Modelling, Preference Profiles, Bayesian Learning.

1. INTRODUCTION
In bilateral negotiation, two parties aim at reaching a joint agreement. They do so by exchanging various offers or bids using e.g. an alternating offers protocol [11] called the “negotiation dance” in [11]. In reaching such an agreement both parties usually aim to satisfy their own interests as best as possible, but have to take their opponent’s preferences into account as well to reach an agreement at all. This is complicated, by the fact that negotiating parties are generally not willing to reveal their preferences in order to avoid exploitation. As a result, both parties have incomplete information which makes it hard to decide on a good negotiation move and hard to reach an optimal agreement.

Research has demonstrated that human negotiators may feel they did well in a negotiation but also shows that the results of untrained negotiators are in general suboptimal [1]. One reason for this is the limited computing abilities of humans when confronted with multiple issues that are negotiated. Software agents can outperform humans in well-defined negotiation domains [7]. However, in general such agents cannot reach optimal outcomes either without sufficient knowledge about the negotiation domain or their opponents. As negotiation is recognized as an important means for agents to achieve their own goals efficiently [12] the challenge thus is to maximize the performance of automated negotiation agents given this limited availability of information.

Various options for improving the performance of negotiating agents have been outlined in the literature. The performance of a negotiating agent is to a large extent determined by the strategy used for proposing offers. Typically, in the automated negotiation literature *concession-based* strategies have been proposed. A concession-based strategy proposes as a next offer a bid that has a decreased utility compared to the previously proposed offer. An example of such a strategy, which does not use any domain or opponent knowledge, is the ABMP strategy [6]. The ABMP strategy decides on a negotiation move based on considerations derived from the agent’s own utility space only. Such a strategy cannot search through the negotiation outcome space for outcomes that are mutually beneficial for both parties and thus is not always able to reach so-called *win-win* outcomes [11]. The ABMP strategy will therefore most likely be inefficient in complex negotiation domains although it has been shown to outperform humans in small domains [1].

A natural suggestion then is to try and incorporate additional knowledge into a negotiating agent to improve its performance. The effectiveness of providing knowledge about the domain of negotiation has been demonstrated in the Trade-off strategy introduced in [5]. In particular, this paper shows that domain knowledge (coded as so-called *similarity functions*) can be used to select bids that are close to an opponent’s bids, thus increasing the likelihood of acceptance of a proposed bid by that opponent. In this approach, the knowledge represented by similarity functions is assumed to be public. As is to be expected, if similarity functions can be found, the Trade-off strategy outperforms a concession-based strategy such as ABMP [3]. Incorporating public domain knowledge into a strategy, however, still does not take into account the private preferences or priorities that an

Cite as: Opponent Modelling in Automated Multi-Issue Negotiation Using Bayesian Learning, Koen Hindriks, Dmytro Tykhonov, *Proc. of 7th Int. Conf. on Autonomous Agentsand Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 331-338.

opponent associates with negotiated issues. The more knowledge of these preferences is available the better the chance of win-win scenarios and optimal outcomes.

The private preferences of an agent will not simply be revealed to an opponent. For example, generally it is unwise to reveal information about what is minimally acceptable (your *reservation price*) since this will provide an opponent with the opportunity to force this outcome [11]. If the negotiating parties have a sufficient amount of trust in each other, some information might be volunteered. Humans might also offer feedback about the bids received from the opponent (e.g., your last bid is actually worse than your previous bid). If no information is offered freely, an alternative to obtain information about an opponent's private preferences is to derive it from the negotiation moves performed by that opponent during a negotiation. Various learning techniques have been proposed to uncover such private preferences [2, 5, 6, 13]. A complicating factor in this context is that the number of moves performed before reaching an agreement is limited (typically about 5 to 30 moves), and individual bids do not provide much information [14].

In this paper, we show that it is nonetheless possible to construct an *opponent model*, i.e. a model of the opponent's preferences, that can be effectively used to improve negotiation outcomes. We provide a generic framework for learning both the preferences associated with issue values as well as the weights that rank the importance of issues to an agent. The main idea is to exploit certain structural features and rationality principles to limit the possible set of preference profiles that can be learned. We present a learning algorithm based on Bayesian learning techniques that uses assumptions about the structure of opponent preferences and the rationality of the bidding process itself. Our approach can be integrated into various negotiation strategies since the main focus is on learning an opponent's utility space. The framework allows for the incorporation of prior available opponent knowledge but does not require any such knowledge. It thus extends and generalizes previous work on learning in negotiation by introducing a technique to learn opponent preferences for multi-issue negotiation.

The remainder of the paper is organized as follows. Section 2 discusses related work in the area of opponent modelling. In Section 3 the approach for learning an opponent model is introduced and the structural and rationality assumptions that enable such learning are explained. Section 4 presents experimental results to demonstrate the effectiveness of the approach. In Section 5 the learning algorithm is presented and additional techniques are introduced to manage the computational complexity of the learning algorithm. Finally, Section 6 concludes the paper and suggests several directions for future research.

2. RELATED WORK

Previous research analyzing various negotiation domains and algorithms, see e.g. [3, 5, 14], has shown that efficient negotiation requires both knowledge about the negotiation domain as well as about opponent preferences. In particular, some idea of what the opponent preferences are like is required to avoid so-called *unfortunate* steps in which a bid is proposed that is worse for both parties than one of the previous bids [3]. Related work in the area of opponent modelling in negotiation has resulted in a variety of approaches that usually focus on learning one aspect of the negotiation process, such as learning the opponent's reservation point [13], issue priorities (typically weights are used to model the relative importance of each issue; [2, 6]), or the negotiation strategy itself [8]. This is only natural given the limited amount of evidence that can be used to learn from in a single negotiation.

In order to position our own work, we discuss its relation to that of others. In [8] an approach to learn an opponent's negotiation strategy as a sequence of bids made by that party is presented. The approach uses Markov chains to model the opponent strategy and Bayesian learning to update the probabilities of the transitions between states in the Markov chain. It does assume, however, that negotiations involve only *one issue.* Automated learning of a negotiation strategy is hard and is only feasible using data from *multiple, successive negotiations*. In this paper we do not attempt to learn a negotiation strategy but instead assume an opponent uses some form of *concession-based strategy*. The framework we present is able to learn *multi-issue preferences during a single negotiation*.

In [5] a model is presented that incorporates domain knowledge for deciding on a negotiation move. This approach is extended in [2], which proposes to use kernel density estimation (KDE) to learn the issue priorities (weights) of an opponent. The basic framework modelling issue preferences by means of domain knowledge remains intact but is complemented to *learn private issue priorities*. We use the same structure of preference profiles, which allows for *arbitrary sets of issue values* and assumes that *issues are independent* [11]. In our approach, however, *both issue priorities as well as preferences over issue values can be learned*. Our framework also allows for the incorporation of available domain knowledge before a negotiation is started.

Our approach is most related to work based on Bayesian learning. An interesting approach to opponent modelling is that of learning some of the parameters of an opponent strategy [13]. The opponent modelling proposed by [13] uses the Bayesian update rule to learn an opponent's *reservation point in one-issue negotiation*. In [7] an opponent profile is learned in a qualitative negotiation setting. It is assumed that a *fixed set of possible opponent profiles* is given. Bayesian learning then is used to determine the likelihood that an opponent has one of these given profiles. The profile types are assumed to be public knowledge and an agent only has to learn which type of profile its opponent most likely has. Our approach to learning opponent preferences is also based on Bayesian learning but we introduce a general learning algorithm that is able to learn both issue preferences as well as issue priorities in a *multi-issue negotiation* and enables the learning of opponent *preference profiles* that have *not been previously fixed*.

3. LEARNING AN OPPONENT MODEL

Our goal is to introduce a learning approach that can be used to model an opponent in a negotiation with imperfect information. In this sense, negotiation can be viewed as an instance of a Bayesian game. In game theory, the class of Bayesian games refers to games in which players do not have complete information about each others' preferences (or *types*) [11]. In such a setting, players can use evidence (or so-called *signal functions*) to update their beliefs about the other party. In a Bayesian game, in order to be able to learn, it is necessary to specify the strategy spaces and type spaces. Ideally, these spaces are defined generically enough to allow learning of a rich variety of opponent profiles. At the same time, however, these spaces should not be so rich to make it impossible to learn an opponent profile from the limited available evidence (in our case, the opponent's bids). In this section, we present the hypothesis space that defines the range of opponent profiles that can be learned. We do so by introducing various reasonable assumptions about the structure of opponent profiles as well as about an opponent's negotiation strategy. These assumptions are introduced to ensure the task of learning an opponent model is feasible. In Section 4 we present evidence that the proposed model is both effective as well as rich enough to learn opponent preferences in various negotiation domains.

3.1 Structural Assumptions

Our first assumption is a common one, see e.g., [11], and assumes that the utility of a bid can be computed as a weighted sum of the utilities associated with the values for each issue. Utility functions modelling the preferences of an agent thus are linearly additive functions and are defined by a set of weights w_i (or priorities) and corresponding evaluation functions $e_i(x_i)$ for each of n issues by:

$$u(b_t) = \sum_{i=1}^{n} w_i e_i (x_i \in b_t) \quad (1)$$

where x_i is the value of issue i in bid b_t in the negotiation round t. To ensure that a utility function has a range in [0, 1], the range of the evaluation functions is assumed to be in [0,1] and the weights are assumed to be normalized such that their sum equals 1.

In order to learn an opponent's preference profile or utility function $U(b)$ we need to learn both the issue priorities or weights w_i as well as the evaluation functions $e_i(x_i)$. The objective of learning an opponent model thus is to find a model as defined by (1) that is the most plausible candidate or best approximation of the opponent's preference profile.

Our next assumption concerns the issue priorities in a preference profile (1). Some knowledge about issue priorities is important in order to be able to propose a trade-off on issues that are valued differently by negotiating parties. In [3] it is shown that in general it is not sufficient to know issue preferences, i.e. evaluation functions $e_i(x_i)$, to be able to make trade-offs. Trade-offs are an important means to get closer to the Pareto efficient frontier. To be able to propose a trade-off an agent must know at least two issues one of which is valued more by itself than its opponent and one which is more valued by the opponent than itself. In that case, an agent can make a concession on a less-valued issue that is valued more by its opponent and propose an issue value that is more highly valued by the agent itself.

In [5] it is argued that it is typically sufficient to know the ranking of the weights to be able to make trade-offs and significantly increase the efficiency of an outcome. We propose to define the set of hypotheses H^w about the private weights of an opponent as the set of all possible rankings of weights. It is then straightforward to associate real-valued numbers again with a $h_j \in H^w$ about weights, which can be computed as a linear function of the rank and also ensures weights are normalized, as follows:

$$w_i = 2\frac{r_i^j}{n(n+1)}$$

where r_i^j is the rank of weight w_i in the hypothesis h_j and n is the number of issues.

Finally, we need to impose some additional structure on the evaluation functions in order to be able to learn a preference profile. To facilitate the learning of an opponent's preferences over issue values we introduce a hypothesis space of predefined function types. A third assumption thus concerns the shape of evaluation functions and we assume that preferences over issue values can be modelled by means of three types of functions:

- *downhill* shape: minimal issue values are preferred over other issue values (think, e.g., of price and delivery time for a buying agent), and the evaluation of issue values decreases linearly when the value of the issue increases;
- *uphill* shape: maximal issue values are preferred over other issue values (think, e.g., of price and delivery time for a selling agent), and the evaluation of issue values increases linearly when the value of the issue increases;
- *triangular* shape: a specific issue value somewhere in the issue range is valued most and evaluations associated with issues to the left ("smaller") and right ("bigger") of this issue value linearly decrease (think, e.g., of an amount of goods).

Figure 1 below illustrates this set of functions and introduces labels $h^e_{i,j}$ to refer to the hypothesis that issue i has associated *e*valuation function j.

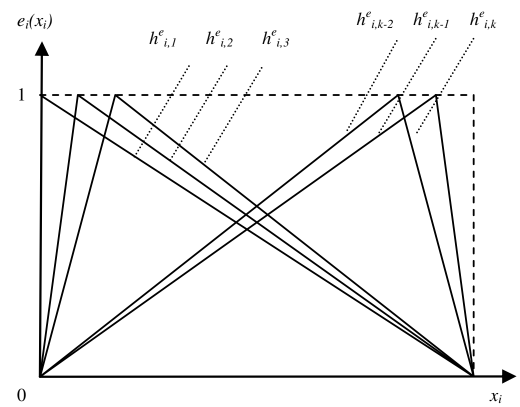

Figure 1. Hypothesis space of possible evaluation functions.

The three function types that define the range of possible evaluation functions are common in the literature, and, most importantly, in combination allow for the modelling of other types of function as well (see Figure 2 below).

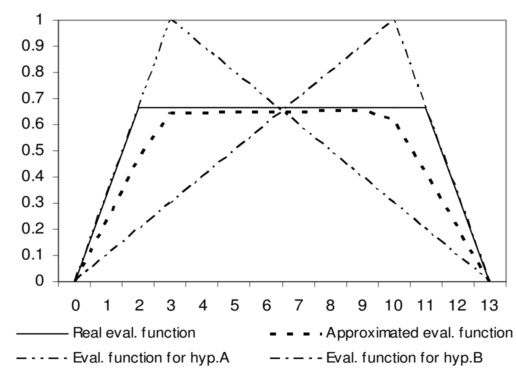

Figure 2. Approximation of an evaluation function that is not in the hypothesis space by means of two evaluation functions.

In order to see this, it should be taken into account that a probability distribution is associated with each hypothesis. This allows other types of functions to be approximated by associating different probabilities with various hypotheses. The predicted evaluation of an issue value is derived from all hypotheses that are assigned a non-zero probability. The evaluation thus can be viewed as computing a most probable evaluation value of an issue value by computing the weighted sum of all evaluations of an issue value associated with some hypothesis with non-zero probability. Different probability distributions thus allow for approximating different types of evaluation functions that do not need to match any single evaluation function from the hypothesis space. Figure 2 shows an example of the approximation of a more complex evaluation function (solid line) that is not present in the hypothesis space. Many complex evaluation functions thus can be successfully approximated by a composition of several simple evaluation functions from the hypothesis space. The preferences of an agent can be viewed as a membership function that assigns a degree of membership to each hypothesis in the hypothesis space similar to membership in fuzzy set theory. In our case the membership of an evaluation function is modelled as a probability distribution and our approach is similar to that of triangular membership functions [9].

To summarize, the set of hypotheses concerning an opponent's preference profile is a Cartesian product of the hypotheses about issue weights H^w and shapes of issue evaluation functions H^e_i: $H = H^w \times H^e_1 \times H^e_2 \times \ldots \times H^e_n$.

3.2 Rationality Assumptions

The idea is to learn an opponent preference profile from its negotiation moves, i.e. the bids it proposes during a negotiation. In a Bayesian learning approach, this means we need to be able to update the probability associated with all hypotheses given new evidence, i.e. one of the bids. More precisely, we want to compute $P(h_j|b_t)$ where b_t is the bid proposed at time t. In order to be able to use Bayes' rule to do this, however, we need some information about the utility the opponent associates with bid b_t.

As this information is not generally available, we need to introduce an additional assumption to be able to make an educated guess of the utility value of b_t for an opponent. The assumption that we need is that our opponent follows a more or less rational strategy in proposing bids. In particular, we will assume that an opponent follows some kind of concession-based strategy. Although assuming such behaviour may not always be realistic it typically is necessary to perform at least some concession steps in order to reach an agreement. Moreover, in game-theoretic approaches and in negotiation it is commonly assumed that agents use a concession-based strategy [4, 10].

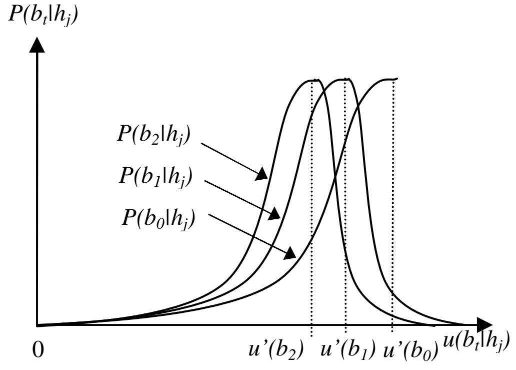

Figure 3. Conditional probability distribution of tactics.

In line with [4] we assume that a rational agent uses a time-dependent tactics (TDT). In line with such a strategy it starts with a bid of maximal utility and moves towards its reservation value when approaching the negotiation deadline. Thus, it is assumed that an agent's tactics during a negotiation can be defined by a monotonically decreasing function. This assumption still allows that an opponent uses various kinds of tactics and no exact knowledge about an opponent's negotiation tactics is assumed. More specifically, the rationality assumption is modelled as a probability distribution associated with a range of tactics (see Figure 3); as a result, each utility associated with an opponent's bid thus also has an associated probability.

In this paper we use linear functions to estimate the predicted utility value: $u'(b_t) = 1-0.05 \cdot t$. This assumption allows us to compute the conditional probability $P(b_t|h_j)$ representing the probability of bid b_t given hypothesis h_j at time t. This is done by defining the probability distribution $P(b_t|h_j)$ over the predicted utility of b_t using the rationality assumption and the utility of b_t according to hypothesis h_j (see Figure 3). Here the predicted utility $u'(b_t)$ of a next bid of the opponent is estimated as $u'(b_{t-1})$-$c(t)$ using a function $c(t)$ that is the most plausible model of the negotiation concession tactic used by the opponent. We use the following function to model the conditional distribution, where $u(b_t|h_j)$ is the utility of bid b_t according to the hypothesis h_j:

$$P(b_t|h_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(u(b_t|h_j)-u'(b_t))^2}{2\sigma^2}} \quad (2)$$

This probability distribution can be used consecutively to update the probabilities of the hypotheses using Bayes' rule to compute $P(h_j|b_t)$.

The spread σ of the conditional distribution used in (2) defines the *certainty* of the agent about its opponent's negotiation tactics. If an agent is certain about the utility of an opponent's bid b_t then σ can be set to a low value. A higher level of certainty increases the learning speed, since hypotheses predicting an incorrect utility value of a bid in that case would get assigned an increasingly lower probability, and vice versa. Overestimating the level of certainty, however, may lead to incorrect results, and some care should to be taken to assign the right value to σ.

3.3 Bayesian Learning Approach

The framework for learning introduced above can now be applied. In order to do so, the first step to perform is to initialize the probability distribution associated with each of the hypotheses in the hypothesis space H introduced in Section 3.1. This means either assigning a probability distribution to hypotheses based on available knowledge about opponent preferences, or, if no such *a priori* knowledge is available, to assign a uniform distribution.

During a negotiation at every time t when a new bid b_t is received from the opponent the probability of each hypothesis should be updated using Bayes' rule:

$$P(h_j|b_t) = \frac{P(h_j)P(b_t|h_j)}{\sum_{k=1}^{m} P(h_k)P(b_t|h_k)}$$

Here the conditional probability $P(b_t|h_j)$ represents the probability that bid b_t might have been proposed given hypothesis h_j (using the predicted utility according to rationality assumption (2)) and $P(h_j)$ is the current probability of hypothesis h_j. The normalization factor in the denominator of Bayes' rule ensures that the probability of the entire hypothesis space is 1.

The learning approach outlined will increase the probability of a hypothesis about an opponent's preference profile that is most consistent with the bid sequence received so far from that opponent and provides the best match with the utilities of these bids, estimated using the conditional probability distribution associated with tactics. As a result, the more consistent the predicted utility is with a hypothesis, the higher the probability associated with this hypothesis will be. It is possible that several hypotheses predict (almost) the same utilities for a given bid sequence, but this simply means that it is not possible to distinguish different preference profiles based upon that bid sequence and more evidence would be needed to do so.

The spread of the probability distribution $P(h_j)$ associated with the hypothesis space might also be used as a measure of the effectiveness of learning the opponent model. Presumably, successful learning of an opponent model will increase the probability of some of the hypotheses that best fit the bidding sequence received from an opponent and the number of hypotheses still considered viable would decrease. If not, the probability distributions $P(h_j)$ would remain a more or less uniform distribution. In the latter case the agent does not learn from the bids exchanged and it could use this fact in the negotiation strategy. For instance, negotiating against an eratic opponent that seems to more or less randomly propose bids, the agent might start using a Boulware strategy [4], in order to wait until an acceptable offer of the opponent is received.

Finally, during a negotiation an agent can use the updated probability distribution to compute estimates of the utility of counteroffers it considers and choose one that e.g. maximizes the utility of its opponent, to increase the likelihood of acceptance by that opponent. The expected utility $\bar{u}(b_t)$ of a counteroffer b_t may be computed as follows, where w_i and e_i are the weights respectively evaluation functions predicted by hypothesis $h_j \in H$:

$$\bar{u}(b_t) = \sum_{j=1}^{|H|} P(h_j) \sum_{i=1}^{n} w_i e_i (x_i \in b_t)$$

4. EXPERIMENTAL ANALYSIS

In this section, experiments are performed to show the effectiveness of our approach to learn the opponent model and to use it to find a good counteroffer. The Bayesian learning agents used in the experiment update their opponent model each time a new bid is received from the opponent in line with the Bayesian learning approach introduced above.

The strategy used by the Bayesian learning agents is based on the smart meta-strategy of [5]. The agent starts with proposing a bid that has maximal utility given its own preferences. Each consecutive turn the agent can either accept the opponent's bid or send a counter-offer. The agent accepts a bid from its opponent when the utility of that bid is higher than the utility of its own last bid or the utility of the bid it would otherwise propose next. Otherwise, the agent will propose a counter-offer.

The basic idea of the smart meta-strategy is to propose a counter-offer that has the same utility (lies on the same utility iso-curve) as the previous bid of the agent but improves the utility of the opponent whenever possible. Formally, the strategy searches for a bid b_{t+1} that satisfies, where u_{own} denotes the agent's own utility function and τ denotes a target utility:

$$b_{t+1} = \underset{b \in \{x \mid |u_{own}(x) - \tau| \le \delta\}}{\arg\max} \bar{u}(b)$$

The set $\{x \mid |u_{own}(x) - \tau| \le \delta\}$ represents the utility iso-curve that have the same utility for the agent, (within a small interval $[\tau - \delta; \tau + \delta]$) but might have different utilities for its opponent. The strategy selects a bid from the iso-curve that maximizes the expected utility of the opponent. The bid b_{t+1} lies on the predicted Pareto frontier according to the current opponent model. If it is not possible to find a bid that thus improves the utility of the opponent, a concession step will be performed after performing smart steps (i.e. steps that stay on the same iso-curve and try to improve the next bid for the opponent by using the updated opponent model). The agents perform a concession step by decreasing the target utility τ of their next bid by a fixed concession step c.

Two sets of experiments were run: one based on a negotiation domain with 5 issues taken from [8], and one based on a negotiation domain with 4 issues taken from [5]. To compare the performance of the Bayesian learning approach, the agents using opponent modelling were compared with agents using the Trade-off strategy and the ABMP strategy discussed in Section 2. Two variants of learning agents were tested: one with and one without initial domain knowledge; the first to compare with the Trade-off strategy which uses domain knowledge and the second to compare with the ABMP strategy which does not. All agents played against the same opponent, which used the Trade-off strategy, to be able to compare negotiation traces and results.

4.1 Experimental Results

In the first domain, the setting is that of an employee and an employer who negotiate about a job assignment and related issues such as salary. An interesting aspect of this domain is that both parties have the same preferences with regards to one of the issues. Figure 4 shows the results of the experiments, including the resulting negotiation traces as well as the Pareto efficient frontier. The agreements reached are also marked explicitly.

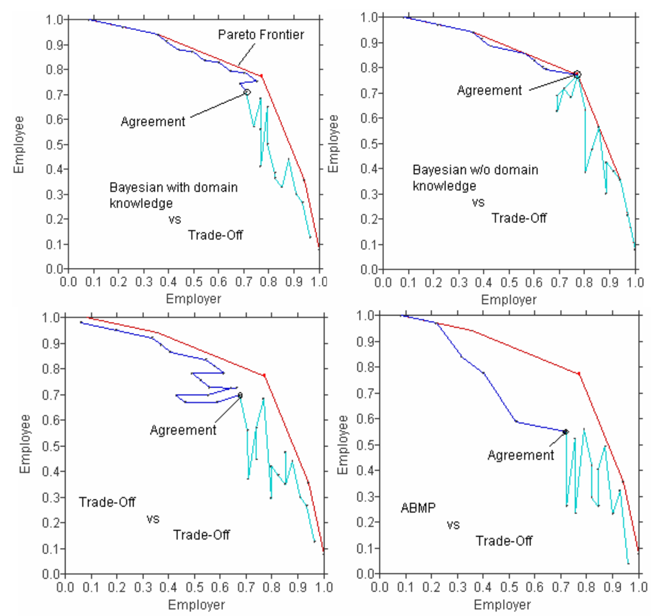

Figure 4. Employee-Employer negotiation domain.

In this domain, the Bayesian agents very efficiently learn issue weights when they are provided with domain knowledge, indicated by the fact that the negotiation trace almost coincides with the Pareto frontier. But even without domain knowledge the

Bayesian agent needs little time to learn the issue evaluation functions and consecutively improves the weight estimations. The Trade-off strategy, which uses domain knowledge but simply assumes that issue priorities are uniformly distributed, makes a number of unfortunate steps in this domain due to the fact that different issues are important to each party. Finally, the ABMP strategy is clearly outperformed by the strategy using Bayesian learning and almost uniformly concedes on all issues without considering the opponent's weights. ABMP lacks the capability of trading-in less important issues for more important ones. Since the Trade-off strategy is influenced by the efficiency of the opponent's strategy, it moreover performs less efficient against the ABMP strategy. Note that only the Bayesian agents were able to reach an agreement close to the Pareto efficient frontier.

Due to space limitations, we only provide the utilities of the agreements reached in the second domain, the Service-oriented negotiation (SON) domain from [5]. This domain has four issues, 30 values each (810,000 possible outcomes) and preferences of both parties are strictly opposing on all issues. Table 1 shows the results, where the dealer role is varied and again, for comparison reasons, an agent using the Trade-Off strategy was used to play the buyer role. The negotiation traces do not add much information compared to the previous domain, although the Trade-off strategy performs better on this domain. The results provide evidence that the learning approach performs consistent over various domains.

Strategy of Dealer	Utility of the outcome	
	Dealer	Buyer
Bayesian with domain knowledge	0.83	0.76
Bayesian	0.83	0.76
Trade-Off	0.78	0.77
ABMP	0.64	0.56

Table 1. Negotiation outcomes in the SON domain.

5. SCALABLE LEARNING ALGORITHM

In this section, the learning approach is refined and an outline of a scalable algorithm is discussed. The experimental results of the previous section clearly demonstrate the effectiveness of the approach outlined in Section 3. Here our main concern will be the size of the hypothesis space $H = H^w \times H^e_1 \times H^e_2 \times \ldots \times H^e_n$. This space is exponential in the number of issues and consists of $n! \cdot m^n$ hypotheses where m denotes the number of evaluation function hypotheses (see Figure 1). Clearly, even though the approach is very effective in small domains, it is not computationally feasible to update this many hypotheses in larger negotiation domains. In order to deal with larger domains, some additional independence assumptions will be introduced. As is to be expected, this will impact the performance of the learning algorithm, but we will present additional experiments that show improved performance compared to that of the other strategies discussed here.

To enable scaling of the proposed learning approach for negotiation domains of high dimensionality it will be assumed that the probability of individual components of a hypothesis $h=\langle h^w, h^e_1, \ldots, h^e_n \rangle$ about a complete preference profile can be learned independently. That is, it will be assumed that weight ranking hypotheses h^w and the shape of each issue evaluation function h^e_i can be learned independently from each other. This is a reasonable approximation since each bid may be presumed to give at least some information about one issue relative to the available knowledge about the other issues.

First, we will explain how each of the evaluation function hypotheses can be learned independently. The idea is illustrated in Figure 5. Figure 5(a) shows the approach outlined in Section 3 as a Bayesian network whereas Figure 5(b) illustrates how the independence assumption can be exploited to split up each hypothesis into its components and add these as nodes to the network. To simplify the notation we assume that symbol $h^e_{i,j}$ can be applied to a bid as a function and results in an evaluation value of the bid according to the evaluation function of hypotheses j for the issue i. The size of the local probability distribution table of each hypothesis in the original approach is $n! \cdot m^{n-1}$. In the approximation method, which introduces additional nodes for every hypothesis, the size of such a local probability distribution table is only m. Each of these additional nodes represents an expected value of the evaluation function for a given bid:

$$\bar{h}_i^e(b_t) = \sum_{j=1}^{m} P(h^e_{i,j}) \cdot h^e_{i,j}(b_t)$$

Second, we need to consider an approximation method for learning weight ranking hypotheses. Note that the number of possible weight orderings is $n!$ which is prohibitive for large n. To reduce the number of weight ranking hypotheses the normalization requirement associated with weights is relaxed. Instead of $n!$ hypotheses a set of m hypotheses for each weight is introduced, where each hypothesis represents a possible value of the weight. Similar to the hypotheses for evaluation functions we introduce the symbol $h^w_{i,j}$ to denote the hypothesis about the value of the weight for issue i according to hypothesis j, and will also sometimes use it to denote the value of the associated weight, i.e. $h^w_{1,1}=0,\ h^w_{1,2}=0.1,\ h^w_{1,3}=0.2, \ldots$. Then, the expected value of an issue weight can be calculated as follows:

$$\bar{h}_i^w = \sum_{j=1}^{m} P(h^w_{i,j}) \cdot h^w_{i,j};$$

The nodes of expected values for evaluation functions and weights are used to update local probability distributions only. The expected utility of a bid b_t is now calculated as follows:

$$\bar{u}(b_t) = \sum_{i=1}^{n} \bar{h}_i^w \cdot \bar{h}^e_{i,j}(b_t)$$

Since a utility function is assumed to be linearly additive this approximation of weight ranking hypotheses does not influence the selection of a bid that maximizes the opponent's utility (when computing a counteroffer). However, the approximation may affect the prediction of the utility of an opponent's bid thus influencing the quality of learning when updating the probability of the hypotheses in line with the conditional distribution associated with the opponent's tactics.

Now we proceed and show that this approximation solves the scalability problem. Note, that instead of normalizing probabilities over complete set of possible utility spaces the probability distribution over weights and evaluation functions are normalized for every issue:

$$\sum_{j=1}^{m} P(h^w_{i,j}) = 1, i = 1,\ldots,n \ ; \ \sum_{j=1}^{m} P(h^{\grave{e}}_{i,j}) = 1, i = 1,\ldots,n$$

Taking this into account, we can show that the expected utility of a bid is the same as in the original approach when the same a priori probability distributions are used. The main idea concerns the modification of the learning itself, i.e. the update of the probabilities associated with hypotheses about single weights and

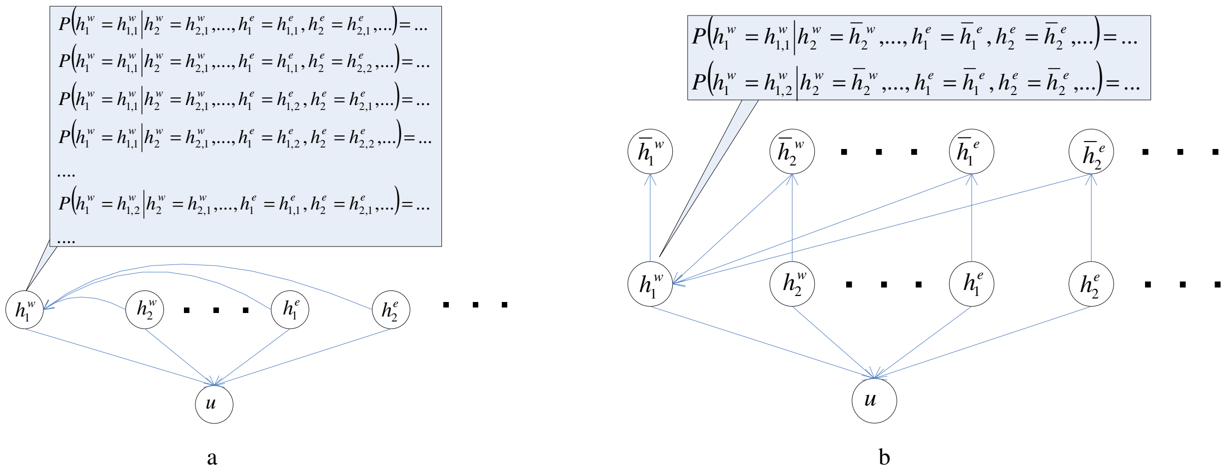

Figure 5 – Bayesian network representing learning probabilities (a) over complete preference profiles hypotheses and over (b) individual hypotheses for weights and shapes of evaluation functions.

evaluation functions of single issues. Instead of calculating the probability distribution for a given hypothesis with respect to all possible partial opponent models we now use the best prediction (or expected value) of the *current* model. In other words, the probability distribution of a hypothesis is estimated by using the probability distributions provided by the model learned so far. The update of the probability of a hypothesis thus assumes that these probability distributions of other hypotheses yield a reasonably good prediction of the opponent's preferences.

It can be shown that if this is the case, the obtained probabilistic model would correspond to the same model built for the hypothesis space over complete preference profiles. In other words, we can show for $h_k \in H$ that:

$$P(h_k) \leftarrow \prod_{i=1}^{n} P(h_{i,j}^w \in h_k) \prod_{i=1}^{n} P(h_{i,j}^e \in h_k), \quad h_k \in H$$

It thus is clear that the approach will greatly benefit from the use of partial domain knowledge when available. In that case, the update of the probability distribution associated with a hypothesis would not be based on probabilistic information associated with the opponent model but on given domain knowledge.

5.1 Updating Probabilities of Hypotheses

Because the first bid has maximal utility for a negotiator according to one of the rationality assumptions introduced earlier, this bid does not provide any information about an opponent's issue priorities. The first bid thus only can be used to update probability distributions of hypotheses about an opponent's evaluation functions and the probability distributions of hypotheses about weights can be updated only after the agent has received more than one bid from an opponent.

Taking this into account, the conditional distribution associated with tactics can be used to update the hypothesis of issue k using the expected evaluation values and weights of the rest of the issues as defined by the current opponent model. So, suppose we need to update the probability distribution of the hypothesis for issue k after receiving a bid b_t from the opponent. In order to do so, we introduce a partial expected utility $\bar{u}_{<-k>}(b_t)$ of bid b_t that does not take the contribution of issue k to the utility of the bid into account, and is defined as follows:

$$\bar{u}_{<-k>}(b_t) = \sum_{i=1,2..,k-1,k+1,...n} \bar{h}_i^w \cdot \bar{h}_{i,j}^e(b_t)$$

The probability of the hypotheses over the shape of the evaluation function can then be updated according to Bayes' rule as follows:

$$P(h_{k,j}^e | b_t) = \frac{P(h_{k,j}^e) P(\bar{u}_{<-k>}(b_t) + h_{k,j}^e \bar{h}_k^w | h_{k,j}^e)}{\sum_{i=1}^{m} P(h_{k,i}^e) P(\bar{u}_{<-k>}(b_t) + h_{k,i}^e \bar{h}_k^w | h_{k,i}^e)}, \quad j = 1,...,m$$

where $\bar{h}_k^w$ is the expected value of the weight of issue k.

The probability of the hypotheses related to the weight of issue k can be updated in a similar way as follows:

$$P(h_{k,j}^w | b_t) = \frac{P(h_{k,j}^w) P(\bar{u}_{<-k>}(b_t) + h_{k,j}^w \bar{h}_k^w | h_{k,j}^w)}{\sum_{i=1}^{m} P(h_{k,i}^w) P(\bar{u}_{<-k>}(b_t) + h_{k,i}^w \bar{h}_k^w | h_{k,i}^w)}, \quad j = 1,...,m$$

Because the application of Bayes' rule to multiple hypotheses needs to be implemented as a sequential procedure, care should be taken to perform a Bayesian update by using the expected utility, weights and evaluation values that are derived from the probability distribution before any Bayesian update has been performed. Otherwise, any hypotheses that are updated after other hypotheses have been updated would be biased by the updated probability distributions of these hypotheses that already have been updated. Additionally distributions of a priori probabilities have to be adjusted in such a way that the sum of the expected values of the weights equals one, i.e.:

$$\sum_{i=1}^{n} \bar{h}_i^w = 1$$

5.2 Experimental Results

In this section, additional results are presented to demonstrate the effectiveness of the scalable learning algorithm on larger domains. The same experimental setup is used as in Section 4.1. However, a more complex domain is used: the AMPO vs. City domain of [11], which consists of 10 issues, 5 values in average each (total of 7,128,000 possible outcomes). The results on this domain presented in Figure 6 show, as is only to be expected, that it becomes harder to stay close to the Pareto efficient frontier. The performance of the Bayesian learning agents is now similar to that of the agent based on the Trade-Off strategy and both stay close to the Pareto frontier. The ABMP strategy shows similar behaviour as on the earlier negotiation domains, and is outperformed by the other strategies. The results thus are still very good. Also, note that the agreement reached by the Bayesian agents has a higher utility than that reached by the other strategies and that both the Bayesian agent without domain knowledge as well as the Trade-off agent make quite big unfortunate steps.

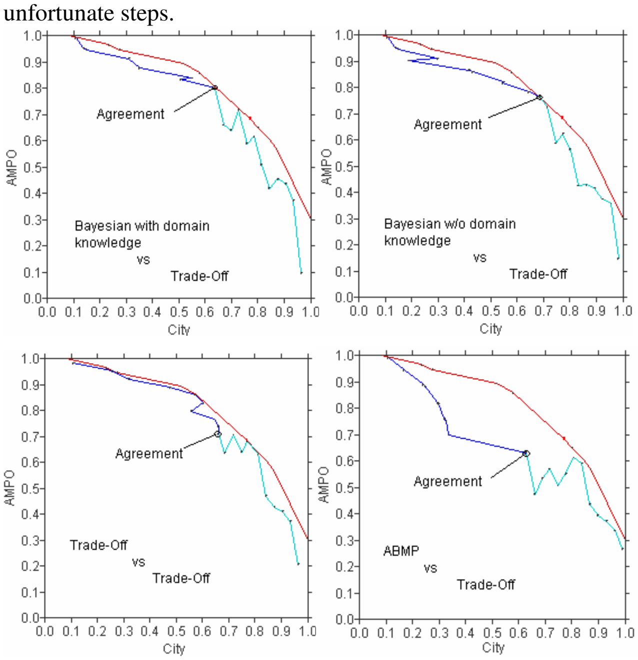

Figure 6. Negotiation dynamics for the AMPO vs. City domain.

6. CONCLUSIONS AND FUTURE WORK

In this paper, an opponent modelling framework for bilateral multi-issue negotiation has been presented. The main idea proposed here to make opponent modelling in negotiation feasible is to assume that certain structural requirements on preference profiles and on the strategy of an opponent are in place. Due to the probabilistic nature of the model, these assumptions still allow for a great diversity of potential opponent models.

The learning approach has been tested on several domains to demonstrate the effectiveness of the approach. The results moreover showed the effectiveness of using an opponent model in a negotiation strategy to improve the efficiency of the bidding process. In the future work we will analyze the quality of the learned opponent modelled with respect to the original preferences profile of the opponent. We will investigate influence of the negotiation domain, preference profile, and opponent's strategy on the quality of the learning.

The learning approach does not rely on prior knowledge about e.g. the domain, but if such knowledge is available it can be incorporated and used to initialize probability distributions in the opponent model. However, domain knowledge would be useful to increase the efficiency of learning a correct opponent model in the scalable learning algorithm proposed.

One interesting line of future research is to test and initialize the learning algorithm for specific domains with an "average preference profile" derived from (large sets) of negotiator profiles for that domain. It is expected that performance of the algorithm on specific domains can be further enhanced. We are currently setting up an experiment to collect preference profiles for a negotiation domain and will test how our learning algorithm performs when it is initialized with such an aggregated profile.

Another direction for future research concerns the hypothesis space used in the opponent modelling framework. Although we think the evaluation functions proposed in this paper provide a good basis for approximating many preference profiles in practice other choices of function types might prove more effective in certain domains.

7. REFERENCES

[1] Bosse, T., and Jonker, C.M. 2005. Human vs Computer Behaviour in Multi-Issue Negotiation, Proc. of the 1st Int. Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RSS'05), pp.11-24.

[2] Coehoorn, R.M., Jennings N.R. 2004. Learning an Opponent's Preferences to Make Effective Multi-Issue Negotiation Trade-Offs, In Proc. of 6th International Conference on E-Commerce, pp. 59-68.

[3] Hindriks, K., Jonker, C. M., Tykhonov, D. 2007. Analysis of Negotiation Dynamics, In: Proc. of 11th Int. Workshop CIA 2007 on Cooperative Information Agents, Delft, The Netherlands, Springer-Verlag, Lecture Notes in AI, vol.4676.

[4] Faratin, P., Sierra, C., and Jennings, N. R. 1998. Negotiation Decision Functions for Autonomous Agents, in: Int. Journal of Robotics and Autonomous Systems, 24 (3-4) 159-182.

[5] Faratin, P., Sierra, C., and Jennings, N. R. 2003. Using Similarity Criteria to Make Negotiation Trade-Offs, Journal of Artificial Intelligence, 142 (2), pp. 205-237.

[6] Jonker, C., Robu, V., Treur, J. An Agent Architecture for Multi-Attribute Negotiation Using Incomplete Preference Information, In: Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS), Springer-Verlag (in press, 2006)

[7] Lin R., Kraus, S., Wilkenfeld, J., Barry, J. 2006. An Automated Agent for Bilateral Negotiation with Bounded Rational Agents with Incomplete Information. ECAI 2006, pp. 270-274.

[8] Narayanan, V., Jennings, N.R. 2006. Learning to Negotiate Optimally in Non-stationary Environments, CIA 2006, LNAI 4149, pp. 288-300.

[9] Niimura, T., Ozawa, K., Hirano, O., 2001. Fuzzy Set-based Decision Support System for Transactions of Electricity in A Deregulated Environment, IFSA World Congress, pp. 87-92.

[10] Osborne, M. J. and Rubinstein, A. 1994, A Course in Game Theory, The MIT Press.

[11] Raiffa, H. 1982. The Art and Science of Negotiation, Harvard University Press.

[12] Rosenschein, J.S., and Zlotkin, G.1994. Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. MIT Press.

[13] Zeng, D., and Sycara, K. 1998. Bayesian Learning in Negotiation, International Journal of Human Computer Systems, vol. 48, pp. 125-141.

[14] Zeng, D., and Sycara, K. 1997. Benefits of Learning in Negotiation, in Proc. of the Fourteenth National Conference on Artificial Intelligence (AAAI-97), Providence, RI.