A Bayes Net Approach to Argumentation Based Negotiation

Sabyasachi Saha and Sandip Sen

Department of Mathematical & Computer Sciences,
University of Tulsa,
Tulsa, Oklahoma, USA
{sabyasachi-saha, sandip-sen}@utulsa.edu

Abstract. Negotiation is one of the most fundamental and effective mechanism for resolving conflicts between self-interested agents and producing mutually acceptable compromises. Most existing research in negotiation presumes a fixed negotiation context which cannot be changed during the process of negotiation and that the agents have complete and correct knowledge about all aspects of the issues being negotiated. In practice, the issues being negotiated may change and the agents may have incorrect beliefs of relevant issues updated during the negotiation process. Argumentation-based negotiation approaches have therefore been proposed to capture such realistic negotiation contexts. Here we present a novel Bayesian network based argumentation and decision making framework that allows agents to utilize models of the other agents. The agents will generate effective arguments to influence the other agent's belief and produce more profit.

1 Introduction

Agents deployed for real-world applications like electronic commerce, recommender systems, and personal assistants have limited, specialized capabilities and have to depend on other agents to achieve their goals. They often interact in an open environment with other agents or humans. Agents with conflicting interests need to negotiate to improve profits. Negotiation allows agents to reach a mutually acceptable agreement.

Automated negotiation has drawn significant attention in Multiagent systems research [3, 8, 13]. Negotiation is viewed as a distributed search through a space of potential agreements [3]. Existing frameworks allow agents to propose counter offer in addition to accepting or rejecting the previous offer. Offers include attributes which belong to some pre-fixed *issues*. Agents are assumed to have correct and complete knowledge of preferences and the negotiation context as well as agents' preferences are held constant during the course of negotiation. In real-life negotiation scenarios, however, the participating individuals do not have fixed preferences. Also, at times they might have incorrect belief about the world or may not be cognizant about all pertinent attributes. In such situations,

I. Rahwan et al. (Eds.): ArgMAS 2004, LNAI 3366, pp. 208–222, 2005.
© Springer-Verlag Berlin Heidelberg 2005

agents can influence each other by argumentation with convincing, relevant information. The existing game theoretic or heuristic based approaches do not provide a framework for argumentation-aided negotiation.

In last few years argumentation based framework for negotiation is discussed in Multiagent systems research [4, 10]. Most existing argumentation based negotiation frameworks are logic or rule-based [8, 9, 11]. While these approaches provide a formal framework with provable properties, we believe there is a need for alternative frameworks that can better capture the uncertainty and complexity of real-life negotiations. In particular, the factors influencing an agent's decisions may be incompletely known and be gradually revealed to a negotiator. Accordingly, negotiation frameworks should incorporate approximate opponent models represented in a form that can capture complex relationships between domain attributes and can be efficiently updated based on information revealed during negotiation. The specific research questions we are interested in include the following:

• When processing an offer or a counter-offer, what decision mechanisms should an agent use to decide whether to accept a proposal, argue about its last proposal, or generate a new proposal?
• How are arguments for negotiation generated/selected?
• Should an agent try to persuade the other agent by reward, threat, etc.
• How and when does an agent update its belief about the other agent or about the negotiation issues based on received arguments and offers?
• How does the agent's model of the opponent influence its argumentation and proposals?

In this paper, we present a decision architecture of the arguing agent. We propose to use a Bayesian network model [5] to represent the influences of different factors on agent decisions. An agent's knowledge of such causal factors and their relative importance is captured in the topology of the network as well as the prior and conditional probability assignments. Initial, approximate knowledge of an agent can be further refined based on actual negotiation experiences. If values of all the factors are known, then the actual decision taken by another agent given these factors can be used to update the conditional probabilities at the outcome nodes. If some of the factors values are not known, the decision taken and the values of the known factors can be used to update either the conditional probabilities at the outcome nodes or the prior probabilities of the unknown factors. In this paper we focus on the decision mechanism that allows a modeling agent to use its knowledge to determine negotiation offers and select arguments to influence the opponent to accept offers that it has turned down. The goal is to use the Bayes net model of the opponent to select and manipulate the negotiation context to increase the chance of an favorable offer being accepted by the other party.

Though the general framework of Bayesian network based argumentation can be used in peer-level or symmetric negotiation, we have focused our discussion in this paper on asymmetric scenarios where a knowledgeable domain-expert agent is negotiating with a user agent. Hence we assume that our agent has access to significant domain knowledge that can be used to argue against possibly incorrect assumptions made by the user. This asymmetry also means that the expert agent can use argumentation based on its model of the user agent to influence that agent to accept offers that are preferable to the expert agent. The Bayes net model is the key component that allows the expert agent to select initial offers, respond to counter offers with convincing arguments or with further offers that are acceptable to both parties to the negotiation.

The paper is organized as follows. Section 2 presents a few motivating examples for the types of argumentative negotiation we are interested in. Next, in Section 3, we have discussed relevant research works that have influenced the research in this field and the relationship of our work with these existing reasearch. In Section 5, we present an architecture that allows an agent to choose from and construct from a set of different classes of negotiation arguments based on a Bayesian network based opponent model. Following this, we present decision mechanisms that select arguments and offers based on the probabilistic model. We conclude with observations about the strength and applicability of such a coherent and powerful approach to argumentation-augmented negotiation.

2 Argumentation Scenarios

In this section, we use the running example of a negotiation scenario between a travel agent and a customer. Here, we have described the generation of arguments or counter proposals from the travel agent's perspective. At first, we present a conversation that shows the necessity of the argumentation in negotiation process. Then we have produced three more conversations to clarify the importance of modelling opponent's belief.

Consider the following conversation between our domain expert, a travel agent (TA), and another buyer agent (A) who has contacted TA for a ticket from Tulsa to Calcutta on the first week of February.

Conversation 1:
TA: Ticket Offer: $<$ \$1400, # stop 1, waiting hrs = 5, Date 2/4 $>$.
A: Reject because price is high.
TA: I can offer deals as cheap as \$1200 but if you purchase the previous offer you will get a free round trip within continental USA.
A: That's cool. I accept the previous deal.

In this conversation, **TA** has influenced the preference of A by rewarding him with a free RT offer which was not in the original negotiation context. This is an example of negotiation based on arguments. To produce convincing arguments, it becomes extremely crucial to know the opponent's belief model because the same argument may not work for different opponents. Consider the following three conversations:

Conversation 2:
In response to the request for a cheaper deal by another agent **B** for the same itinerary in Conversation 1, the travel agent responds

TA: Unfortunately no ticket below $1400 is available for February 4 and if you delay the price will go up to $1600.
B: OK then give me this deal.

Conversation 3:
The travel agent tries the same "threat" as in Conversation 2 for the same itinerary with another agent **C** who responds
C: Then I am not interested.

Conversation 4:
In response to the request for a cheaper deal by another agent **D** for the same itinerary in Conversation 1, the travel agent responds
TA: I fear I can not give you any ticket below $1400 on February 4 but if you take this deal I can give you 15,000 frequent flier bonus miles.
D: OK then I will purchase the ticket.

In conversations 2 and 3 we find that the same argument can result in opposite results. The agent has missed the deal in the second conversation. For the agent B the fear that the price of the ticket may increase dominates its decision whereas agent C believes that it can get better deals. For agent C the reward offer clinches the sale. Notice that here the travel agent TA need to concede some utility in Conversation 4 to seal the deal. Which of the arguments the TA should use, will depend on available offers, local utility function for the deals and the opponent models that can be used as a predictor for offer/argument acceptance. In reality, even though it is unlikely that the TA will have an exact knowledge of the user agent's belief model, such models can be approximated from domain knowledge, interactions with other agents and previous interactions with this agent. We propose a Bayesian network based approach for opponent agent modeling.

The above-mentioned negotiations are based on a set of *issues*, e.g., *price, # stops, waiting time, departure date, destination city, departure city, etc.*. Some of these issues are negotiable and some of them are constraints and can be determined from domain knowledge. In the conversation 4, though *bonus miles* was not part of the original set of issues being negotiated, the TA may have the model that it can be used as a leverage on agent D. In other scenarios an agent may have incorrect belief about some attributes. For example, a customer agent G has a belief that airlines E has poor luggage handling record. When an offer is rejected based on this premise, the TA will need to argue to correct this misconception. This may, in turn convince G to accept the proposed deal.

3 Related Work

When we talk about negotiation process or argumentation in the negotiation process in a multiagent society, it becomes extremely important to decide upon communication language, domain language and negotiation protocols [1]. Common agent languages like FIPA ACL does not provide all the locutions which are required to capture necessary expressions in the negotiation process. So, researchers introduce explicit locutions for expressions [14]. In this paper, we concentrate on the argumentation generation in the negotiation process and handle the communication introducing some explicit terminologies.

In a recent survey [10], Rahwan *et. al.* has presented a clear current state of research in argumentation based negotiation. They have compared different existing frameworks in the light of main characteristics. Kraus *et. al.* address the problem of argumentation negotiation as a multiagent system problem and proposed a framework for persuasion [8]. In their framework, agents used threats, rewards, etc. as argumentation. They assume a prefixed set of arguments. Some research has focused on providing framework for argumentation based negotiation [4, 7]. They have mainly focused on the protocol for negotiation. Parsons et. al. design it as a finite state model [9]. Our work is quite different from these work, as unlike others we have concentrated on decision mechanism of the agents. Ramchurn et. al. proposed a fuzzy logic based approach for selecting rhetorics for persuasion [12]. They have addressed the problem of negotiation process. They evaluate different locutions based on utility and trust. Rahwan et. al. proposed a goal based approach for argumentation [11]. He argues that since preferences are adopted to fulfill some goals, the arguing agent can influence the other agent by influencing the associated or subgoals. Our work is also much different from them. In our work, we have dealt with preferences which are subject to change and use a continually updated model of the opponent. Then, we propose a novel Bayesian network based decision mechanism for arguing during negotiation. Since we use expected utility based evaluation of the proposals and arguments, the preference ordering also change during the negotiation process and always produce the proposal which is most suitable at that point of time and with the uncertainty of the domain and acquired knowledge. We believe that using our model it will be possible to address argumentation during a negotiation process in a more rich and dynamic environment. Zukerman *et. al.* have used Bayesian networks to generate arguments in natural language as part of a human-computer interaction scenario [16]. Given a goal proposition by the user, the system, NAG generates arguments to justify it. We are using argumentation to enable a neogtiating agent to strike better deals with a peer agent. As such, the problem solving and communication protocol, as well as the nature of arguments are fundamentally different.

4 Definitions

In this section we present the formal definitions of different arguments or offers. $\mathcal{A}$ is the whole set of attributes in the environment. We call $name_i$ and $state_{i,j}$ as the name and jth state, respectively of the ith attribute in the environment, $j = 1(1)n_i$. We assume that each agent will be aware of all possible values of $name_i$ and $state_{i,j}$, $j = 1..n_i$. The domain of $state_{i,j}$ may be numbers or discrete values like *high, low, good, etc.*. We now define the attributes used in negotiation

• $\mathcal{I} \subset \mathcal{A}$ to be the set of current context attributes.
• $\mathcal{E} \subset (\mathcal{A} - \mathcal{I})$ to be the set of additional attributes which are not in the current negotiation context but they can be included in $\mathcal{I}$ during the process of negotia-

tion. That means an element $a \in \mathcal{E}$ can be removed from the set $\mathcal{E}$ and included in the set $\mathcal{I}$.
• $\mathcal{P} \subset (\mathcal{A} - (\mathcal{I} \cup \mathcal{E}))$ to be the set of persuasive attributes which are used for argumentation but are non-negotiable, e.g., *reward-bonus-miles*, *threat-increase-future-price*, etc.
• $\mathcal{V}$ to be the collection of attributes with their name and a particular state value in the outgoing proposal. An agent constructs this set with the name state pairs of the attributes it choose for argument or offer.

We broadly categorize the *locutions* used in the conversation into following types *viz, request for proposal (or req), offer, argument, accept, reject and terminate*. Within each categorization, there are different types of locutions. Here we discuss only the important locutions.

request($\mathcal{V}$): This is asked at the beginning of the conversation where the asking agent states its basic need.
offer($\mathcal{V}$): This may be a completely new offer satisfying all constraints or it may be the one stating the opponent that if it waives one or more constraints this offer matches the other specifications and may be useful to it.
Argument: We define four different types of arguments:
conflict-argument($\mathcal{V}$): This is argument to the opponent about the conflict in belief this agent has about the attributes in the $\mathcal{V}$. It states $\langle name_i, state_i\rangle$ as its belief about the ith attribute in $\mathcal{V}$.
emphasizing-argument($\mathcal{V}$): This is argument to emphasize some additional attributes in the offer to influence the opponent to accept the previous offer. Here $\mathcal{V} \subset \mathcal{E}$.
persuasive-argument($\mathcal{V}$): This argument is used to persuade the opponent. Here $\mathcal{V} \subset \mathcal{P}$.
justification($\mathcal{V}$): This argument is used to justify a previous argument. Here $\mathcal{V} \subset (\mathcal{A} - (\mathcal{I} \cup \mathcal{E} \cup \mathcal{P}))$.
accept($\mathcal{V}$): This is used to accept any proposal from the opponent.
reject($\mathcal{V}$): This is used to reject any proposal from the opponent. $\mathcal{V}$ contains the name state pair of the attributes which are the reason for this rejection.
terminate(): used for termination of the conversation.

5 Architecture of Argumentation Based Agent

In this section, we present the architecture of our agent Ag (now onward we maintain the convention of calling the agent by Ag and the opponent agent by $OpAg$) for negotiation using arguments. Figure 5 shows the different components in the agent architecture. We will discuss the different components here.

Proposal Analysis: The opponent agent can send either a counter offer or it can simply reject previous offer made by the agent. Here we like to emphasize one thing, when an agent rejects one proposal, we assume that it gives some argument for his decision. It can also send *null* argument. This component

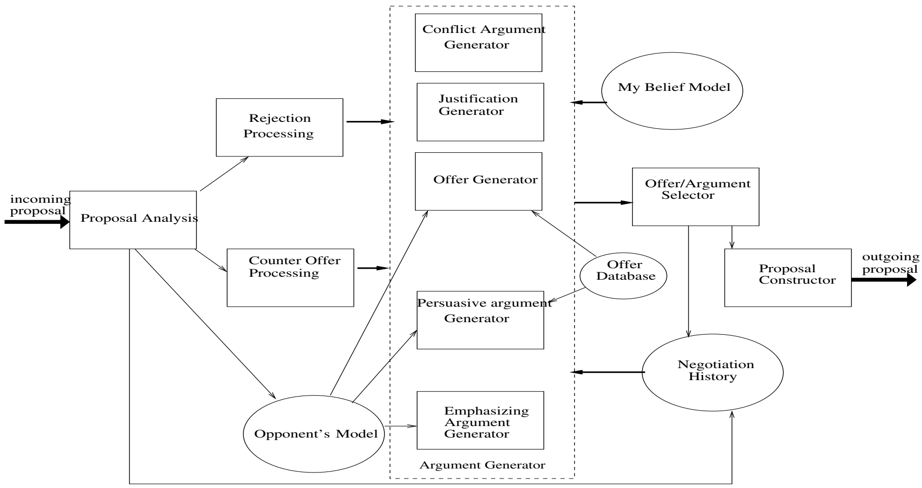

Fig. 1. Decision Architecture of the arguing agent

recognizes opponent's proposal type and update opponent's model. If opponent sent an counter offer it inform the *Counter offer processing* component and if opponent rejects previous offer with some argument it informs the *Rejection processing* component. If the other agent terminate the negotiation after updating the opponent model it terminates the process.

Rejection Processing: It interacts with the *argument generator* to generate argument or another offer to change opponent's rejection.

Counter Offer Processing: It interacts with the *argument generator* to generate argument or counter offer for the opponent agent.

Argument Generator: It has five subcomponents which are responsible to generate arguments or counter offer or rejection for the opponent. They are *Conflict Argument Generator, Justification Generator, Emphasizing Argument Generator, Persuasive Argument Generator* and *Offer Generator*. It starts working with *Conflict Argument Generator*.

Conflict Argument Generator: This subcomponent decides if the opponent agent rejects the earlier offer because of any "wrong" or conflicting belief about some attribute. Based on the negotiation history it determines whether to argue with this conflict. If it decides to argue with this conflict it informs the *Offer/Argument Selector*, which is described later in this section, to generate *conflict-argument* with the attribute(s) it finds conflict, otherwise, it relinquish the control to *Justification Generator*. For example, opponent agent may have a belief that E airlines has a poor luggage handling reputation. So, when the travel agent offer a ticket with E airlines in a reasonable price, G rejects that with an argument of $\langle$*luggage handling: poor*$\rangle$. Then travel agent needs to argue with confidence that in recent years E does not have any complain of poor luggage handling. Then opponent agent may accept this deal. And if it rejects again then this component updates the opponent model with this as a constraint.

Justification Generator: If the opponent rejects the previous offer and send name-state pair of a set of attributes, say W, as the reason. If there is no conflict in belief but the agent finds that there is one more attributes v_1 in the environment ($\notin (\mathcal{I} \cup \mathcal{E})$) which influence some attributes in W and v_1 is not under the control of any agent, then the agent asks *Offer/Argument Selector* to justify the previous proposal with v_1. For example, if an agent reiterates that the *price* is high then this agent can justify it as the ⟨ peak-season?: yes⟩. If this sumcomponent does not find any justification to make it calls the following subcomponents to take control.

Emphasizing Argument Generator: Based on the *Opponent model*, discussed later in this section, this subcomponent decides if there is any *emphasizing-argument* which can influence the opponent's decision. If it finds any such argument it send that to *Offer/Argument Selector* to form *emphasizing-argument.* Later we will discuss in detail which attributes are chosen and how. This argument may seem unnecessary as the agent could have sent all the attributes along with the attributes in $\mathcal{I}$. But in practice, an agent slowly expands the context, if possible. This gives the other agent a feeling that it is offered as a benefit. Moreover, in some real life negotiation, the number of issues that may influence it is large and uncertain. So, initially, $\mathcal{I}$ is chosen as the combination of the attributes that the other agent precisely mentioned and other dominant attributes known by this agent from the domain knowledge. Then $\mathcal{I}$ is changed during negotiation based on the interaction of the agents. For example, suppose *Ag*'s proposal of ⟨ \$1300, # stop = 1, waiting hr. = 15 ⟩ is rejected by *OpAg* with an argument of that ⟨ waiting hr. =15, problematic: yes ⟩. Now, from the belief model of *OpAg*, agent *Ag* knows that the notification that ⟨ hotel facilities: 5 star, accommodation: free ⟩ with the previous offer will decreasing the probability of ⟨ problematic: yes ⟩ and in turn increase the probability of acceptance of the last offer.

Persuasive Argument Generator: This subcomponent decides with the help of *Opponent model* if there is any persuasive argument which can influence the opponent to accept the previous offer. For example, suppose a reward of bonus miles has a very positive influence on the opponent. Then if you reward a reward of 10,000 bonus miles, it may accept your previous offer. Sometimes threat about rising price may cause an acceptance of the offer which the opponent previously rejected. It sends the persuasive arguments *persuasive-argument* to the *Offer/Argument Selector.*

Offer Generator: This subcomponent generates the offer which it finds best. If there is no offer that matches the conditions given by the opponent then it also finds the offers which is possible if some weak constraints are removed. It compares the best offer with the offer of the opponent, if any. It informs the *Offer/Argument Selector* which one is better and if the opponent needs to drop any attribute.

Offer Database: This is a filtered repository of all the offers relevant to the opponent. This also includes the offers with *weak* constraints. After each interaction if the agent finds some new *strict constraints* in the opponent's model it updates the *offer database*.

My Model: This is a collection of the agent's own belief about the domain attributes. Say for example, it has a knowledge of the services in the airlines, luggage handling, flight security, crew service quality, insurance facilities, etc. Belief about an attribute may be strong or weak. A weak belief may change hearing some strong counter arguments from someone whom it trusts. For a domain expert (e.g. here the travel agent) we assume that all the belief are *strong*.

Opponent Model: Opponent model consists of *Constraint information* and *Opponent's belief model*. We believe that, in any negotiation it is important to recognize which attributes are *strict* constraints and which are negotiable. If *OpAg* asks quote for tickets from JFK airport, NY to London and the travel agent offer him cheap deals from NY to Shanghai, it will be enough reason for the agent *OpAg* to terminate the conversation. But if it is difficult to find deals from JFK, NY to London it will be a reasonable suggestion to try from another airport of NY. Using the domain knowledge we initially classify the constraints. Then it is updated based on the response from the opponent. We present *Opponent's belief model* as a Bayesian network and it will be discussed in the next section.

Negotiation History: This consists of the history of offers and arguments from both the agents.

Offer/Argument evaluator: This is an implicit component of the architecture. Each of the above three argument uses this component. Based on the Opponent's belief model and the agent's own evaluations of the corresponding offer or persuasion, this component finds out the expected utility of the offers or arguments.

Offer/Argument Selector: This component chooses the offer or argument producing maximum expected utility. It compares among the offers or arguments which are sent by different argument and offer generator subcomponents. If the opponent's counter offer is the most profitable producing maximum utility then it asks the *Proposal Constructor* to accept the negotiation. If the offer/argument generated exceeds the opponent's proposal then send that to *Proposal Constructor* and if it does not receive any profitable offer from the other components it asks the *Proposal Constructor* to terminate the negotiation.

Proposal Constructor: This forms the outgoing proposal and send it to the opponent agent.

6 Bayes Net Model of Opponent's Belief

In section 5, we have briefly discussed the architecture of a negotiating agent. We have described how the decision mechanism largely depends on the agent's approximation of the opponent's model. We have discussed that, one proposal may be very quite profitable for one opponent but may be unacceptable for another opponent. This makes it necessary and desirable for the negotiating agent to model its opponent. In practice, one agent may have only approximate *a priori* estimates of the dependencies and influences of the different factors on the other agent's behavior. We propose the use of Bayesian networks to capture the causal dependencies of the different factors on the decision mechanism of the opponent. Bayesian networks can capture the inherent uncertainty in the domain. We use an augmentation of the Bayesian network to evaluate the utility of different actions of the modeling agent. The extended network is known as *influence diagram.* This mechanism will allow the modeling agent to choose the action that will produce maximum expected utility. We have shown an example of modeling the opponent's belief in Figure 2. We will discuss the details of the decision mechanism in the next section.

6.1 Bayesian Networks and Influence Diagram

A Bayesian network is a graphical method of representing causal relationships [5], i.e. dependencies and independencies among different variables that together define a real-world situation. Technically it is a Directed Acyclic Graph (DAG) with nodes as the variables and a directed edge represent a causal relationship between the corresponding nodes. In addition to its structure, a Bayesian network is also specified by a set of parameters θ, that qualify the network. The causal relationship is characterized by the corresponding conditional probability tables (CPTs).

Consider a vector X of variables and an instantiation-vector x. If the immediate parents of a variable X_i is the vector Pa_{X_i}, with its instantiation pa_{x_i}, then

$$Pr[X = x|\theta] = \prod_i Pr[X_i = x_i|Pa_{X_i} = pa_{x_i}, \theta].$$

This defines the joint distribution of the variables in X, where each variable X_i is conditionally independent of its non-descendants, given its parents. For more detailed discussion on Bayesian networks we refer [2, 5].

We use Bayesian networks for representing belief structures, for the following reasons:

- Bayesian network can readily handle incomplete data sets.
- It allows one to learn about causal relationships. This is useful to gain understanding about a problem domain. It successfully represent the non-linear causal relationships of the variables.
- It handles uncertainty in the domain efficiently.
- Bayesian networks in conjunction with Bayesian statistical techniques facilitate the combination of domain knowledge and data.
- It offers a method of updating the belief or the CPTs.

An influence diagram is a Bayesian network augmented with action variables and utility functions. There are three types of nodes in the influence diagram: *chance nodes, value nodes* and *action nodes* [6]. The action variables represent different actions of the decision maker. There exists utility function attached to the value nodes in the network. Influence diagram can be used to calculate the utilities of different values of the decision variables. In the context of negotiation, we want to use such networks to find out the conditional probability of accepting a proposal given the proposal contents listed as an attribute-value vector.

6.2 An Illustration of the Agent Belief Model

In our negotiation framework, we assume that the arguing agent has an approximation of the belief model of the opponent agent. In this paper, we model the opponent's belief as a Bayesian belief network. We have assumed that the arguing agent knows the exact structure of the network and it has an approximate idea of the conditional probability tables (CPT) from the domain and earlier interactions with the opponent agent. For the sake of simplicity we have assumed all variables to be discrete. In figure 2, we show an example of modeling opponent agent's belief. It shows the model of a customer agent *OpAg* approximated by the travel agent *Ag* in our example. The agent *OpAg* has asked for a round-trip airline ticket from Tulsa to Calcutta in the first week of February. He send the request for proposal *req*(V) where V is collection of the attributes ⟨ from: Tulsa, To: Calcutta, Roundtrip: yes, # of stops: $\leq$ 2, date: 02/04/04⟩. In the network shown, the decision node represents the decision whether *OpAg* accepts the offer or argument. The outcome is boolean, yes or no. The chance nodes, value nodes and action nodes are represented as circles, rhombus and rectangles, respectively. Double circles imply that the offer deterministically determines their values. The double circles that are joined by a solid line with their parents implies that they are initially among the set $\mathcal{I}$ of the negotiation and the double circles joined

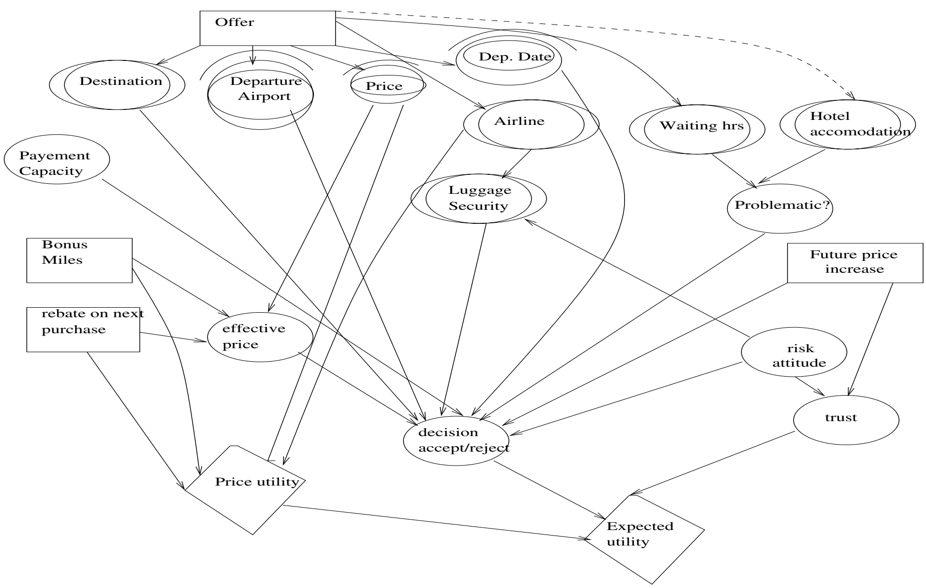

Fig. 2. Approximate belief model of the opponent

with dotted lines belong to $\mathcal{E}$ in the negotiation. An arc above some double circles qualify them as *negotiable*. The action nodes represent the actions taken by agent Ag and its influence on $OpAg$'s decision is represented by the CPT's.

Here the action *offer* by the agent Ag determines the value of different nodes like *airlines name, # of stops, date of journey, destination city, departure city, requirement for transit visa, etc.* Whereas, Ag has a belief about the value of some chance nodes for $OpAg$ like *risk attitude, payment capacity, etc.* which influence the decision of $OpAg$. We consider different offers as different possible actions for the node *offer*.

For each offer available to Ag it can find out a conditional probability of acceptance by $OpAg$ given evidences in the offer. Based on the reply of $OpAg$ to an offer or argument the CPTs are updated by the sequential update rule of Bayesian network [15].

7 Offer or Argument Selection Procedure

In section 5, we discuss the different components that influence the decision of the agent Ag. In this section, we will present an algorithm to clearly state the decision taking procedure. Also we will discuss briefly how the agent evaluates different and choose the offer or argument.

The decision procedure is presented in Algorithm 1. Now lets describe some methods of the algorithm.

• **Find-best-offer:** Each offer sets different values to the attributes. From agent's own model it will get a utility for the offer with corresponding values of the attributes. It will also get the corresponding probability that this will be accepted by the other agent from the opponent agent's belief model presented by the Bayesian network. So, for a each offer with specified values of the attributes, the agent can find the expected utility. The offer generator will choose the one producing maximum expected utility. If it is not possible to find an offer for the constraints, it will check if removing some weak constraint can yield some good deal. If so, it chooses that deal as the best offer.

• **Conflict-belief:** If the opponent agent rejects some proposal and produces some attribute which has conflict with the belief of this agent. The subcomponent *Conflict Argument Generator* generates *conflict-argument* with the belief it has about the conflicting attribute.

• **Justify-belief:** This argument is used if the opponent rejects the offer and send name-state pair of a set of attributes, say W, as the reason. If there is no conflict in belief but the agent finds that there is an attribute v_1 in the environment ($\notin (\mathcal{I} \cup \mathcal{E})$) which influence some attributes in W and v_1 is not under the control of any agent, then the agent send justify with $v_1 \in \mathcal{V}$. This is done by *Justification Generator*.

•**Find-significant-emphasizing-argument:** If the opponent agent rejects or produces counter offer the agent will emphasize on those attribute values which have significant influence on the opponent but not mentioned in the

Algorithm 1 Decision(proposal($\mathcal{V}$)): Decision Algorithm of the Agent

Update-opponent-model($\mathcal{V}$);
if proposal($\mathcal{V}$) is reject or argument **then**
  Process-Rejection-Processing($\mathcal{V}$);
else {proposal($\mathcal{V}$) is counter offer}
  Counter-Offer-Processing($\mathcal{V}$)

Process-Rejection-Processing($\mathcal{V}$):
finalarg = null;
finalarg = conflict-belief($\mathcal{V}$,myModel,NegoHistory);{This is done by *Conflict Argument Generator*}
if (finalarg==null) {there is no conflict in belief} **then**
  finalarg = Justify-belief($\mathcal{V}$,myModel,NegoHistory);{*Justification generator* controls this method}
if (finalarg==null) {no additional justification for offer} **then**
  finalarg = OfferOrPersuasiveOrEmp(m,null);{this method which option generates maximum expected utility}
  {where, m =$\langle\mathcal{V}$,myModel,OpModel,NegoHistory$\rangle$}
  {null second argument corresponds to no counter-offer from opponent}
Offer-argument-select(finalarg);{final proposal formed}

Counter-Offer-Processing($\mathcal{V}$):
finalarg = null;
finalarg = Justify-belief($\mathcal{V}$,myModel,NegoHistory);
if (finalarg==null) {no additional justification for offer} **then**
  finalarg = OfferOrPersuasiveOrEmp(m,proposal($\mathcal{V}$);

OfferOrPersuasiveOrEmp(m,proposal($\mathcal{V}$)):
$\langle u_2, arg_1\rangle$ = Find-significant-emphasizing-argument(m);{this method finds the significant emphasizing argument$\in \mathcal{E}$}
$\langle u_3, arg_2\rangle$ = Find-best-persuasive-argument(m);{this method finds out best persuasive argument and corresponding expected utility}
$\langle u_4, offer\rangle$ = Find-best-offer(m);{finds out the best offer}
u_1 = getutility(proposal($\mathcal{V}$));{Opponent's offer utility}
find out which utility, u_i is maximum.
if max u_i is positive **then**
  Offer-argument-select(finalarg) {finalarg is the generated argument or offer or opponent proposal for which the corresponding u_i is maximum}
else
  Terminate()

Offer-argument-select(finalarg):
form outgoing proposal based on finalarg.

negotiation. This arguments reinforce the negotiation context to make the offer more acceptable to the opponent.

So, the task is to find out significant attributes for argumentation. Now, given the evidences in the offer, we need to find out those attributes ($\in \mathcal{E}$) in the offer which has significant influence but not yet included in $\mathcal{I}$. Suppose, x_i's are the values already in $\mathcal{I}$. We need to calculate $P(acceptance|\mathcal{I}, y_{j,k})$, from the

Bayesian network of the opponent's belief,where $y_{j,k}$ is the kth value of $y_j \in \mathcal{E}$, for different j's.

Having obtained this probabilities we can test of significant of the influence of the y_j in the acceptance of the other agent. Conduct a t-test for the test of significance of individual probabilities. We consider the null hypothesis as H_{0i} : i^{th} *varianle in* $\mathcal{E}$ *has significant influence* against H_1 : , i^{th} *variable in* $\mathcal{E}$ *has no significant influence*. We choose those values, for which null hypothesis is accepted. If more than one value of a single variable has chosen to be significant take that value for which the corresponding p-value is maximum. We choose those attributes and the corresponding value and form the set $\mathcal{V}$ and the argument *emphasizing-argument*($\mathcal{V}$). If we found that no attribute has significant influence on the opponent's decision then no argument is chosen.

• **Find-best-persuasive-argument:** Each time the opponent agent rejects an offer, the agent try to find based on the opponent's belief model, if some persuasive arguments can increase the probability of accepting the offer by the opponent agent. This has a corresponding utility. persuasive argument is sent if its expected utility exceeds the expected utility of the other arguments or offers. There may be more than once persuasion the agent want to make in one *persuasive-argument*.

8 Conclusion and Future Work

In this paper, we have presented a novel architecture that allows an agent to negotiate better deals by using argumentation. We also propose the use of Bayesian network for representing the opponent's belief model and provide a framework by which such a model can be used to generate arguments that are likely to convince the opponent to accept proposed offers. The use of Bayes nets allow us to formally capture the complex interrelationships between domain issues and their influence on the opponent's decisions. Such models allow agents to efficiently arrive at profitable negotiated settlements. Such models can also be updated based on negotiation history and can serve as useful repositories for dealing with steady customers.

We have presented an asymmetric negotiation model, with a knowledgeable domain expert interacting with a user agent. We plan to extend this model for peer-to-peer level interaction scenarios. In particular we are interested in applying such techniques in P2P environments, where both the agents may have similar knowledge, for resource procurement and exchange.

Acknowledgements

This work has been supported in part by an NSF award IIS-0209208. We thank the anonymous reviewers for their valuable comments.

References

1. L. Amgoud, N. Maudet, and S. Parsons. An argumentation-based semantics for agent communication languages. In F. Van Harmelen, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI-2002)*, pages 38–42, Lyon, France, July 2002. IOS Press.
2. David Heckerman. A tutorial on learning bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, March 1995.
3. Nicholas Jennings, P. Faratin, Alessio R. Lomuscio Simon Parsons, Carles Sierra, and Michael Wooldridge. Automated negotiation: prospects, methods and challenges. *Int. Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.
4. Nicholas Jennings, Simon Parsons, P. Noriega, and Carles Sierra. On argumentation-based negotiation. In *Proc. of Int. Workshop on Multi-Agent Systems*, 1998.
5. Finn Jensen. *An Introduction to bayesian Networks.* UCL Press and Springer Verlag, 1996.
6. Finn V. Jensen. Bayesian networks and influence diagrams. *Risk Management Strategies in Agriculture; Huirne et al (eds.); Mansholt Studies*, pages 199–213, 1997.
7. Antonis Kakas and Pavlos Moraitis. Argumentation based decision making for autonomous agents. In *Proceedings of AAMAS-03*, pages 883–890, 2003.
8. Sarit Kraus, Katia Sycara, and Amir Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence journal*, 104(1-2):1–69, 1998.
9. Simon Parsons, Carles Sierra, and Nicholas R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.
10. Iyad Rahwan, Sarvapali D. Ramchurn, Nicholas R. Jennings, Peter McBurney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *Knowledge Engineering Review*, 2004.
11. Iyad Rahwan, Liz Sonenberg, and Frank Dignum. Towards interest-based negotiation. In *Proceedings of AAMAS-03*, pages 773–780, 2003.
12. Sarvapali D. Ramchurn, Nicholas R. Jennings, and Carles Sierra. Persuasive negotiation for autonomous agents: A rhetorical approach. In *Proc. IJCAI Workshop on Computational Models of Natural Argument*, pages 9–17, 2003.
13. Tuomas W. Sandholm and Victor R. Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *First International Conference on Multiagent Systems*, pages 328–335, 1995.
14. Carles Sierra, Nicholas R. Jennings, Pablo Noriega, and Simon Parsons. A framework for argumentation-based negotiation. *Intelligent Agents IV LNAI*, 1365:177–192, 1998.
15. D. J. Spiegelhalter and S. L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605, 1990.
16. Ingrid Zukerman, Richard McConachy, and Kevin B. Korb. Bayesian reasoning in an abductive mechanism for argument generation and analysis. In *Proceedings of AAAI-98*, pages 833–838, 1998.