The current state of automated negotiation theory: a literature review[*]

Sam Vente[1][0000−0002−5485−6571], Angelika Kimmig[1][0000−0002−6742−4057], Alun Preece[1][0000−0003−0349−9057], and Federico Cerutti[2] [1][0000−0003−0755−0358]

[1] Cardiff University, Cardiff, CF10 3AT, UK
{VenteDA,KimmigA,PreeceAD,CeruttiF}@cardiff.ac.uk
[2] Department of Information Engineering, University of Brescia, Italy
federico.cerutti@unibs.it

Abstract. Automated negotiation can be an efficient method for resolving conflict and redistributing resources in a coalition setting. Automated negotiation has already seen increased usage in fields such as e-commerce and power distribution in smart girds, and recent advancements in opponent modelling have proven to deliver better outcomes. However, significant barriers to more widespread adoption remain, such as lack of predictable outcome over time and user trust. Additionally, there have been many recent advancements in the field of reasoning about uncertainty, which could help alleviate both those problems. As there is no recent survey on these two fields, and specifically not on their possible intersection we aim to provide such a survey here.

Keywords: Automated negotiation · Multi-agent systems · Constraints

1 Introduction

Negotiation is an important method of conflict resolution and resource allocation in multi-agent systems. In this survey I will focus on *bilateral* negotiations, meaning negotiations with only two parties, but many of the techniques will also be applicable to multilateral negotiations. The current state of the art in automated negotiation places a lot of emphasis on *Proposal Based Negotiation* (PBN) [4] which will be explained in more detail in Section 2.1. This means that the only method of communication the protocol allows is either accepting a previous proposal, offering a counter-proposal or terminating the negotiation altogether.

Work has already been done to develop negotiation support systems (NSS) [20] and verify that they do transfer benefits to the user when used [18,24,38]. Since emotion can have big impacts on how people act both inside and outside negotiations, affective computation, that is computation relating to human emotion, could potentially be worthwhile integrating into NSS. In fact, authors such as Broekens, Jonker and Meyer [7] assert that integrating useful affective computation into NSS is possible in the short term (several years). Despite this work, however, as identified by Baarslag et. al. in [5], and by Pommeranz et. al. in [29] adoption of autonomous negotiation agents and NSS systems still faces significant challenges. Baarslag et. al. [5] identify three main challenges to more widespread adoption:

1. Necessity for domain knowledge and difficulty of preference elicitation
2. Long-term perspective (having non-stationary preferences and performing well over multiple encounters)
3. User trust

They identify that the absence of predictable outcome and transparent preference elicitation form a major barrier to overcome, especially to non specialist users. They also mention that there exists a tension between predictability and performance, since if an agent is predictable to its adversary, it becomes exploitable, meaning that an agent should be predictable but only to the user. They mention that this can be alleviated by, amongst others, quantifying uncertainty. Therefore, it is my hypothesis that it is beneficial to focus on incorporating models of computation that can inspire more user trust by providing the following:

1. Explanations for the agent's beliefs and actions
2. Certain guarantees or quantifying the risk of certain outcomes occurring
3. Assessments of how likely an opponent is to honour the outcome of a negotiation by measuring credible commitment

I argue that one way to achieve better transparent consequences is through integrating a Reasoning About Uncertainty (RAU) approach into negotiations. RAU is a well known and well-studied problem with a mature body of research written about it [31,35,23,10,17]. RAU also often offers models that are in some way more explainable than many other systems. User trust often not only has to do with the outcome of a certain task but also the wider implications of that outcome. Using methods such as probabilistic reasoning will allow an agent to consider the wider implications and present these to the user. It is, therefore, my belief that integrating an RAU approach into automated negotiation is a good way to achieve the goals outlined above.

2   Basic Negotiation theory

In this section I will discuss relevant theory of automated negotiation, summarising work from [4,3,8,19,21,30,33]

2.1 Negotiations

A high-concept illustration of the components of an automated negotiation is depicted in Figure 1. The *negotiation setting* describes the entire negotiation. Its main components are the agents, the *negotiation protocol* and the *negotiation scenario*. We will discuss these elements in more detail below.
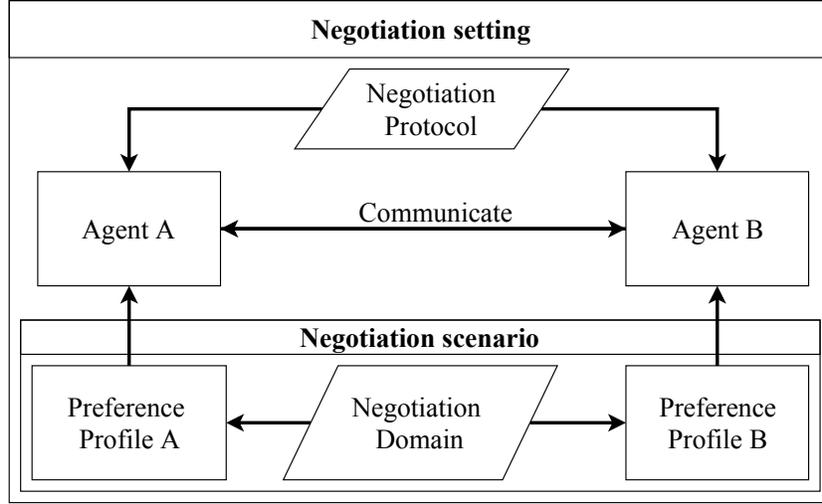

Fig. 1. An illustration of the elements defining an automated bilateral negotiation. Reproduced from [4]

PBN Protocols The negotiation protocol details the methods of communication. In this work, it is assumed that both agents adhere to the same protocol. One of the more well-known protocols is the Alternating-Offers Protocol (AOP). In this protocol, the two agents take turns making offers, or bids to each other until one of the agents accepts [4]. When we refer to *Proposal Based Negotiation* (PBN) we refer to negotiations that utilise AOP. While the AOP is specifically focused on bilateral negotiations, there are also versions for multilateral negotiations. Such an example is the Stacked Alternating Offers Protocol (SAOP) [2] where every agent can make a new proposal or choose to accept the most recent proposal. In general, PBN can also refer to negotiations using a multilateral version AOP (such as SAOP, but more exist), but those are not discussed here.

Of course, different protocols provide different benefits and disadvantages. Therefore it is important and also non-trivial to choose the correct protocol for framing a given problem. In [26] Marsa-Maestre et. al. provide rules-of-thumb and strategies for matching a beneficial method of modelling the negotiation based on the problem details, including which protocol to use.

Negotiation scenarios The negotiation domain consists of the matters being discussed in the negotiation, meaning that the domain comprises the semantic part of the interaction. Formally we say that the *negotiation domain*, *negotiation space* or *outcome space*, which is usually denoted as Ω, is the product of at least one *issue*. An issue, denoted Λ, is a countable set whose elements are called *values*. Each issue represents one matter that needs to be agreed upon, and the values represent possible assignments. A *proposal* is a vector $\omega \in \Omega$, representing a possible assignment to the issues being proposed.

The negotiation domain informs the *preference profiles* of the agents. An ordinal preference profile is a linear preorder $\succeq$, meaning a linear ordering that is not necessarily anti-symmetric, on the outcome space. We say that an agent *weakly prefers* ω to ζ if $\omega \succeq \zeta$. If in addition $\zeta \succeq \omega$ holds, we say that the agent is *indifferent* towards the offers, which is written as $\zeta \sim \omega$. If $\omega \succeq \zeta$ holds but $\zeta \succeq \omega$ does not, then we say that ω is *strictly* or *strongly preferred* to ζ, and write $\omega \succ \zeta$ [30]. Finally, a proposal ρ is called *Pareto optimal* if it cannot be improved upon for one agent without making it worse for another. More formally this means that a proposal ρ is Pareto optimal if, whenever an agent prefers ζ to ρ then the other agent prefers ρ to ζ. This is a measure of the efficiency of an agreed proposal, because if a proposal is not Pareto optimal, there exists an offer that all parties prefer or are indifferent to. The piece-wise linear curve connecting all Pareto optimal solutions is called the *Pareto frontier*. An agent may also specify a *reservation value*, which is the least favourable proposal that is still preferable to ending the negotiation without an agreement [4].

Although an ordinal preference profile can be formulated by listing all of the ordinal relationships, for large negotiation spaces this quickly becomes unwieldy. A more common way of formulating a preference profile is by use of a *utility function*. This is a function $u : \Omega \to \mathbb{R}$, which assigns a numeric value to each outcome which represents how much the agent appreciates that outcome. Every utility function u induces a preference profile via the following relationship

$$\omega \succeq \zeta \iff u(\omega) \geq u(\zeta). \tag{1}$$

Such preference profiles are called *cardinal* preference profiles. Sometimes additional assumptions are made about utility functions to help alleviate computational complexity but this almost always comes at the cost of expressiveness. One common way of doing this is by using *linear additive functions* for which it holds that for any $\omega \in \Omega$ where Ω consists of n issues, the following equation holds

$$u(\omega) = \sum_{i=1}^{n} w_i e_i(\omega_i) \tag{2}$$

[4]. Here w_i are normalised weights representing the relative importance of each issue.

Time In the real world, negotiation is often very costly both because the issues debated might be time sensitive (resources decaying for example) and because

the act of negotiating itself requires resources. Therefore it is often desirable to take into consideration the time needed to negotiate [8]. One common way to model this is by using a *time discounted* utility. For a certain discount factor $\delta \in (0, 1]$ and $t \in [0, 1]$ the normalised time this is defined as

$$u^\delta(\omega) = u(\omega) \cdot \delta^t. \tag{3}$$

Here a small δ represents high time pressure and $\delta = 1$ represents no time-pressure. [4]

2.2 Success metrics

Measuring negotiation outcomes There are a variety of metrics commonly used to assess the outcome of a negotiation:

- *Average utility* This is by far the most common metric used to assess the efficiency of a negotiation agent. The average utility an agent can achieve is indicative of its performance. This metric does have its limitations however, since it can provide only a metric relative to a certain utility function, and cannot provide a more general notion of performance. For example, if one agent uses a utility function u and another uses the same strategy but uses the utility function $w(\omega) = 10 \cdot u(\omega)$ they will make the same decisions but the agent using w will in general appear to perform much better.
- *Fairness of outcome* In some negotiation settings, such as cooperation settings, or setting where repeated interactions are likely, the fairness of an outcome can be an important factor. Often this is characterised as the average distance to solutions that satisfy fairness characteristics such as the Nash solution, or the Kalai-Smorodinsky solution. Here the Nash solution is defined as the bid which maximises the product of the utilities [4]:

$$\omega_{Nash} = \max_{\omega \in \Omega} u_a(\omega) \cdot u_b(\omega) \tag{4}$$

  and the Kalai-Smorodinsky solution is defined as the Pareto optimal bid ω for which the following equation holds:

$$\frac{u_{max_a}}{u_{max_b}} = \frac{u_a(\omega)}{u_b(\omega)} \tag{5}$$

  meaning that each agent gets a utility proportional to their maximal possible utility. The main difference between these measures is that Nash equilibria are stable under removing irrelevant options from the domain, while the Kalai-Smorodinsky solution guarantees that if there are more resources to divide, then agents won't have their utility decreased by claiming them [22].
- *Distance to Pareto frontier* As I mentioned in Section 2.1, a Pareto bid is in some ways maximally efficient because if a bid is not Pareto optimal, by definition there exists a bid that is better for at least one agent and nobody will have to concede. Therefore the average distance to the nearest Pareto bid is

often used as a measure of the efficiency of an outcome. One problem with analysing Pareto optimally is that it requires perfect information. However, preference strategies and thus Pareto optimal bids can be estimated using machine learning, which will be explored in more detail in section 2.4. Calculating the Pareto frontier can be costly, especially since it suffers from the curse of dimensionality. Genetic algorithms are often used to approximate it such as in [21] but exact algorithms have also been proposed [19].

− *Joint utility* Another measure of fairness that is sometimes used is the joint utility since this in some ways can compensate for exploitation. Many authors simply use the sum of the utilities. However, this can have sensitivity issues since it has trouble detecting unfair outcomes if the scale of the utilities greatly differs. Another possibility is to take the geometric mean of the utilities, as shown in equation 6, which suffers less from the sensitivity problem.

$$u_{joint}(\omega) = \sqrt{u_a(\omega) \cdot u_b(\omega)} \tag{6}$$

− *Robustness* Especially when assessing the possible autonomy of an agent and its long term viability, it is often important to consider its robustness. Robustness is a measure of how easily the agent can be exploited by an adversary switching tactics. Highly specialised agents can produce very good outcomes but if they have no means of adapting, they can easily become exploitable by agents who have knowledge of the strategy [3]. Often robustness is measured using some form of Empirical Game-Theoretic analysis (EGT), sometimes also called Evolutionary Game Theoretic analysis. Examples of works using this kind of analysis include [3] and [11]. EGT attempts to provide insight into the robustness of agents by simulating many negotiations among different agents while letting agents change strategies based on the outcomes of the previous round of simulations. Robustness can then be measured by observing the evolution of the distributions of the strategies over time. Work has also been done recently by Tuyls et. al. [36] to prove theoretical guarantees on an EGT analysis technique called meta-game analysis. This technique models higher level strategies using EGT instead of the atomic actions.

− *Time until agreement* As stated earlier, many negotiations have some kind of time pressure. In those scenarios it can be useful to measure the time until an agreement is reached, if at all. Depending on the scenario, this time can be measured in real time, or in the number of messages exchanged, which might be more important in certain low bandwidth scenarios. One disadvantage of this method is that it does not take the quality of the outcome of the negotiation into account.

2.3   Argumentation based negotiation

In [33] Rahwan et. al. mention that models of PBNs tend to have several disadvantages. For example, usually it is assumed that the preference profile is static and completely known prior to the interaction. This presents problems when

agents have to negotiate more complex situations such as trade union negotiations where the value of an outcome is not only unclear, but debatable. Another example is that sometimes agents can fail to reach an agreement because of incorrect assumptions, or the fact that some of the assumptions of perfect rationality cannot always be guaranteed for agents due to matters such as computational resource constraints. In an attempt to address these problems work has been done to create protocols for negotiation that allow more than just proposals to be exchanged. This type of negotiation are referred to as Argumentation Based Negotiations (ABN). This does, however, bring with it some unique challenges. In addition to the challenges any agent should overcome, such as when to agree, when to walk away, and generate bids. Rahwan et. al. [33] identify several other challenges which argumentation based agents must also address:

- *Message interpretation:* In contrast to PBN a message in an ABN may or may not contain, amongst others, a proposal, a rejection, an acceptation, an argumentation for or against a fact, an assertion or anything else that is allowed by the protocol specification used. All these possibilities need to be parsed and differentiated so that they can be dealt with appropriately. When I say message interpretation, that usually refers to the parsing of the low-level symbols into a more semantic representation of the message. For example, imagine that agents A and B are in an ABN. When A receives a message containing the literal string `"assert(isAlly,A,B)"`, then message interpretation refers to the process of converting this string into an internal representation of the fact that B is asserting that A and B are allies (whatever that may mean in context). This is more challenging than in the PBN case since all of the possibilities mentioned above might have their own associated symbols and/or grammar. For example, consider the difference between how the (arbitrary) messages `"assert(isAlly,A,B)"`, and `"propose(yes,yes,no,4,3,no)"` should be parsed. The first is an assertion that can be accepted or rejected and the second is a proposal that has to be considered in a very different way based on its utility. Several options for communication standards are explored in [9].
- *Message evaluation:* In ABNs evaluating messages is no longer only limited to calculating the utility of an offer, but must involve some form of reasoning. For example take the example of `"assert(isAlly,A,B)"`. First of all the agent must decide whether to accept or reject this assertion and then the agent must consider the implications and how it values those.
- *Message generation:* In addition to generating new proposals, an agent must now also be able to generate new arguments and decide when it is appropriate to do so.
- *Belief revision:* Although not technically necessary, a belief state is integral to many forms of argumentation[1,33]. In addition to that, several authors also note that incorporating belief revision into an agent has benefits, such as being able to avoid non-agreement due to incorrect assumptions [9,28]. This opens up a potential way of modelling credible commitment or assessing

how likely it is an agent can and will make good on its agreements. This is an avenue which is not explicitly addressed in the current literature.

ABN Protocols A defining trait of an ABN is that the protocol in some way allows agents to express the reasoning behind a bid or rejection. In [33], Rahwan et. al. mentions that in addition to the language ABN agents use to communicate, they also often need a *domain language* to refer to concepts of the environment which are relevant to the negotiation. One common standard protocol is the FIPA ACL protocol [12,13,14], which is an architecture specification and an associated protocol specification that attempts to promote inter-operability between negotiation agents. FIPA ACL does not specify any domain language, leaving the interpretation of the messages to the agent.

Some work has also been done by Luo et. al. in [25]. They propose a method of negotiation based on the AOP protocol, but also develop mechanics to allow agents to express fuzzy constraints to each other, which they can use to find Pareto optimal solutions.

2.4   Negotiation and machine learning

Forms of opponent modelling Opponent modelling can aid an agent to achieve better outcomes in a number of ways, but the most common ones are being able to avoid exploitation by adapting to the opponent, minimising the cost of negotiations and being able to reach win-win arguments. While the modelling techniques mentioned below can be applied to any negotiation, their viability depends more on details of the encounter such as the availability of relevant data, or the representation of the preference profiles that agents use.

- *Learning the acceptance strategy* Having information about the acceptance strategy of the opponent has a huge strategical advantage since it allows an agent to look for the bid that the agent itself prefers the most which is still acceptable to the opponent. However, this is often kept secret to avoid exploitation. A popular way of learning the acceptation strategy is estimating the *acceptance probability*, which is the probability that the agent will accept any given proposal. Sometimes it can also be beneficial to model the opponent's deadline since that can reveal whether stall-tactics can be effective or not [4].
- *Learning the bidding strategy* Having information about an opponent's bidding strategy and adjusting one's own preference information accordingly can be beneficial. For example, demanding higher concessions from agents that are known to concede easily can lead to better outcomes. However, this is a challenging task, especially since an agent's strategy may not be static over time. Time series forecasting is sometimes used to predict the future behaviour of opponents [4].

Conversely, the aim of using an opponent model is usually to increase performance, therefore it is also very popular to measure the difference in performance

when using the model. For this any of the metrics outlined in Section 2.2 can be used.

3   Reasoning about uncertainty

It is my belief that RAU provides methods of computation that can satisfy the goals I have set out in the introduction of this piece. Therefore I will first give a brief overview of the theory of probabilistic reasoning I intend to employ to achieve those goals.

3.1   DTProbLog

First order logic While a basic understanding of first order logic is assumed, I will give a very brief overview of the used terminology.

- A *constant* is a symbol referring to a specific object such as 2 or `bart`
- A *variable* is a symbol referring to a wider variety of objects such as `X` or `Person` Here I use the convention that logical constants start with a lower case letter and logical variables start with an upper case letter.
- A *functional expression* is a functional constant, together with n terms called its *arguments* enclosed in parenthesis and separated by commas. Here n is called the *arity* of the function. For example if f is a functional constant of arity 2 and `a` and `Y` are terms (to be defined below), then `f(a,Y)` is a functional expression.
- A *term* is a variable, a constant or a functional expression.
- A *literal* is an atomic logical sentence, or its negation, such as p or $\neg q$.
- A *clause* is a disjunction of literals, such as $p \wedge \neg q$.
- A *predicate* is an assertion such as `isFatherOf(homer,bart)`.
- An *atom* is a sentence consisting of a single predicate.

A logical expression is called *ground* if it contains no variables, i.e. `isFatherOf(X,bart)` is not ground while `isFatherOf(homer,bart)` is.

A set of clauses is often referred to as either a *program* or a *knowledge base* (KB). The set of all ground atoms occurring in a knowledge base KB is its *Herbrand base*. A *Herbrand interpretation* is a mapping from a Herbrand base onto the set of truth values $\{true, false\}$. Whenever a Herbrand interpretation maps an atom or clause to *true* this interpretation is called a *model* for that atom or clause. An interpretation is a model for a knowledge base if it is also a model for every clause in that knowledge base [15]. For a knowledge base KB and clause φ we say that KB *logically entails* or *models* φ if every model for KB is also a model for φ, which is written as $KB \models \varphi$. Furthermore, if R is a set of inference rules, and φ can be deduced using the rules in R from a set of clauses called Δ, then we say that φ is *provable* in Δ, and we write $\Delta \vdash_R \varphi$ or often $\Delta \vdash \varphi$ if R is understood. If $\Delta \models \varphi$ holds whenever $\Delta \vdash_R \varphi$ holds R is called *sound*, which is to say that everything that is provable from a set is also logically entailed by it. If the reverse always holds, R is called *complete*.

A *substitution* θ is a mapping from variables to terms. For example, let $\theta = \{X \rightarrow \text{homer}, Y \rightarrow \text{bart}\}$ and $\varphi = \text{isFatherOf(X,Y)}$ then we can apply θ to φ which is denoted as $\varphi\theta$ which results in `isFatherOf(homer,bart)`. Note that in this example $\varphi\theta$ is ground. When this is the case, we call θ a *grounding* of φ [16].

Prolog As mentioned in [34] Prolog is one of the most widely used logic programming languages. A Prolog program is a set of *definite* or *Horn clauses*, meaning sentences or clauses in first-order logic containing at most one positive literal, such as $u \vee \neg p \vee \neg q$, which can be rewritten as $(p \wedge q) \rightarrow u$. Here u is

called the *head* of the clause and $p \wedge q$ is called the *body*. In Prolog this is written as

$$\texttt{u} : -\texttt{p}, \texttt{q}.$$

This can be read as "u **if** p **and** q". As mentioned in [15] Prolog uses what is called Selective Linear Definite clause resolution or SLD resolution, which is both sound and complete for Horn clauses. However, it should be noted that Prolog uses a depth-first search manner to search for proofs, and may therefore not terminate for certain programs.

In the context of Prolog, a *fact* is a clause with an empty body. A *rule* is a clause with a non-empty body.

ProbLog ProbLog [32] is a probabilistic extension of Prolog. We will assume basic knowledge of Prolog in this work. For a more detailed instruction see for example [15]. Where in pure Prolog one is commonly interested in whether a given query fails or succeeds, in ProbLog one is focused on computing the probability that a query succeeds in a given ProbLog program. A ProbLog program $\mathcal{T} = \mathcal{F} \cup \mathcal{BK}$ consists of a set of labelled probabilistic facts $\mathcal{F}$, and a set of definite clauses $\mathcal{BK}$ called the background knowledge. Here, $\mathcal{F}$ is of the form $\mathcal{F} = \{p_1 :: f_1, \ldots, p_n :: f_n\}$ where $f_i\theta$ is true with probability p_i for all substitutions θ grounding f_i. The probabilities of all the facts are assumed to be independent. Furthermore, let $\mathcal{F}\Theta$ denote the set of all possible ground instances of the facts in $\mathcal{F}$ and let $\mathcal{F}_L \subset \mathcal{F}\Theta$. Then A ProbLog program defines a probability distribution of all the sub Prolog programs $L = \mathcal{F}_L \cup \mathcal{BK}$ via the following identity:

$$\mathbb{P}(L|\mathcal{T}) = \prod_{f_i \in \mathcal{F}_L} p_i \prod_{f_i \in \mathcal{F}\Theta \setminus \mathcal{F}_L} (1 - p_i) \tag{7}$$

From that we can determine the *success probability* of a given query q using the following identities:

$$\mathbb{P}(q|L) = \begin{cases} 1 & \exists \theta : L \models q\theta \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$\mathbb{P}(q, L|T) = \mathbb{P}(q|L) \cdot \mathbb{P}(L|T) \tag{9}$$

$$\mathbb{P}(q|T) = \sum_{M \subseteq L_T} \mathbb{P}(q, M|T) \tag{10}$$

So the probability of a query given a program is the probability of the query having a proof in a randomly sampled subprogram according to the distribution.

DTProbLog DTProbLog [6] is an extension of ProbLog that allows for strategic planning, given some quantified uncertainty by maximising the expected utility of some actions.

A DTProbLog program is of the form $\mathcal{T} = L_T \cup \mathcal{D} \cup \mathcal{U}$, where L_T is a ProbLog program, $\mathcal{D}$ is the set of *decision facts* and $\mathcal{U}$ is the set of *utility attributes*. Decision facts are of the form ? :: d where d is an atom and ? indicates that it is a decision variable for which the program should optimise. A *strategy* σ is a function $\sigma : \mathcal{D} \to [0,1]$ denoting the probabilities with which the agent will assign that literal as true. For a set of decision facts $\mathcal{D}$ and a strategy σ the following set is also defined: $\sigma(\mathcal{D}) = \{\sigma(d) :: d | (? :: d) \in \mathcal{D}\}$. Note that $L_T \cup \sigma(\mathcal{D})$ is a standard ProbLog program and we will denoted it with $\sigma(\mathcal{DT})$.

Furthermore a utility attribute is a statement of the form $u_i \to r_i$ where u_i is a literal and r_i is a numerical reward that will be awarded whenever u_i is satisfied. We then define the utility of a subprogram L as

$$\text{Util}(L) = \sum_{(u_i \to r_i) \in \mathcal{U}} r_i \cdot \mathbb{P}(u_i | L) \tag{11}$$

where $\mathbb{P}(u_i | L)$ can be computed according to equation 8. Since a ProbLog program $\mathcal{T}$ defines a probability distribution over the subprograms we can define the expected utility of a program as

$$\text{Util}(\mathcal{T}) = \sum_{(u_i \to r_i) \in \mathcal{U}} r_i \cdot \mathbb{P}(u_i | \mathcal{T}) \tag{12}$$

This leads us to the definition of the utility of a single utility attribute a_i under a given strategy:

$$\text{Util}(a_i | \sigma, \mathcal{DT}) = r_i \cdot \mathbb{P}(u_i | \sigma(\mathcal{DT})) \tag{13}$$

and the total utility of a strategy

$$\text{Util}(\sigma, \mathcal{DT}) = \text{Util}(\sigma(\mathcal{DT})) = \sum_{a_i \in \mathcal{U}} \text{Util}(a_i | \sigma(\mathcal{DT})) \tag{14}$$

So, in other words, the utility of a strategy is defined as the sum of the expected utilities of all the utility attributes.

4 Future Work

4.1 An RAU implementation for better predictability of outcome and broader awareness

As detailed in the introduction Baarslag et. al. identify in [5] that a lack of predictable outcome to the user poses a major challenge to overcome to ensure wider adoption of automated negotiations and NSS. Using probabilistic reasoning to implement negotiation agents provides a clear and inbuilt way of providing that predictability, and also provides a way to incorporate a broader view of the situation into the workings of an agent instead of only the issues pertaining to the actual negotiation. It has already been seen in [27] that context awareness

has advantages for negotiation agents. Applications using probabilistic reasoning methods such as ProbLog [32] could incorporate information from various sensors, such as demonstrated by [39], to be incorporated in both bidding strategies and opponent modelling. This could provide potentially huge advantages to agents that are able to apply these concepts well.

4.2   Constrained proposal methods

Some work has already been done in [25] by Luo et. al. to incorporate fuzzy constraints into a buyer-seller negotiation setting and found that it ensures that agreements are Pareto optimal. In addition it also minimises the amount of information agents have to disclose, and improves the efficiency of the negotiations over standard AOP settings. It has already been shown by [37] that incorporating very simple constraints into an AOP setting can significantly speed up the negotiations without impacting the outcome. The authors did not test scenarios with time discounted utility functions, more complex or perhaps "approximate" constraints. If agents were to incorporate more sophisticated methods of discovering constraints, reasoning about them and incorporating possible constraints into bidding strategies, this would convey many opportunities to improve both in terms of explainability and performance.

References

1. Amgoud, L., Vesic, S.: A formal analysis of the role of argumentation in negotiation dialogues. Journal of Logic and Computation **22**(5), 957–978 (2012). https://doi.org/10.1093/logcom/exr037
2. Aydoğan, R., Festen, D., Hindriks, K.V., Jonker, C.M.: Alternating Offers Protocols for Multilateral Negotiation. In: Modern Approaches to Agent-based Complex Automated Negotiation, pp. 153–167. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-51563-2_10
3. Baarslag, T., Fujita, K., Gerding, E.H., Hindriks, K., Ito, T., Jennings, N.R., Jonker, C., Kraus, S., Lin, R., Robu, V., Williams, C.R.: Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. Artificial Intelligence **198**, 73–103 (2013). https://doi.org/10.1016/j.artint.2012.09.004, `http://dx.doi.org/10.1016/j.artint.2012.09.004`, publisher: Elsevier B.V.
4. Baarslag, T., Hendrikx, M.J., Hindriks, K.V., Jonker, C.M.: Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. Autonomous Agents and Multi-Agent Systems **30**(5), 849–898 (2016). https://doi.org/10.1007/s10458-015-9309-1, publisher: Springer US ISBN: 1045801593091
5. Baarslag, T., Kaisers, M., Gerding, E.H., Jonker, C.M., Gratch, J.: When Will Negotiation Agents Be Able to Represent Us? The Challenges and Opportunities for Autonomous Negotiators. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. pp. 4684–4690. International Joint Conferences on Artificial Intelligence Organization, California (Aug 2017). https://doi.org/10.24963/ijcai.2017/653, `https://www.ijcai.org/proceedings/2017/653`

6. Van den Broeck, G., Thon, I., van Otterlo, M., De Raedt, L.: DTProbLog: A Decision-Theoretic Probabilistic Prolog. In: Proceedings of the twenty-fourth AAAI conference on artificial intelligence. vol. 2, pp. 1217–1222 (2010)

7. Broekens, J., Jonker, C.M., Meyer, J.J.C.: Affective negotiation support systems. Journal of Ambient Intelligence and Smart Environments **2**(2), 121–144 (2010). https://doi.org/10.3233/AIS-2010-0065

8. Carnevale, P.J., O'Connor, K.M., McCusker, C.: Time Pressure in Negotiation and Mediation. In: Time Pressure and Stress in Human Judgment and Decision Making, pp. 117–127. Springer US, Boston, MA (1993)

9. Casali, A., Pilotti, P., Chesñevar, C.: Assessing communication strategies in argumentation-based negotiation agents equipped with belief revision. Argument and Computation **7**(2-3), 175–200 (2016). https://doi.org/10.3233/AAC-160012

10. Cerutti, F., Kaplan, L., Kimmig, A., Sensoy, M.: Probabilistic logic programming with beta-distributed random variables (2018)

11. Chen, S., Hao, J., Weiss, G., Tuyls, K., Leung, H.f.: Evaluating Practical Automated Negotiation Based on Spatial Evolutionary Game Theory. In: Lutz, C., Thielscher, M. (eds.) Evaluating Practical Automated Negotiation Based on Spatial Evolutionary Game Theory, pp. 147–158. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-11206-0_15, `http://link.springer.com/10.1007/978-3-319-11206-0_15`

12. El-Sisi, A.B., Mousa, H.M.: Argumentation based negotiation in multiagent system. Proceedings - ICCES 2012: 2012 International Conference on Computer Engineering and Systems pp. 261–266 (2012). https://doi.org/10.1109/ICCES.2012.6408525, iSBN: 9781467329613

13. FIPA: FIPA Abstract Architecture Specification (2002), `http://www.fipa.org/specs/fipa00001/`

14. FIPA: FIPA ACL Message Structure Specification (2002), `http://www.fipa.org/specs/fipa00061/index.html`

15. Flach, P.: Simply Logical: Intelligent Reasoning by Example, vol. 14. John Wiley & Sons, Inc., New York, NY, USA (Feb 1994). https://doi.org/10.1016/0169-023X(95)90019-D, publication Title: Data & Knowledge Engineering ISSN: 0169023X

16. Genesereth, M., Kao, E.: Introduction to Logic, Second Edition. Synthesis Lectures on Computer Science **4**(2), 1–163 (Aug 2013). https://doi.org/10.2200/S00518ED2V01Y201306CSL006, iSBN: 978-0-205-82037-5

17. Halpern, J.Y.: Reasoning about uncertainty. The MIT Press, Cambridge, Massachusetts, second edition. edn. (2017)

18. Han, X., Feng, Y.Q., Xue Han, Yu-Qiang Feng: Research on negotiation support system based on case-based reasoning. In: Gastroenterology. vol. 23, pp. 2850 – 2854 Vol. 5. IEEE (2005). https://doi.org/10.1109/ICMLC.2005.1527428, `http://ieeexplore.ieee.org/document/1527428/`

19. Hu, X.B., Wang, M., Di Paolo, E.: Calculating complete and exact pareto front for multiobjective optimization: A new deterministic approach for discrete problems. IEEE Transactions on Cybernetics **43**(3), 1088–1101 (2013). https://doi.org/10.1109/TSMCB.2012.2223756, publisher: IEEE

20. Jonker, C.M., Aydoğan, R., Baarslag, T., Broekens, J., Detweiler, C.A., Hindriks, K.V., Huldtgren, A., Pasman, W.: An Introduction to the Pocket Negotiator: A General Purpose Negotiation Support System. In: Multi-Agent Systems and Agreement Technologies. vol. 10207, pp. 13–27 (2017), iSSN: 16113349

21. Kakimoto, S., Fujita, K.: Estimating pareto fronts using issue dependency for bilateral multi-issue closed nonlinear negotiations. Proceedings - IEEE 7th International Conference on Service-Oriented Computing and Applications, SOCA 2014 pp. 289–293 (2014). https://doi.org/10.1109/SOCA.2014.30, publisher: IEEE ISBN: 9781479968336

22. Kalai, E., Smorodinsky, M.: Other Solutions to Nash's Bargaining Problem. Econometrica **43**(3), 513–518 (1975). https://doi.org/10.2307/1914280, `http://www.jstor.org/stable/1914280`, publisher: [Wiley, Econometric Society]

23. Kaplan, L., Cerutti, F., Sensoy, M., Preece, A., Sullivan, P.: Uncertainty aware ai ml: Why and how (2018)

24. Lin, R., Oshrat, Y., Kraus, S.: Investigating the Benefits of Automated Negotiations in Enhancing People's Negotiation Skills. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1. pp. 345–352 (2009). https://doi.org/10.1145/765891.766049, `http://portal.acm.org/citation.cfm?doid=765891.766049`, iSSN: 15582914

25. Luo, X., Jennings, N.R., Shadbolt, N., Leung, H.f., Lee, J.H.m.: A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. Artificial Intelligence **148**, 53–102 (Aug 2003). https://doi.org/10.1016/S0004-3702(03)00041-9, `https://linkinghub.elsevier.com/retrieve/pii/S0004370203000419`, iSBN: 0004-3702

26. Marsa-Maestre, I., Klein, M., Jonker, C.M., Aydoğan, R.: From problems to protocols: Towards a negotiation handbook. Decision Support Systems **60**(1), 39–54 (Apr 2014). https://doi.org/10.1016/j.dss.2013.05.019, `https://linkinghub.elsevier.com/retrieve/pii/S016792361300167X`

27. Mirzayi, S., Taghiyareh, F., Hussein Kazemi, S.M.: IQSon: A Context-aware Negotiator Agent with Enhanced Utility and Decision Making Speed. In: 2018 9th International Symposium on Telecommunications (IST). pp. 603–608. IEEE (Dec 2018). https://doi.org/10.1109/ISTEL.2018.8661115, `https://ieeexplore.ieee.org/document/8661115/`

28. Pilotti, P., Casali, A., Chesñevar, C.: A belief revision approach for argumentation-based negotiation agents. International Journal of Applied Mathematics and Computer Science **25**(3), 455–470 (2015). https://doi.org/10.1515/amcs-2015-0034

29. Pommeranz, A., Wiggers, P., Brinkman, W.P., Jonker, C.M.: Social acceptance of negotiation support systems: Scenario-based exploration with focus groups and online survey. Cognition, Technology and Work **14**(4), 299–317 (2012). https://doi.org/10.1007/s10111-011-0181-8, iSBN: 1435-5558

30. Poole, D.L., Mackworth, A.K.: Artificial Intelligence: Foundations of Computational Agents. Cambridge University Press, New York, NY, USA, 2nd edn. (2010)

31. Preece, A., Braines, D., Cerutti, F., Pham, T.: Explainable ai for intelligence augmentation in multi-domain operations (2019)

32. Raedt, L.D., Kimmig, A., Toivonen, H.: ProbLog: A Probabilistic Prolog and its Application in Link Discovery. Proceedings of the 20th International Joint Conference on Artificial Intelligence pp. 2462 – 2467 (2007), publisher: Morgan Kaufmann Publishers Inc. Place: San Francisco, CA, USA

33. Rahwan, I., Ramchurn, S.D., Jennings, N.R., Mcburney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. The Knowledge Engineering Review **18**(4), 343–375 (Dec 2003). https://doi.org/10.1017/S0269888904000098, publisher: Cambridge University Press ISBN: 0269-8889

34. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edn. (2009)

35. Tomsett, R., Kaplan, L., Cerutti, F., Sullivan, P., Vente, D., Vilamala, M.R., Kimmig, A., Preece, A., Şensoy, M.: Uncertainty-aware situational understanding. In: Pham, T. (ed.) Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications. vol. 11006, pp. 196 – 210. International Society for Optics and Photonics, SPIE (2019). https://doi.org/10.1117/12.2519945, `https://doi.org/10.1117/12.2519945`

36. Tuyls, K., Perolat, J., Lanctot, M., Leibo, J.Z., Graepel, T.: A Generalised Method for Empirical Game Theoretic Analysis. CoRR **abs/1803.0**(July) (Mar 2018), `http://arxiv.org/abs/1803.06376`, arXiv: 1803.06376

37. Vente, S., Kimmig, A., Cerutti, F.: Increasing negotiation performance at the edge of the network. p. 15 (2020)

38. Wu, X., Yin, S., Xiang, X., He, L.: Research on negotiation support systems based on hybrid genetic algorithms. In: 2009 WRI World Congress on Software Engineering, WCSE 2009. vol. 2, pp. 429–433. IEEE (2009). https://doi.org/10.1109/WCSE.2009.264

39. Xing, T., Vilamala, M., Garcia, L., Cerutti, F., Kaplan, L., Preece, A., Srivastava, M.: Deepcep: Deep complex event processing using distributed multimodal information. pp. 87–92 (06 2019). https://doi.org/10.1109/SMARTCOMP.2019.00034