ACAN2011

The 4th International Workshop on Agent-based Complex Automated Negotiations

Taipei, Taiwan, May 3, 2011

ACAN2011:

The 4th International Workshop on Agent-based Complex Automated Negotiations

Preface

Complex Automated Negotiations have been widely studied and are becoming an important, emerging area in the field of Autonomous Agents and Multi-Agent Systems. In general, automated negotiations can be complex, since there are a lot of factors that characterize such negotiations. These factors include the number of issues, dependency between issues, representation of utility, negotiation protocol, negotiation form (bilateral or multi-party), time constraints, etc. Software agents can support automation or simulation of such complex negotiations on the behalf of their owners, and can provide them with adequate bargaining strategies. In many multi-issue bargaining settings, negotiation becomes more than a zero-sum game, so bargaining agents have an incentive to cooperate in order to achieve efficient win-win agreements. Also, in a complex negotiation, there could be multiple issues that are interdependent. Thus, agent's utility will become more complex than simple utility functions. Further, negotiation forms and protocols could be different between bilateral situations and multi-party situations. To realize such a complex automated negotiation, we have to incorporate advanced Artificial Intelligence technologies includes search, CSP, graphical utility models, Bays nets, auctions, utility graphs, predicting and learning methods. Applications could include e-commerce tools, decision-making support tools, negotiation support tools, collaboration tools, etc. We solicit papers on all aspects of such complex automated negotiations in the field of Autonomous Agents and Multi-Agent Systems, including but not limited to:

- Complex Negotiations
- Multi-Issue Negotiations
- Concurrent Negotiations
- Multiple Negotiations

- Sequential Negotiations
- Bilateral Negotiations
- Multilateral negotiation
- Negotiation and Coordination Mechanisms
- Negotiation under Asymmetric Information
- Large Scale Negotiation
- Matchmaking and Brokering Mechanisms
- Coordination for Local and Global Consistency
- 2-sided Matching
- Predicting Opponent's Behaviors in Negotiation.
- Utility models and Preference models
- Complexity aspects of Multi-issue negotiation
- Negotiation Simulation
- Negotiations in Social Networks
- Preference Elicitation
- Practices and Applications

These issues are being explored by researchers from different communities in Autonomous Agents and Multi-Agent systems. They are, for instance, being studied in agent negotiation, multi-issue negotiations, auctions, mechanism design, electronic commerce, voting, secure protocols, matchmaking & brokering, argumentation, and co-operation mechanisms. The goal of this workshop is to bring together researchers from these communities to learn about each other's approaches, form long-term collaborations, and cross-fertilize the different areas to accelerate progress towards scaling up to larger and more realistic applications.

Out of the 16 paper submissions, 9 papers were finally selected as full papers and 6 papers were selected as short papers. Each paper was carefully reviewed by three reviewers, who are considered as experts in the topic.

From 2010, ACAN is tightly cooperating with ANAC (Automated Negotiating Agents Competition). Based on the great success of ANAC2010, the ANAC2011 will be held at AAMAS2011 at Taiwan. This year, we, ACAN, have the ANAC special session, in which the finalists of ANAC will describe their negotiating agents.

Organization

Organizing Chairs:

Takayuki Ito (Nagoya Institute of Technology, Japan / Univ. of Tokyo, Japan)
Minjie Zhang (University of Wollongong, Australia)
Valentin Robu (University of Southampton, UK)
Shaheen Fatima (Loughborough University, UK)
Tokuro Matsuo (Yamagata University, Japan / UCI, US)
Naoki Fukuta (Shizuoka University, Japan)
Hirofumi Yamaki (Nagoya University, Japan)
Miguel A. Lopez-Carmona (Universidad de Alcalá, Spain / MIT, US)
Quan Bai (CSIRO, Australia)
Ivan Marsa-Maestre (Universidad de Alcala, Spain / MIT, US)

***Program Committee*:**

Scott Buffett (National Research Council Canada)
Brahim Chaib-draa (Laval University, Canada)
Joaquin Delgado (Yahoo, inc., United States)
Enrico Gerding (Southampton, UK)
Hiromitsu Hattori (Kyoto University, Japan)
Mark Klein (Massachusetts Institute of Technology (MIT), United States)
Sarit Kraus (Bar Ilan University, Israel)
Katia Sycara (Carnegie Mellon University (CMU), United States)
Paul Scerri (Carnegie Mellon University (CMU), United States)
Guoming Lai (Carnegie Mellon University (CMU), United State)
Dr Xudong Luo (Nanyang Technological University, Singapore)
Miniar Hemaissia (THALES Research & Technology France, France)
Catholijn Jonker (Technical University of Delft, The Netherlands)
Koen Hindriks (Technical University of Delft, The Netherlands)
Zhiqi Shen (Nanyang Technoligical University)
Lotzi Boloni (University Florida, United States)
Kate Larson (University of Waterloo, Canada)

Index

Automated Negotiation Applications

Consensus Policy Based Multi-Agent Negotiation

Miguel A. Lopez-Carmona[*] , Ivan Marsa-Maestre[†] and Mark Klein
Center for Collective Intelligence, MIT Sloan School of Management,
Massachusetts Institute of Technology
NE25-754 Cambridge Center 02142
miguelangel.lopez@uah.es
ivan.marsa@uah.es
m_klein@mit.edu

ABSTRACT

Multiagent negotiation may be understood as a consensus based group decision-making which ideally should seek the agreement of all the participants. However, there exist situations where an unanimous agreement is not possible or simply the rules imposed by the system do not seek such unanimous agreement. In this paper we propose to use a consensus policy based mediation framework (CPMF) to perform multiagent negotiations. This proposal fills a gap in the literature where protocols are in most cases indirectly biased to search for a quorum. The mechanisms proposed to perform the exploration of the negotiation space are derived from the Generalized Pattern Search non-linear optimization technique (GPS). The mediation mechanisms are guided by the aggregation of the agent preferences on the set of alternatives the mediator proposes in each negotiation round. Considerable interest if focused on the implementation of the mediation rules where we allow for a linguistic description of the type of agreements needed. We show empirically that CPMF efficiently manages negotiations following predefined consensus policies and solves situations where unanimous agreements are not viable.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*heuristic methods*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Algorithms, Designs, Experimentation

Keywords

multiagent negotiation, multiparty negotiation, consensus policy, pattern search

[*]Visiting from Universidad de Alcala, Spain
[†]Visiting from Universidad de Alcala, Spain

1. INTRODUCTION

Most research in multiparty automated negotiation has been focused on building efficient mechanisms and protocols to reach agreements among multiple participants, being an objective to optimize some type of social welfare measurement [6, 5, 7, 2, 15, 4, 8, 12, 13]. Examples of such measurements would be the *sum or product of utilities*, the *min* utility, etc... However, social welfare has not been usually placed itself as an integral part of the negotiation process.

There are remarkable works which incorporate a social welfare criterion within the search process [1, 3, 10]. In these works, the authors build mechanisms to obtain fair agreements by using fair direction improvements in the joint exploration of the negotiation space. Put simply, first a mediator proposes a solution and agents provide their utility gradients in the solution, and finally the mediator proposes a new contract in the bisector or in an arbitrary direction which is considered fair enough. These proposals present however several limitations. Firstly, they work only when utility functions are derivable and quasi-concave. Secondly, the absolute value of the gradient is not considered, and so, the marginal utility obtained by the agents in each negotiation round may not be fair. Finally, even considering that the agents reveal also the gradient magnitude, the protocol is prone to untruthful revelation to bias the direction generated by the mediator.

We argue that the type of consensus by which an agreement meets in some specific manner the concerns of all the negotiators should be considered as an integral part within the multiparty negotiation protocols. To study this hypothesis this paper proposes CPMF, a *Consensus Policy Based Mediation Framework for Multi-Agent Negotiation.* CPMF relies on a novel distributed agreement exploration protocol based on the *Generalized Pattern Search* optimization technique (GPS) [9], and on the use of *Ordered Weighted Averaging* (OWA) operators [17]. This framework allows to search for agreements following predefined consensus policies, which may take the form of linguistic expressions in order to satisfy system requirements or to circumvent situations where unanimous agreements are not possible.

Next section presents first the GPS algorithm for unconstrained optimization and then the basic operation of the negotiation protocol. Section 3 focuses on the mechanisms used by the mediator to aggregate agents' preferences and Section 4 presents the agreement search process. The last section summarizes our conclusions and sheds lights on some future research.

2. THE MEDIATION PROTOCOL

We shall assume a set of n agents $A = \{A_i | i = 1, \ldots, n\}$ and a finite set of issues $X = \{x_l | l = 1, \ldots, s\}$, where each issue x_l can be normalized to a continuous or discrete range $d_l = [x_l^{min}, x_l^{max}]$. Accordingly, a *contract* is a vector $x' = \{x'_l = 1, \ldots, s\}$ defined by the issues values. Furthermore, we assume that each agent A_i has a real or virtual mapping $V_i : X \to \mathbb{R}$ function that associates with each contract x a value $V_i(x)$ that gives the payoff the agent assigns to a contract. The exact nature of this mapping need not be known. All that we want to assume is that each agent has some means for formulating a preference function over a set of alternatives. Thus, the preference function can be described as any mapping function between the negotiation space contracts and the set of real numbers. We make a general assumption that the preference of each agent can be non-monotonic and non-differentiable. We only require the preferences to be rational:

DEFINITION 2.1. *The ordinal preference $\precsim_i$ of agent A_i in the negotiation domain is rational if it satisfies the following conditions:*
1. Strict preference is asymmetric: There is no pair of x and x' in X such that $x \prec_i x'$ and $x' \prec_i x$;
2. Transitivity: For all x, x', and x'' in X, if $x \precsim_i x'$ and $x' \precsim_i x''$, then $x \precsim_i x''$;
3. Completeness: For all x and x' in X, either $x \precsim_i x'$ or $x' \precsim_i x$;
where $x \precsim_i x'$ (or $x \prec x'$) indicates that the offer x' is at least as good as (or better than) x for agent i.

The aim of the agents will be to reach an agreement on a contract x', maximizing their individual payoff and minimizing the revelation of private information.

Next, we describe in detail the GPS for unconstrained optimization, which is used in the construction of the negotiation protocol. GPS belongs to the familiy of *Direct Search Based* optimization algorithms [9]. Note, however, that our negotiation protocol is not a single-objetive or multi-objective centralized optimization process.

2.1 Generalized Pattern Search Algorithm for Unconstrained Optimization

The optimization problem can be defined as max $f(x)$, where $f : \mathbb{R}^m \to \mathbb{R}$, $x \in \mathbb{R}^m$. At an iteration k of the protocol, we have an iterate $x(k) \in \mathbb{R}^m$ and a step-length parameter $\triangle_k > 0$. We successively look at the points in the *mesh* $x^+(k) = x(k) \pm \triangle_k e_j$, $j \in \{1, \ldots, m\}$, where e_j is the jth standard basis vector, to search for a contract $x'(k)$ in $x^+(k)$ for which $f(x'(k)) > f(x(k))$. We will use the notation $x^{+o}(k)$ to designate the mesh at round k including the current point $x(k)$. Figure 1 illustrates the set of points

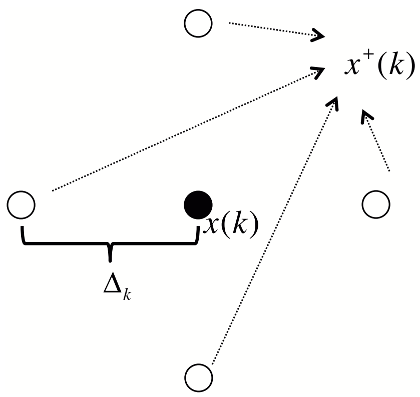


Figure 1: An illustration of Generalized Pattern Search for unconstrained optimization.

among which we search for $m = 2$. This set of points or mesh is an instance of what we call a *pattern*, from which pattern search takes its name. If we find no $x'(k)$ such that $f(x'(k)) > f(x(k))$, then we reduce $\triangle_k$ by half and continue; otherwise, we leave the step-length parameter alone, setting $\triangle_{k+1} = \triangle_k$ and $x(k+1) = x(k)$. In the latter case we can also increase the step-length parameter, say, by a factor of 2, if we feel a longer step might be justified. We repeat the iteration just described until $\triangle_k$ is deemed sufficiently small. One important feature of pattern search that plays a significant role in a global convergence analysis is that we do not need to have an estimate of the derivative of f at $x(k)$ so long as included in the search is a sufficient set of directions to form a positive spanning set for the cone of feasible directions, which in the unconstrained case is all of $\mathbb{R}^m$. In the unconstrained case the set $\{\pm e_j | j = 1, \ldots, m\}$ satisfies this condition, the purpose of which is to ensure that if the current iterate is not a stationary point of the problem, then we have at least one ascendent direction.

The set e_j is defined by the number of independent variables in the objective function m and the positive standard basis set. Two commonly used positive basis sets in pattern search algorithms are the maximal basis, with $2m$ vectors, and the minimal basis, with $m + 1$ vectors. For example, if there are two independent variables in the optimization problem, the default for a $2m$ positive basis consists of the following pattern vectors: $e_1 = \{1, 0\}$, $e_2 = \{0, 1\}$ and $-e_1 = \{-1, 0\}$, $-e_2 = \{0, -1\}$. An $m + 1$ positive basis consists of the following standard basis set: $e_1 = \{1, 0\}$, $e_2 = \{0, 1\}$ and only a negative vector $-e_1 = \{-1, -1\}$. In our approach we will take the $2m$ positive basis. We will use the notation $x^{e_j}(k)|j = 1, \ldots, 2m$ to describe each point in a mesh, and $x(k)$ or $x^{e_0}(k)$ to designate the current point. For example, $x^{e_1}(k)$ specifies the contract generated by the current contract $x(k)$ and the vector e_1 for the current step-length $\triangle_k$, while $x^{e_{m+1}}(k)$ points to the negative version of $x^{e_1}(k)$.

2.2 Basic Operation of the Negotiation Protocol

The basic protocol of the proposed negotiation process is the following:

1. The mediator proposes a mesh from an initial contract $x^{ini}(1)$ for a step-length parameter $\triangle_1$. The point $x^{ini}(1)$ is randomly chosen by the mediator.

2. Each agent provides the mediator their preferences for the contracts in the current mesh x^{+o}, in terms of a mapping $S_i : X \rightarrow [0,1]$ such that for example $S_i(x^{e_j}(k))$ indicates agent i's support for the alternative $x^{e_j}(k)$. An agent does not know the other agents' support for the contracts. Though agents are free to provide support values which are coincident or not with the corresponding private valuation function $V_i(x^{e_j}(k))$, in this work we will assume a perfect correspondence between both values.

3. The individual agent preferences for each contract are aggregated by the mediator to obtain the corresponding group preferences for each of the contracts in the mesh. We shall refer to this as the **aggregation of preferences** step.

4. Mediator decides which is the **preferred contract** in the mesh according to the group preferences for the different contracts.

5. Based on the **the preferred contract**, mediator decides to **expand** or **contract** the mesh. Should a contraction make $\triangle_k$ small enough negotiation ends, otherwise go to step 2.

We assume that the negotiation process is such that a solution from X is always obtained. Negotiation may end when $\triangle_k$ is below a predefined threshold value or when a deadline expires. Essentially, the multi-agent negotiation is a dynamic process where at each stage of the process an agent provides a support measure determined by its underlying payoff function and any information available about the previous stages of the negotiation. The process of choosing the specific support for the different alternatives in a mesh at each round of the negotiations then constitutes a participating agent's strategy. An important consideration in an agent's determination of their strategy are the rules and procedures used in the negotiation process. In the following we shall describe the implementation of the negotiation process steps outlined above.

3. THE AGGREGATION OF PREFERENCES

Here we look at the process where the mediator aggregates the individual support for the contracts in the mesh at round k. Our point of departure here is a collection of n agents and a set $x^{+o}(k)$ of contracts (mesh) given a current contract $x(k)$ at round k. We assume each agent has provided at round k her preference $S_i(x^{+o}(k))$ over the set $x^{+o}(k)$ such that it indicates the degree to which each agent A_i supports each contract. The mediator objective in this mediation step is to obtain a group preference function $G : x^{+o} \rightarrow [0,1]$ which associates with each alternative $x^{e_j}(k) \in x^{+o}(k)$ a value $G(x^{e_j}(k)) = M(S_1(x^{e_j}(k)), \ldots, S_n(x^{e_j}(k)))$.

The form of M is called the *mediation rule*, which describes the process of combining the individual preferences. The form of M can be used to reflect a desired mediation imperative or *consensus policy* for aggregating the preferences of the individual agents to get the mesh group preferences. M will guide the mediator in the expansion-contraction decisions in order to meet the desired type of agreements for the negotiation process.

The most widespread consensus policy found in the automated negotiation literature suggests using as an aggregation imperative a desire to satisfy *all* the agents. However, the policy of requiring that all the agents be satisfied by a solution may not be suitable for multi-agent preference aggregation, or simply the system may need to implement more sophisticated forms of aggregation.

We propose to use other mediation rules to improve the negotiation processes where either a quorum is not necessary or simply such quorum is not possible. For example, a solution may be acceptable if *most* of the agents support it. To incorporate these notions into our negotiation framework we will use a more general class of aggregation rules. The idea is to use a *quantifier guided aggregation*, which allows a natural language expression of the quantity of agents that need to agree on an acceptable solution. As we shall see the *Ordered Weighted Averaging* (OWA) operator [16] will provide a tool to model this kind of softer mediation rule.

3.1 OWA Operators

An aggregation operator $M : S^n \rightarrow G, (S, G \in [0,1])$ is called an OWA operator of dimension n if it has an associated weighting vector $W = [w_1 w_2 \ldots w_n]$ such that $w_t \in [0,1]$ and $\sum_{t=1}^{n} w_t = 1$ and where $M(S_1, \ldots, S_n) = \sum_{t=1}^{n} w_t b_t$ where b_t is the tth largest element of the aggregates $\{S_1, \ldots, S_n\}$.

Note that in the definition of OWA we have used the notation M to identify the aggregation operator with the mediation rule, S^n to make reference to the preferences of the agents, and G to define the group preference. In the OWA aggregation the weights are not directly associated with a particular argument but with the ordered position of the arguments. If ind is an index function such that $ind(t)$ is the index of the tth largest argument, then we can express $M(S_1, \ldots, S_n) = \sum_{t=1}^{n} w_t S_{ind(t)}$. It can be shown that OWA aggregation has the following properties:

1. Commutativity: The indexing of the arguments is irrelevant

2. Monotonicity: If $S_i \geq \hat{S}_i$ for all i then $M(S_i, \ldots, S_n) \geq M(\hat{S}_i, \ldots, \hat{S}_n)$

3. Idempotency: $M(S, \ldots, S) = S$

4. Boundedness: $Max_i[S_i] \geq M(S_i, \ldots, S_n) \geq Min_i[S_i]$

Under these conditions the OWA operator is a mean operator. The form of the aggregation is dependent upon the associated weighting vector. We have a number of special cases of weighting vector are worth noting. The vector W^* defined such that $w_1 = 1$ and $w_t = 0$ for all $t \neq 1$ gives us the aggregation $Max_i[S_i]$. Thus, it provides the largest possible aggregation. The vector W_* defined such that $w_n = 1$ and $w_t = 0$ for all $t \neq 1$ gives the aggregation $Min_i[S_i]$. The weighting vector W_{ave} defined such that $w_t = 1/n$ gives us the average $\frac{1}{n}\sum_{i=1}^{n} S_i$. Finally, an interesting family of OWA operators are the E-Z OWA operators. There are two families. In the first family we have $w_t = 1/q$ for $t = 1$ to q, and $w_t = 0$ for $t = q+1$ to n. Here we are taking the average of the q largest arguments. The other family defines $w_t = 0$ for $t = 1$ to q, and $w_t = \frac{1}{n-q}$ for $t = q+1$ to n. We can see that this operator can provide a softening of the original *min* and *max* mediation rules by modifying q.

3.2 Quantifier Guided Aggregation

In the preceding, we have seen how the OWA operators can be used to compute the group preference for different alternatives, in our case, the different contracts in the current mesh $x^{+o}(k)$. However, our aim is to define consesus policies in the form of a linguistic agenda for our mediation mechanisms. For example, the mediator should make decisions regarding the exploration of the negotiation space, i.e. expansion and contraction of the mesh, following mediation rules like

Most agents must be satisfied by the contract, *at least* α agents must be satisfied by the contract, *many* agents must be satisfied, ...

The above statements are examples of *quantifier guided aggregations*. Zadeh [18] suggested a formal representation of these linguistic quantifiers using fuzzy sets. He suggested that any relative linguistic quantifier can be expressed as a fuzzy subset Q of the unit interval $I = [0, 1]$. In this representation for any proportion $y \in I$, $Q(y)$ indicates the degree to which y satisfies the concept expressed by the term Q. In most applications of the quantifier guided aggregation we use a special case class of these liguistic quantifiers, called *Regular Increasing Monotone* (RIM) quantifiers. These types of quantifiers have the property that as more agents are satisfied our overall satisfaction can't decrease. Formally, these quantifiers are characterized in the following way: 1) $Q(0) = 0$, 2) $Q(1) = 1$ and 3) $Q(x) \geq Q(y)$ if $x > y$. Examples of this kind of quantifier are *all*, *most*, *many*, *at least* α. Two examples of RIM quantifiers are *all* which is represented by Q_* where $Q_*(1) = 1$ and $Q_*(x) = 0$ for all $x \neq 1$, and *any* which is defined as $Q^*(0) = 0$ and $Q^*(x) = 1$ for all $x \neq 0$.

The question now is how to obtain the OWA operator to satisfy a quantifier guided aggregation. Again assume we have a collection of n agents. These agents have their preferences represented as fuzzy subsets over the set of alternatives in the mesh $\{S_1(x^{+o}(k)), \ldots, S_n(x^{+o}(k))\}$. Under the quantifier guided mediation approach a group mediation protocol is expressed in terms of a linguistic quantifier Q indicating the proportion of agents whose agreement if necessary for a solution to be acceptable. The basic form of the mediation rule in this approach is

Q agents must be satisfied by the contract,

where Q is a quantifier.

The formal procedure used to implement this mediation rule is described in the following. The quantifier Q is used to generate an OWA weighting vector W of dimension n. This weighting vector is then used in an OWA aggregation to determine the group support for the contract. For each contract in the mesh the argument of this OWA aggregation is the degree of support for that contract by each of the agents, $S_i(x^{e_j}(k))$, $i = 1, \ldots, n$. Thus, the process used in the quantifier guided aggregation is as follows:

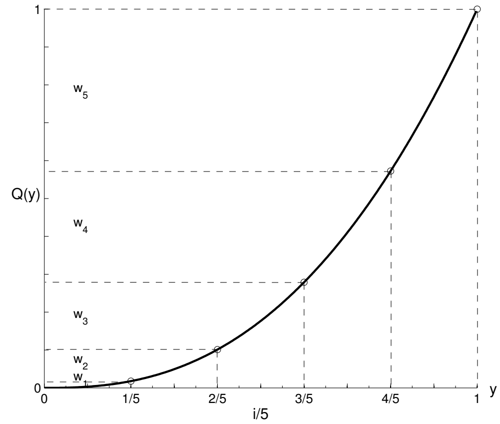


Figure 2: Example of how to obtain the weights from a quantifier for $n = 5$ agents.

1. Use Q to generate a set of OWA weights, $w_1, \ldots, w_n$.
2. For each contract $x^{e_j}(k)$ in $x^{+o}(k)$ calculate the overall group support $G(x^{e_j}(k)) = M(S_1(x^{e_j}(k)), \ldots, S_n(x^{e_j}(k)))$.

The procedure used for generating the weights from the quantifier is to divide the unit interval into n equally spaced intervals and then to compute the length of the mapped intervals using Q

$$w_t = Q(\frac{t}{n}) - Q(\frac{t-1}{n}) \,\text{for}\, t = 1, \ldots, n\,.$$

Because of the nondecreasing nature of Q it follows that $w_t \geq 0$. Furthermore from the regularity of Q, $Q(1) = 1$ and $Q(0) = 0$, it follows that $\sum_t w_t = 1$. Thus we can see that the weights generated are an acceptable class of OWA weights.

In Figure 2 we show an example of a RIM linguistic quantifier and illustrate the process of determining the weights from the quantifier. We see that the weights depend on the number of agents as well as the form of Q. In Figure 3 we show the functional form for the quantifiers *all*, *any*, Q_*, Q^*, *at least* α *percent*, *linear quantifier*, *piecewise* Q_{Z_β} and *piecewise* Q_{Z_α}. The quantifiers *all*, *any* and *at least* α describe the consensus policy using a natural language verbal description. However, more generally any function $Q : [0, 1] \to [0, 1]$ such that $Q(x) \geq Q(y)$ for $x \geq y$, $Q(1) = 1$ and $Q(0) = 0$ can be seen to be an appropriate form for generating mediation rules or consensus policies. Thus there are two techniques to generating these quantifier based mediation rules. One possibility is to start with a linguistic expression and then obtain Q. The second approach is to allow the mediation rule to be directly expressed in terms of a function Q. One

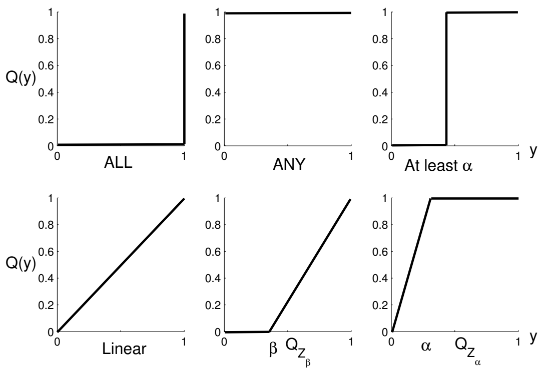


Figure 3: Functional form of typical quantifiers: all, any, at least, linear, piecewise linear Q_{Z_β} and piecewise linear Q_{Z_α}.

important characteristic of this second method is that we can easily introduce into our mediation a number of formal properties that are not very easily expressed using a verbal description of the quantifier. The linear quantifier $Q(y) = y$ for instance generates $w_t = 1/n$, and thus, all the agents get the same weight. The Q_{Z_β} quantifier it is required that at least β agents are satisfied to initiate a Q linear improvement. Q_{Z_α} initiates the Q linear improvement with the first satisfied agent, and once there are α agents satisfied there is no improvement in Q if more agents are satisfied.

One feature which distinguishes the different types of mediation rules is the power of an individual agent to eliminate an alternative. For example, in the case of *all* this power is complete. In order to capture this idea the *Value Of Individual Disapproval* (VOID)

$$VOID(Q) = 1 - \int_0^1 Q(y)dy$$

measures this power. For the *all*, *any*, *at least* α and *linear* quantifiers the VOID measures are respectively 1, 0, α and 0.5. For the Q_{Z_β} quantifier $VOID(Q_{Z_\beta}) = \frac{1}{2} + \frac{\beta}{2})$ and therefore $VOID(Q_{Z_\beta}) \in [0.5, 1]$. The Q_{Z_α} quantifier gets $VOID(Q_{Z_\alpha}) = \frac{\alpha}{2}$ and $VOID(Q_{Z_\alpha}) \in [0, 0.5]$.

Another family of quantifiers are those defined by $Q_p(y) = y^p$ for $p > 0$. In this case $VOID(Q_p) = 1 - \int_0^1 r^p dr = \frac{p}{p+1}$. For this quantifier we can easily obtain the OWA weights with

$$w_t = \left(\frac{t}{n}\right)^p - \left(\frac{t-1}{n}\right)^p .$$

For Q_p we see that as p increases we get closer to the *min* and that as p gets closer to zero we get the *max*.

4. THE SEARCH PROCESS

The search process is based on a mechanism whereby the mediator decides if to generate a new mesh in order to continue with a new negotiation round, or if to finish the negotiation process. This process starts just after any aggregation of preferences process, when the mediator has determined the group preferred contract $x^{e_*}(k)$. The relevant information available to the mediator at this point is at least the group preference $G(x^{+o}(k))$, the preferred contract $x^{e_*}(k)$, the current step-length $\triangle_k$, and the current round number k. With this information, the mediator has to select among three possible alternatives:

1. Move to the group preferred contract $x(k+1) = x^{e_*}(k)$ in $x^+(k)$ and expand the mesh by a factor of two $\triangle_{k+1} = 2 \cdot \triangle_k$.
2. Keep the current contract $x(k+1) = x(k)$ and reduce by half the mesh step-length $\triangle_{k+1} = \triangle_k/2$.
3. Finish the negotiation process.

For this paper we will assume what we call the *Standard Search Process* which selects among the mentioned alternatives as follows.

The mediator selects alternative 1 if the preferred contract is in $x^+(k)$, i.e., $x^{e_*}(k) \in x^+(k)$. If the preferred contract is $x(k)$ then the mediator selects alternative 2. Finally, we define two stopping rules, one which bounds the maximum number of rounds k_{max}, and a second one which stops negotiation when the step-length $\triangle_k$ is below a predefined threshold γ. We assume that in both cases the agreement reached is the preferred group contract in the last negotiation round.

4.1 Preferred Contract Selection in the Search Process

Here are described in detail the mechanisms used to select the preferred contract. The point of departure is the set of final group preferences for the contracts in $x^{+o}(k)$ at round k. We propose a probabilistic selection process to select the winner contract in the mesh at a round k. We associate with each contract $x^{e_j}(k) \in x^{+o}(k)$ a probability

$$P(x^{e_j}(k)) = \frac{G(x^{e_j}(k))^\sigma}{\sum_j G(x^{e_j}(k))^\sigma} .$$

The process selects the winner contract using a biased random experiment with these probabilities. The parameter $\sigma > 0$ works as an indication of the significance we give to the final group preferences. If $\sigma \to \infty$ we select the contract with the maximum support, which means that the mediator is given the higher significance to the group preferences. If $\sigma = 1$ then the probability of selecting $x^{e_j}(x)$ would be proportional to its group support. The rationale behind using this probabilistic process is to introduce randomness and avoid local optima in the following way.

With G the mediator is able to select a contract within the mesh. However, this selection is based on a relative measurement and it is not considering how good is the selection made. The mediator must consider both the G value and the relative values to make the decision of expansion and contraction. Thus, we make σ vary as a function of G and the number of rounds k. If G is high, σ must be high, favouring a deterministic mesh movement, i.e. with a high probability the contract with a higher G is selected. Otherwise, if G is low, σ must be low to induce randomness and avoid local

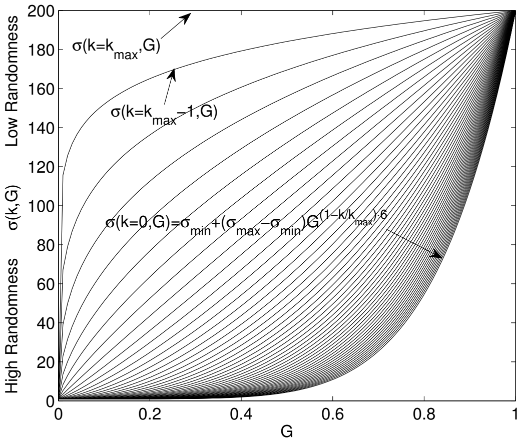


Figure 4: Evolution of $\sigma(k, G)$ for $k_{max} = 50$, $\alpha = 6$, $\sigma_{max} = 200$ and $\sigma_{min} = 1$.

optima. More specifically, for $\sigma = 0$ the selection of contracts is equiprobable, making such selection independent of G. For $\sigma = 1$ the selection probability is proportional to G. Higher values for σ increases the probability of choosing the contract with a higher G. To control σ we define

$$\sigma(k, G) = \sigma_{min} + (\sigma_{max} - \sigma_{min}) \cdot G^{(1-\frac{k}{k_{max}})\cdot\alpha} ,$$

where σ depends on the negotiation round k, the maximum number of rounds k_{max} and G. The function is bounded by σ_{max} and σ_{min} given $G = 0$ and $G = 1$ respectively. The parameter $\alpha > 0$ determines the curvature of $\sigma(k, G)$. As the number of rounds k increases, the function increases its concaveness, which means that G induces higher values for σ, favouring convergence. Figure **??** shows the evolution of $\sigma(k, G)$ for k_{max} =50, $\alpha = 6$, $\sigma_{max} = 200$ and $\sigma_{min} = 1$. The principle of this approach is analogous to the simulated annealing technique without reannealing. We can also introduce reannealing for $k_r < k_{max}$ such that k/k_{max} converts into $\frac{k-k_r}{k_{max}-k_r}$.

5. EXPERIMENTAL EVALUATION

In this section, we test our negotiation framework and show that the mechanisms proposed provide the mediator the tools to efficiently conduct multiagent negotiations by considering different consensus policies.

In the experimental setup, without loosing generality, we have considered 7 agents, 2 issues and 2 different types of negotiation spaces: a negotiation space where agents' utility functions are strategically built to define a *proof of concept negotiation scenario*, and a *complex negotiation scenario* where utility functions exhibit a more complex structure. In both cases utility functions are built using an aggregation of *Bell functions*. This type of utility functions capture the intuition that agents' utilities for a contract usually decline gradually with distance from their ideal contract. Bell functions are ideally suited to model, for instance, spatial and temporal preferences [14, 11]. In addition, they provide with the capability of configurating different negotiation scenarios in terms of different complexity degrees.

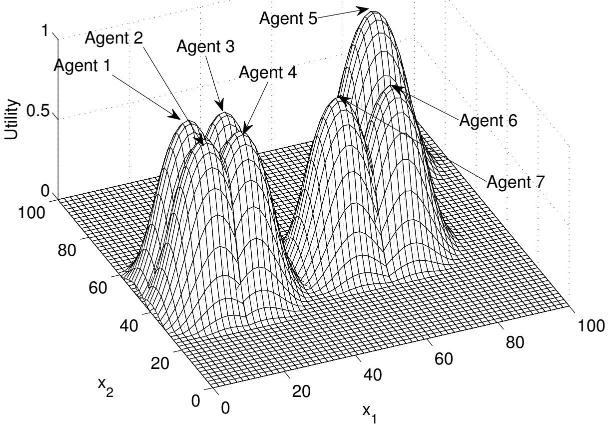


Figure 5: Utility functions for the *proof of concept negotiation scenario*.

DEFINITION 5.1. *A* Bell *is defined by a center c, height h, and a radius r. Let $\parallel s - c \parallel$ be the euclidean distance from the center c to a contract s, then the* Bell function *is defined as*

$$fbell(s, c, h, r) = \begin{cases} h - 2h\frac{\|s-c\|^2}{r^2} & if \parallel s - c \parallel < r/2, \\ \frac{2h}{r^2}(\parallel s - c \parallel - r)^2 & if\, r > \parallel s - c \parallel \geq r/2 \\ 0 & \parallel s - c \parallel \geq r \end{cases} ,$$

and the Bell utility function *as*

$$U_{b,s}(s) = \sum_{i}^{nb} fbell(s, c_i, h_i, r_i) ,$$

where nb is the number of generated bells. The complexity of the negotiation space can be modulated by varying c_i, h_i, r_i and nb.

In the *proof of concept negotiation scenario* each agent has a utility function with a single optimum. Figure 5 shows in the same graph the agents' utility functions in the bidimensional negotiation space $[0, 100]^2$. In this scenario four agents (Agent 1, 2, 3, 4) are in weak opposition (i.e. their preferences are quite similar), Agents 6 and 7 are in weak opposition and in very strong opposition with respect the other agents, and Agent 5 is in very strong opposition with respect the rest of the agents. In the *complex negotiation scenario* each agent's utility function is generated using two randomly located bells. The radius and height of each bell are randomly distributed within the ranges $r_i \in [20, 35]$ and $h_i = [0.1, 1]$. Figure 6 shows the utility functions generated for each agent in this second case.

The configuration of parameters in the mediator is: $k_{max} =$ 50 rounds, mesh tolerance $1e - 6$, and $\alpha = 2$, $\sigma_{min} = 1$, $\sigma_{max} = 200$ for the preferred contract selection process. Previous experiments have confirmed that these parameter values perform well under most negotiation scenarios.

We tested the performance of the protocol under the proof

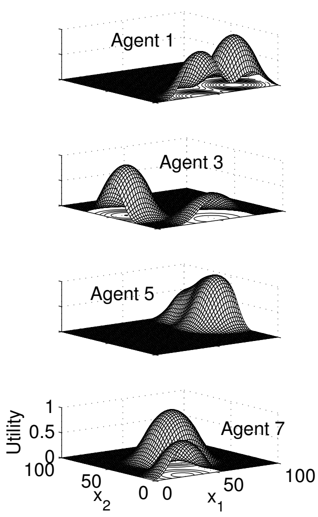


Figure 6: Utility functions for the *complex negotiation scenario.*

of concept and complex negotiation scenarios for 5 different consensus policies defined by the corresponding VOIDNESS degrees: 0, 0.25, 0.5, 0.75 and 0.95, using the quantifier $Q_p(y) = y^p$. We also define a contrast experiment where the consensus policy based mediation process is deactivated, such that the mediator uses the pattern search based process but there is no randomness and the group preference evaluation is limited to compute the sum of agents' valuations for a given contract (i.e. the winner contract is that with the highest sum of valuations).This experiment uses also 50 rounds and a mesh tolerance $1e-6$.

Each experiment consist of 100 negotiations where we capture the utilities achieved by each agent. To analyze the results we first build a 7 agents $\times$ 100 negotiations utility matrix where each row provides each agent's utilities and each column is a negotiation. The matrix is then reorganized such that each column is individually sorted from higher to lower utility values. Note that after this transformation the association row/particular-agent disappears. Given the matrix, we form 7 different utility groups: a first group named *group level 1* where we take the highest utility from each negotiation (i.e. the first row), a second group named *group level 2* with the two first rows and so on. In order to show the performance of the protocol we have used the Kaplan-Meier estimate of the cumulative distribution function (*cdf*) of agents' utilities for each group. Thus, we compute the *cdf* for the highest utilities, for the two highest utilities and so on. The *cdf* estimates the probability of finding agent's utilities below a certain value. The rationale behind using grouping in the analysis is to evaluate the ability of the protocol to find solutions which satisfy groups of agents.

In the proof of concept scenario (see Figure 5) it can be seen that when a quorum is needed, the best alternative is to get satisfied agents 1, 2, 3 and 4. If it is enough to have one agent satisfied, any of the utility peaks would be a good solution. In Figure 7 we show the results for the proof of

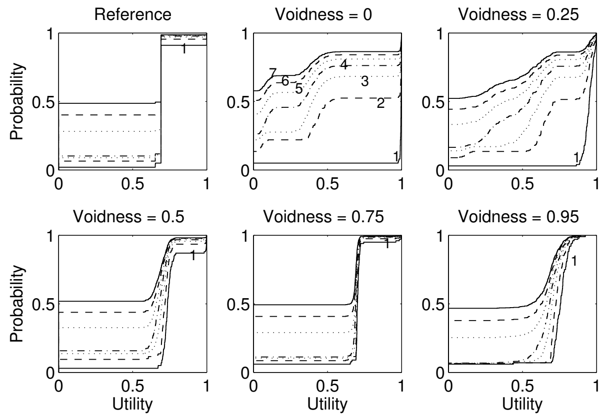


Figure 7: Cumulative distributions of utilities for the *proof of concept scenario.*

concept scenario. Each line shows the *cdf* for a group level and the number above each line identifies the corresponding level. For instance, for the reference experiment and the group level 1 there is approximately a 98% probability of having agents with a utility 0.7, and a 2% probability of having agents with utility 0. In the group level 7 case, there is a 50% probability of having agents with utility 0.7, and a 50% probability of having agents with utility 0. For a VOIDNESS=0 and group level 1, however, the probability of having agents with a utility 1 is around 98%, which means that the mediator is applying efficiently the consensus policy which states that it is good enough to have one agent satisfied. As VOIDNESS increases (i.e. as it is necessary to have more agents satisfied) the *cdf* for group level 1 performs worse, though better than in the reference scenario, and for higher group levels the performance increases.

In Figure 8 are shown the results for the complex negotiation scenario. Here we can also see how as VOIDNESS increases, the mediator biases the search for agreements where more agents are satisfied at the expense of not having individual agents highly satisfied. Globally, the results show that the proposed mechanisms are able to focus the negotiation process in terms of consensus policies and to obtain better results than when using a classical welfare maximization approach.

6. CONCLUSION

The main hypothesis of our work is that the consensus type by which an agreement meets in some specific manner the concerns of all the negotiators should be considered in the construction of multiparty negotiation protocols. We argue that there exist situations where an unanimous agreement is not possible or simply the rules imposed by the system may not seek such unanimous agreement. Thus, we develop a consensus policy based mediation framework to perform multiparty negotiations. The mediation mechanisms proposed to perform the exploration of negotiation space in the multiparty negotiation setting are derived from the Generalized Pattern Search non-linear optimization technique. The exploration performed in the mediator is guided by the aggregation of the agent preferences on the set of alternatives the mediator proposes in each negotiation round. The medi-

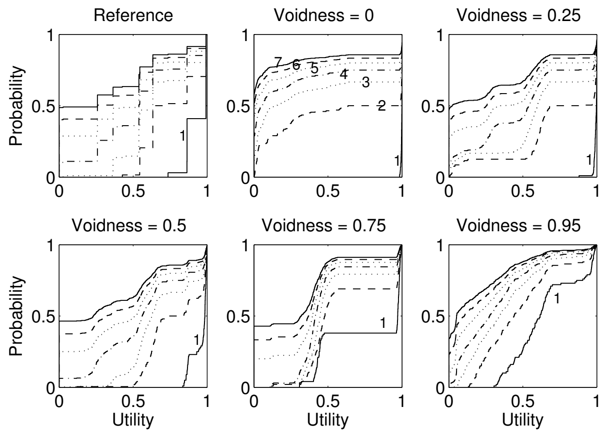


Figure 8: Cumulative distributions of utilities for the *complex negotiation scenario*.

ation rules at the mediator may take the form of a linguistic description of the type of agreements needed. We showed empirically that CPMF efficiently manages negotiations following predefined consensus policies and solves situations where unanimous agreements are not viable.

We believe that the negotiation framework presented opens the door to a new set of negotiation algorithms where consensus criteria will play an important role. However, the strategical issue remains opened. We have assumed that agents reveal their true valuations. It is expected that the performance of the protocol deviates from the optimal if agents act strategically. Thus, the strategy issue needs to be evaluated, and mechanisms need to be implemented to avoid or mitigate the incentive compatibility problem.

7. ACKNOWLEDGMENTS

This work has been supported by the Spanish Ministry of Education and Science grants: TIN2008-06739-C04-04, T2C2 research project, and the mobility grant JC2010-003.

8. REFERENCES

[1] H. Ehtamo, R. P. Hamalainen, P. Heiskanen, J. Teich, M. Verkama, and S. Zionts. Generating pareto solutions in a two-party setting: constraint proposal methods. *Management Science*, 45(12):1697–1709, 1999.

[2] U. Endriss, N. Maudet, F. Sadri, and F. Toni. Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*, 25:315–348, 2006.

[3] P. Heiskanen, H. Ehtamo, and R. P. Hamalainen. Constraint proposal method for computing pareto solutions in multi-party negotiations. *European Journal of Operational Research*, 133(1):44–61, 2001.

[4] T. Ito, M. Klein, and H. Hattori. A multi-issue negotiation protocol among agents with nonlinear utility functions. *Journal of Multiagent and Grid Systems*, 4(1):67–83, 2008.

[5] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Protocols for negotiating complex contracts. *IEEE Intelligent Systems*, 18(6):32–38, 2003.

[6] S. Kraus. *Strategic Negotiation in Multiagent Environments*. Mit Press, 2001.

[7] G. Lai, C. Li, K. Sycara, and J. Giampapa. Literature review on multiattribute negotiations. Technical Report CMU-RI-TR-04-66, Robotics Institute, Carnegie Mellon University, Pittsburgh, USA, December 2004.

[8] G. Lai and K. Sycara. A generic framework for automated multi-attribute negotiation. *Group Decision and Negotiation*, 18:169–187, 2009.

[9] R. M. Lewis, V. Torczon, and M. W. Trosset. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, 124:191–207, 2000.

[10] M. Li, Q. B. Vo, and R. Kowalczyk. Searching for fair joint gains in agent-based negotiation. In Decker, Sichman, Sierra, and Castelfranchi, editors, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1049–1056, Budapest, Hungary, May, 10-15 2009.

[11] M. A. Lopez-Carmona, I. Marsa-Maestre, E. de la Hoz, and J. R. Velasco. A region-based multi-issue negotiation protocol for non-monotonic utility spaces. *Computational Intelligence*, In press:1 – 48, 2011.

[12] M. A. Lopez-Carmona, I. Marsa-Maestre, G. Ibanez, J. A. Carral, and J. R. Velasco. Improving trade-offs in automated bilateral negotiations for expressive and inexpressive scenarios. *Journal of Intelligent & Fuzzy Systems*, 21:165 – 174, 2010.

[13] M. A. Lopez-Carmona, I. Marsa-Maestre, M. Klein, and T. Ito. Addressing stability issues in mediated complex contract negotiations for constraint-based, non-monotonic utility spaces. *Journal of Autonomous Agents and Multiagent Systems*, Published online:1 – 51, 2010.

[14] I. Marsa-Maestre, M. A. Lopez-Carmona, J. R. Velasco, T. Ito, M. Klein, and K. Fujita. Balancing utility and deal probability for auction-based negotiations in highly nonlinear utility spaces. In *21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 214–219, Pasadena, California, USA, July 2009.

[15] Q. B. Vo, L. Padgham, and L. Cavedon. Negotiating flexible agreements by combining distributive and integrative negotiation. *Intelligent Decision Technologies*, 1(1-2):33–47, 2007.

[16] R. Yager. Quantifier guided aggregation using owa operators. *International Journal of Intelligent Systems*, 11:49 – 73, 1996.

[17] R. Yager and J. Kacprzyk. *The Ordered Weighted Averaging Operators: Theory and Applications*. Kluwer, 1997.

[18] L. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computing and Mathematics with Applications*, 9:149 – 184, 1983.

A Scenario Generation Framework for Consistent Comparison of Negotiation Approaches

Ivan Marsa-Maestre[*], Miguel A. Lopez-Carmona[†] and Mark Klein
Center for Collective Intelligence
Massachusetts Institute of Technology
5 Cambridge Center NE25-754, Cambridge MA 02139, USA
{ivmarsa,m_lopcar,m_klein}@mit.edu

ABSTRACT

There is a number of recent research lines addressing automated complex negotiations. Most of them focus on overcoming the problems imposed by the complexity of negotiation scenarios which are computationally intractable, be it by approximating these complex scenarios with simpler ones, or developing heuristic mechanisms to explore more efficiently the solution space. The problem with these mechanisms is that their evaluation is usually restricted to very specific negotiation scenarios, which makes very difficult to compare different approaches, to re-use concepts from previous mechanisms to create new ones or to generalize mechanisms to other scenarios. This makes the different research lines in automated negotiation to progress in an isolated manner. A solution to this recurring problem might be to create a collection of negotiation scenarios which may be used to benchmark different negotiation approaches. This paper aims to fill this gap by providing a framework for the characterization and generation of negotiation scenarios intended to address this problem, facilitating in this way that researchers compare and share their advancements. Experiments show how the proposed framework is able to generate scenarios which can be effectively used to compare the performance of different negotiation approaches.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multi-agent systems*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*coherence and coordination*

General Terms

Algorithms, Measurement, Experimentation

[*]Visiting Researcher from the Universidad de Alcala, Spain. Contact him also at ivan.marsa@uah.es

[†]Visiting Researcher from the Universidad de Alcala, Spain. Contact him also at miguelangel.lopez@uah.es

Keywords

Negotiation, Scenario generation, testbed

1. INTRODUCTION

Automated negotiation provides an important mechanism to reach agreements among distributed decision makers [12, 13]. It has been extensively studied from the perspective of e-commerce, though it can be seen from a more general perspective as a paradigm to solve coordination and cooperation problems in complex systems, providing a mechanism for autonomous agents to reach agreements on, e.g., task allocation, resource sharing, or surplus division.

A variety of negotiation models have been proposed, yielding promising results in a wide range of negotiation problems [4]. However, most of these approaches are evaluated for negotiation scenarios meeting very specific requirements. Given the vast variety of negotiation problems, a recurrent challenge automated negotiation researches have to face is how to justify the models and mechanisms they propose are suitable to solve or model different problems, or how to compare their approaches and methods with the ones of other researchers. In the best cases, there are a few number of previous works similar enough to the new proposal to make a comparison. In most cases, however, this comparison is not possible due to the diversity of scenarios the different research groups deal with, so the different research lines progress in an isolated manner. In addition, though there exist multiple surveys about negotiation in the literature [10, 1], they are more intended to classify the different approaches (mediated, nonmediated, one-shot, iterative...) than to describe or classify the different negotiation problems.

On the contrary, our research focuses on the properties of the negotiation scenario regardless of the approach which may later be used to address it. What we intend is to be able to measure a set of properties of a given negotiation scenario, and to be able to generate negotiation scenarios which have desired values on those properties. The need to have negotiation scenario testbeds to provide reproductivity and coherence to works from different authors has been acknowledged in multiple occasions [8], and there exist some generators and testbeds which allow to standardize the scenarios to a certain extent, though they usually focus on specific negotiation protocols [15] or specific preference representations [6]. In addition, they usually generate scenarios according to low-level properties (e.g. weights for the different issues in a linear additive model, number and width of constraints in a weighted constraints model...) rather than high-level, meaningful scenario properties, such as the structure of the agent utility functions (i.e. autocorrelation, epistasis) or the relationships between the utility functions of the different agents (e.g. shape of the Pareto front).

Our aim in this line is to provide a framework which allows to characterize and generate negotiation scenarios according to high-level properties. The benefit of using such a framework would be twofold. On one hand, it would allow to make it easier to test negotiation mechanisms in a much wider range of scenarios, as well as to compare different approaches in the same scenarios. On the other hand, it would facilitate the creation, by the research community, of a database of negotiation approaches and scenarios according to these high-level properties. In this way, it would be easier to find out which negotiation mechanisms work better for the different subsets of the negotiation problem space. Finally, this would open the door to the rigorous assessment of the applicability of negotiation approaches to real-world problems. For a given real negotiation problem, we could measure the high-level properties of the scenario and use them to find in the database the negotiation approach which performs better for scenarios matching these properties. In this paper we intend to contribute to this goal in the following ways:

- We provide a set of tools which allow to measure high-level properties of a negotiation scenario. This include both structural properties of the agents' utility functions and properties derived from the relationship between the different utility functions (Section 2).
- We propose a negotiation scenario generator which considers the properties outlined above (Section 3). It is based on building utility functions as aggregations of hyper-volumes, and on sharing hyper-volumes among agent utility functions to model zones of potential agreement (or disagreement). This allows us to model scenario complexity in two orthogonal dimensions: the scarcity of mutually acceptable solutions and the difficulty to locate these solutions in the contract space.

A set of experiments have been performed to validate our generator and to asses the possibilities it may bring to the research community on automated negotiation. These experiments are described in Section 4, along with the discussion of the results obtained. Finally, the last section summarizes our contributions and sheds light on some future research.

2. CHARACTERIZING NEGOTIATION SCENARIOS

As we have stated before, our aim is to be able to characterize negotiation scenarios using high-level properties. To do this, we should first define what we understand as negotiation scenarios. Different authors agree that there are three key components in a negotiation model [10]: an interaction protocol which defines the rules of encounter among the negotiating agents, a set of decision mechanisms and strategies which govern agents' decision making, and the preference sets of the different agents which allow them to assess the different solutions in terms of gain or utility and to compare them. From this three components, we can easily see that both the interaction protocol and the decision mechanisms and strategies are more related to the way the model solves the negotiation problem than to the negotiation problem itself. Therefore, in the following we characterize a negotiation scenario according to the preferences of the agents taking part in the negotiation.

2.1 Agent preferences in negotiation scenarios

From the decision theory perspective, preferences express the absolute or relative satisfaction for an individual about a particular choice among different options. In [2], agent preference structures are classified in four broad families: binary, ordinal, cardinal and fuzzy preference structures. Among these families, cardinal preference structures are probably the most widely used in automated negotiations, and are the ones we will be focusing in in the following. In particular, it is usual to define agent preferences by means of utility functions.

Formally, for a given multi-attribute domain $\langle X, D, Ag, U\rangle$, the *utility function* for each agent $j \in Ag$ is defined as

$$U^j : D \to \mathbb{R},$$

assigning to each possible combination of values in X or *deal* $s = \{s_i | i = 1, ..., n; s_i \in d_i\}$ a real number, which represents the utility that deal s yields for agent j.

There are vastly different utility functions in the negotiation literature. Monotonic negotiation scenarios are usually modelled simulated with *Constant Elasticity of Substitution* (CES) utility function [18], which are widely used in economics as production functions, and in consumer theory as utility functions. An example of a CES utility function for a utility space of n issues could be

$$U(s) = (\sum_{i=1}^{n} \alpha_i \cdot x_i)^{1/\beta},$$

where $s = (x_1, x_2, \ldots, x_n)$ is a contract and x_i the ith issue, α_i is the share parameter for issue i, and β is the elasticity of substitution parameter. An interesting property of CES utility functions is that they also model linear utility spaces if we set the elasticity parameter to 1.

To represent non-monotonic utility spaces, we can use for instance k-additive utility functions [7]. Another widely used way to represent preferences and utility functions is the use of constraints over the values of the attributes. A particular case of constraint-based utility representation which has been used to model complex utility spaces for negotiation are *weighted constraints* [9]. There is a utility value for each constraint, and the total utility is defined as the sum of the utilities of all satisfied constraints. More formally, the utility space of the agents may be defined as a set of constraints $C = \{c_k | k = 1, ..., l\}$. Each constraint c_k has an associated utility value $u(c_k)$. If we note as $s \in x(c_k)$ the fact that a given contract $s = \{s_i | i = 1, ..., n\}$ is in the set of contracts that satisfy constraint c_k, an agent's utility for contract s

may be defined as

$$u(s) = \sum_{c_k \in C | s \in x(c_k)} u(c_k),$$

that is, the sum of the utility values of all constraints satisfied by s. This kind of utility functions produces nonlinear utility spaces, with high points where many constraints are satisfied, and lower regions where few or no constraints are satisfied. Due to the hypercube shape of the constraints, the utility functions defined in this way are discontinuous.

An example of a utility representation for continuous, non-monotonic utility spaces can be found in [16]. Here the authors model the utility space of an agent as a sum of *bell-shaped* functions of the form

$$fbell(s,c,h,r) = \begin{cases} h - 2h\frac{\|s-c\|^2}{r^2} & if \ \| s-c \| < r/2, \\ \frac{2h}{r^2}(\| s-c \| - r)^2 & if \, r > \| s-c \| \geq r/2\,, \\ 0 & \| s-c \| \geq r \end{cases}$$

so that the utility function is

$$U(s) = \sum_{i}^{nb} fbell(s, c_i, h_i, r_i)\,,$$

where, c_i, h_i and r_i] are the parameters defining each bell, and nb is the number of bells in the utility functions.

In [19] utility graphs are used to model issue interdependencies for binary-valued issues. Utility graphs are used to decompose highly non-linear utility functions in sub-utilities of clusters of inter-related items. In a different way, in [22] the concept of hierarchical negotiation problems is introduced. Hierarchical negotiation problems are those in which the problem domain can be structured in layers, with different issues being relevant for the different layers. In this way, a negotiation problem involving a high number of interdependent issues may be addressed hierarchically, exploring only a subset of the issues at each layer of the hierarchy. In this way, the internal structure that domain elements have for many real-world negotiation problems could be exploited to allow for a more efficient search for solutions. Here we do not have a single utility function, but a hierarchical tree of utility, where at each layer a function on a subset of the issues allows to decide which is the relevant branch to select in the lower levels of the hierarchy.

The above is just a brief review of a selection of the different kinds of preference representations used in the most relevant works in the field. From the formulations and descriptions we can see the inherent difficulty for the direct comparison of approaches which are intended to work in different kinds of scenarios, and for determining which of the existing approaches would be most effective to address a new scenario. We can wonder, for instance, whether the protocols proven successful for constraint-based utility spaces could be applied, for instance, to hierarchical negotiation scenarios, or whether protocols intended to work with bell utility functions could be applied, with some modifications, to CES-based utility spaces. However, direct comparison of the approaches is often very difficult (if not unfeasible) due to the important differences between the scenarios. Even if a given negotiation mechanism could be applied to two different negotiation scenarios, it is very difficult to establish equivalencies between them, due to the vast differences between the settings of the different scenarios. For instance, [9] describes experiments for a negotiation scenario involving constraint-based utility space, where there are constraints with widths drawn uniformly from the interval $[3, 7]$ (in a domain $[0, 9]$). If we move onto a bell-based utility spaces, now we need to define it in terms of bell radii and heights. Which values would yield a utility function of similar complexity? We believe that such comparison of approaches could be made possible if there existed a framework for the characterization of negotiation scenarios according to a set of common properties. This is what we aim to provide in the following section.

2.2 High-level properties of negotiation scenarios

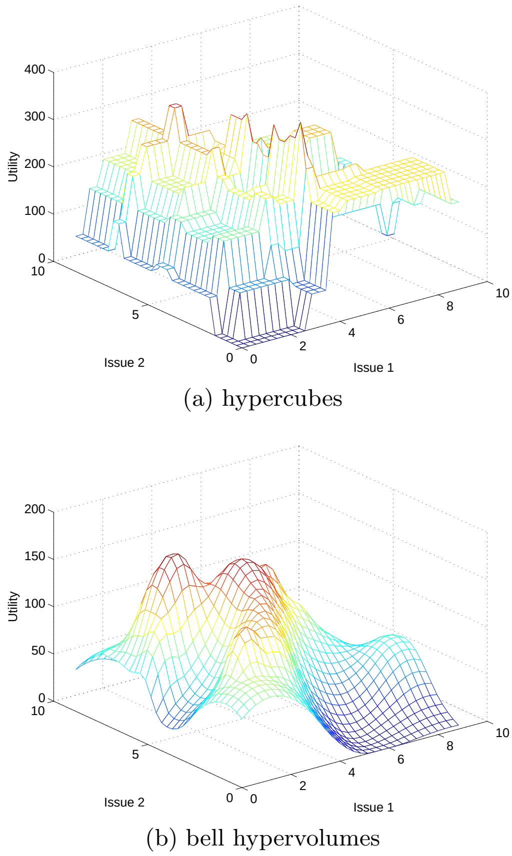
We have defined a negotiation scenario as a set of agent utility functions. Therefore, to characterize a given scenario we have to look at these utility functions and the relationships between them. The most immediate approach is to study its structural properties like the structural properties of a fitness landscape which are interesting regarding search complexity within the space, such as modality, ruggedness, smoothness and neutrality [20]. Most of the approaches we can find in the literature are based on the correlation between different samples of the fitness function f. A metric which is easy to compute in most scenarios and allows to make quantitative evaluations about the complexity of a fitness or utility landscape is *correlation length* or *correlation distance*. Correlation distance is defined as the minimum distance ψ which makes correlation fall below a given threshold (usually 0.5), which gives an idea of the distance we can move throughout the solution space while keeping a certain correlation between samples [17].

A property which is usually related to negotiation complexity is issue interdependency. Negotiations with multiple, interdependent issues are assumed to be harder than those involving independent issues [11]. There are some recent works suggesting to assess the degree of interdependency between issues in order to modify the negotiation strategy accordingly (e.g. by negotiating separately those issues which are less interdependent). However, in most cases issue interdependency is measured in an ad-hoc manner, usually restricted to the kind of utility representation used [19, 5]. Here we propose a measure based on information theory, analogous to the epistasis measure described in [21] for evolutionary computation:

$$\epsilon_{ij} = \begin{cases} \frac{I(i;U)+I(j;U)}{I(i,j;U)} - 1 & \text{if } I(i,j;U) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $I(i;U)$ and $I(j;U)$ are the amount of information that each issue i and j reveals about the value of the utility function U, and $I(i,j;U)$ is the amount of information *both* issues reveal about the value of U.

Another aspect which may be studied is the point distribution in utility space diagrams. These diagrams represent the utilities achieved by the different agents participating in the negotiation for each analyzed solution, allowing for instance to assess the distance of a solution from the Pareto front, which is used in many works as an evaluation metric for negotiation mechanisms. However, to our knowledge no authors have tried to generate negotiation scenarios from arbitrary distributions of points in the utility diagrams, while this distribution may have a great impact on the complexity of a negotiation scenario. Intuitively, a negotiation will be easier as the ratio of mutable acceptable solutions against total potential solution in high, and will be more difficult in the opposite case. The same is true with negotiation efficiently and the ratio of solutions near the Pareto front. Finally, the very shape of the Pareto front may affect significantly the properties of the negotiation scenario, or even the way to analyze it. For the purpose of assessing the complexity of a given scenario, however, it is not enough to know if there exist solutions which yield a given set of utilities to the different agents. With the same utility diagram, a scenario where an 80% of the solutions are both mutually acceptable and are in the Pareto front would probably be much less challenging for a negotiation mechanisms than one with only a 10% of Pareto-efficient, mutually acceptable solutions. Therefore, we have to consider also the number (or ratio) of solutions corresponding to each point in the utility diagram. Taking this into account, we extend the concept of utility diagrams to utility *histograms* $H(\bar{u})$, where $\bar{u}$ is a vector of utility values for the different agents, and the histogram value at $\bar{u}$ represents the number of potential solutions which yield that combination of utility values for the agents. From these utility histogram we can easily derive properties like the ratio of mutually accepted solutions and the ratio of Pareto-efficient solutions.



(a) hypercubes

(b) bell hypervolumes

Figure 1: Generation of utility spaces with weighted aggregation of hypervolumes.

3. A SCENARIO GENERATOR FOR COMPLEX AUTOMATED NEGOTIATIONS

The scenario generation tool we propose in this paper intends to take into account both the structural properties of the agent utility functions and the relationships between the utility functions of the different agents. In the following, we first describe a parametric mechanism to generate utility functions, and then an approach to control the relationship between the utility functions in a scenario through the utility diagram of the scenario.

3.1 Generation of Utility Functions by means of Hypervolumes

We aim to build a generator able to create utility functions which allow to test most of the negotiation approaches we can find so far. This is a rather ambitious goal, since, as we have seen, there are many different types of utility functions used in the negotiation literature. For the purposes of this work, we will restrict ourselves to cardinal utility functions, where contracts are mapped to real numbers which correspond to the utility values they yield. Note that cardinal utility functions may be used to represent ordinal preferences too, by restricting the range of the utility values to natural numbers.

Under this assumption, we can get a fully expressive representation of utility functions by aggregating *hypervolumes*. We define a hypervolume as a constrained cardinal function, where by constrained we mean that there may be constraints regarding when this cardinal function contributes to the utility value of the overall utility function. For instance, we can have a cardinal function $C_1(\bar{x}) = 5$, which is constrained so that it only applies for $\bar{x} \in S_1$, where S_1 is a given subset or region of the solution space S. In this way, we can use hypervolume C_1 as a weighted constraint. In a similar way, if we wanted to generate a linear utility function, we could use a hypervolume defined as a hyperplane, constrained so that it covers all the domain. Our scenario generator currently supports constant, cone, bell and CES cardinal functions as hypervolumes, though it has been designed so that new categories of hypervolumes may be added if needed.

Apart from adding hypervolumes to the utility function, we need to be able to define the aggregation operators we use to compute the overall agent utility from the hypervolumes. The generator covers a wide range of simple aggregation operators, like weighted sum (and average), maximum or minimum. Figure 1 shows three examples of utility functions generated using different kinds of hypervolumes and a weighted sum aggregation.

Finally, hypervolumes are defined to depend on a set of parameters (e.g. width, height, aspect ratio...), so that they may be varied to control the properties of the resulting utility functions. Though the generator API allows to have complete control over all the parameters, sometimes it is preferred to specify more wide-sense requirements for the generated utility functions. In order to do this, we provide sample templates which receive a set of higher-level parameters and generate utility functions from them.

For instance, we provide a template which allows to generate utility functions based on weighted hypercubes by specifying probability distributions on the width, height and placement of the hypercubes, along with the issue interdependencies. The template works as shown in Algorithm 1. Vector $L = \{l_m \,|m = 1, \ldots, n; \sum_m l_m = l\}$ controls the dependencies between issues, determining the number l_m of m-ary constraints generated (1). Each m-ary constraint is first generated as a region R placed at the origin. The length for each one of the m intervals I_i^R which comprise the constraint is generated by means of a probability distribution *dist_length* (2). Different probability distributions may be used to generate these interval length values, like for instance a uniform distribution in the interval $[w_{min}, w_{max}]$. In this way, we can control in a parametric way the correlation length of the generated utility space. Once the different intervals I_i^R have been generated, each interval is mapped to one of the n issues in the negotiation domain using a probabilistic correspondence *map_intervals* (3). This correspondence allows to control the degree of interdependency between the different issues. Finally, the generated region is moved throughout the utility space using a movement vector δ generated by means of a multidimensional probability distribution *dist_move* (4). Again, different probability functions may be used for the distribution of the constraints throughout the agent utility space. The function *restrict_domain* truncates the moved regions to bound them to the domain D (5). Finally, the weights associated to the constraints are also assigned with a probability distribution *dist_weight* (6), which can be a function of different parameters, like constraint dimension or volume, thus allowing to model different situations, like the fact that more specific constraints have more utility, which is usually the case in real scenarios.

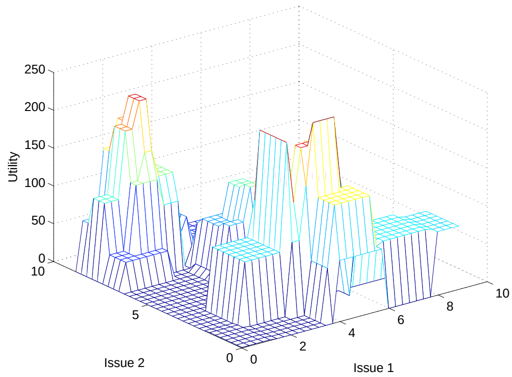
Algorithm 1: Template for the generation of utility spaces based on weighted hypercubes

Input:
- n: number of issues in the utility space
- D: utility space domain
- 1 $L = \{l_m \,|m = 1, \ldots, n; \sum_m l_m = l\}$: vector to control the distribution of hypercube dimensions
- $dist\_length(\ldots)$: probability distribution function for the generation of the intervals I_i^R
- $map\_intervals(\ldots)$: probabilistic correspondence function to map intervals to issues
- $dist\_move(\ldots)$: probability distribution function for the distribution of hypercubes throughout the utility space
- $dist\_weight(\ldots)$: probability distribution function for the weights associated to the hypercubes

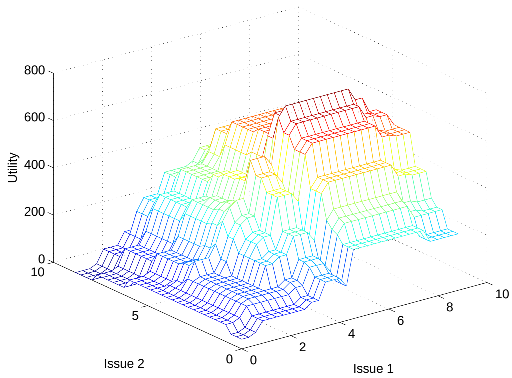
Output:
- C: constraint set
- Ω: set of weights associated to the constraints

$C = \varnothing$;
$\Omega = \varnothing$;
foreach $l_m \in L$ **do**
  $k = 0$;
  while $k < l_m$ **do**
    $R = \varnothing$;
    $d = 0$;
    while $d < m$ **do**
2       $I_d^R = dist\_length(\ldots)$;
      $R = R \cup I_d^R$;
      $d = d + 1$;
    end
3     $R' = map\_intervals(R, \ldots)$;
4     $\delta = dist\_move(\ldots)$;
5     $c = restrict\_domain(R' + \delta, D)$;
    $C = C \cup c$;
6     $\omega = dist\_weight(\ldots)$;
    $\Omega = \Omega \cup \omega$;
    $k = k + 1$;
  end
end

This template allows, for instance, to control the correlation length of the utility spaces depending on the distributions used by the generator and the distribution parameters set. In particular, using a normal distribution of widths with mean μ_w and using uniform probability distributions for interval generation, issue mapping and constraint distribution through the utility space standard deviation σ_w yields an approximate correlation distance $\psi_{0.7} = \mu_w + \sigma_w$ for a threshold 0.7. Figure 2 shows constraint-based utility spaces generated in this way.



(a) $\psi_{0.7} = 2$



(b) $\psi_{0.7} = 6$

Figure 2: Generation of utility spaces with different correlation lengths.

The generation of utility functions by means of aggregation of hypervolumes provides a flexible and expressive way to model different kind of agent preferences, and allows to control to a great extent the complexity of finding high utility regions within the utility space of an agents (by controlling, for instance, correlation length). The complexity of the individual agent utility functions, however, does not fully account for the complexity of the scenario. We may have, for instance, scenarios where agents may find very difficult to determine their high utility regions, but where once these regions have been found agreements are fairly straightforward, because high utility regions for the different agents coincide. On the other hand, we may have smooth utility functions for the agent which make very easy to locate high utility regions, but the negotiation may still be complex because *mutually acceptable* regions are hard to find. A mechanism to take into account the relationships between the utility functions of the different agents is described in the following section.

3.2 Generating negotiation scenarios from utility diagrams

As we stated above, utility diagrams are usually used to characterize negotiation scenarios, since they provide a graphical way to visualize the relationship between the potential solutions to the negotiation problem and the utility values these solutions would yield to the negotiation agents. Utility diagrams are useful, for instance, to determine the existence of mutually acceptable solutions (that is, solutions which utilities are above the reservation values of all agents), or to assess the relative efficiency of the solutions (that is, the distance from the solutions to the Pareto front). Finally, a wide range of notions for optimal solutions (e.g. Nash solution, Kalai-Smorodinsky, etc.) make use of the Pareto frontier.

What we propose here is to use these utility diagrams as the input for scenario generation, so that we are able to generate agent utility functions which match a given utility histogram. In order to generate utility functions for a given utility histogram, we propose to use *shared hypervolumes*. The idea behind shared hypervolumes is to include *similar* hypervolumes in the utility functions of the different agents, adjusting the parameters of the hypervolumes so that they generate appropriate points within the utility histogram. For instance, if we want to generate utility functions for a trivial utility histogram $H(\bar{u})$ for two agents, where the histogram value is v for $\bar{u} = \{a, b\}$ and 0 otherwise, we could achieve this by generating two utility functions which share a hypercube of volume v, with weight a for the first agent and weight b for the second. Of course, as the number of points in the utility diagram increase, the complexity of the generation process also increases, since we have to take into account the effect of the intersection between shared hypervolumes. What we do is to generate a first approximation of the utility functions by dividing the utility space in non-overlapping regions and assigning shared hypervolumes to each region, and then feed this first approximation of the utility diagram to a nonlinear optimizer which tries to minimize the approximation error.

An important property of this scenario generation strategy is that the shape and parameters of the shared hypervolumes may be varied so that additional properties of the generated functions are satisfied. For instance, we can vary the volume of the shared hypervolumes to adjust the correlation length of the utility functions. Figure 3 shows an example for two agents and weighted hypercubes, where we have generated two scenarios with identical utility diagrams and different correlation lengths. The type of hypervolumes or the aggregation operators used may be adjusted as well.

4. USING THE SCENARIO GENERATOR TO COMPARE OPTIMIZATION AND NEGOTIATION APPROACHES

We have seen that our generation tool is able to create negotiation scenarios according to high-level properties, such as correlation length and the shape of the utility histogram. However, the final purpose of the scenario generator is to serve as a tool for the comparative analysis of negotiation approaches. In this section we present a set of experiments to validate its suitability for this purpose.

4.1 Experimental Settings

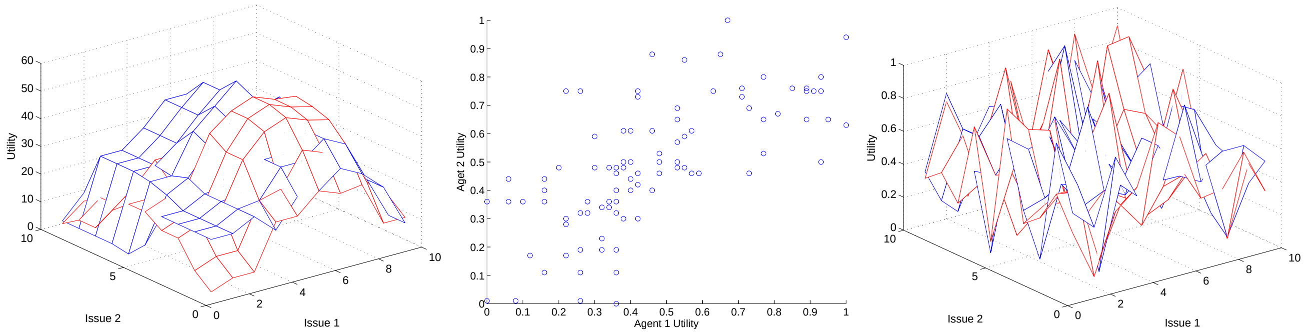
There are two main aspects which define a comparative analysis of negotiation approaches. The first is the set of different circumstances in which the approaches are evaluated, that is, the range of negotiation scenarios used to test them. In our case, this range of negotiation scenarios is given by the different dimensions the generator is able to control:

- *Correlation distance*, determined by the distribution of hypervolume "widths" in the agent utility functions. We have used correlation distances (relatives to the width of the domain) $\psi_{0.7} \in \{0.01, 0.05, 0.1, 0.5\}$.
- *Shape of the Pareto front.* We have generated scenarios corresponding to Pareto fronts of the form $u_2 = (1 - u_1)^{\frac{1}{\beta}}$, with $\beta \in \{0.25, 0.5, 1, 2, 4\}$. This accounts for highly competitive scenarios, zero sum scenarios, and scenarios where high joint gains are achievable.
- *Ratio of solutions in the Pareto front.* Since we generate scenarios according to a given utility histogram, we can control how many solutions we allow to be in the Pareto front. In this case, we have varied the ratio of Pareto-efficient solutions in the range $\rho_{Pareto} \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$.
- *Epistasis.* This first version of the generator is not able to generate specific values of epistasis while controlling the other parameters as well. We can, however, control it roughly by modifying the distribution among the different issues of the hypervolume widths (asymmetric hypervolumes yield lower epistasis for the same volume values). So we have defined *highly-epistatic* scenarios, which are generated using symmetric hypervolumes, and *lowly-epistatic* scenarios, generated using asymmetric ones.

In this range of scenarios, we have tested three negotiation approaches:

- *Similarity-based Negotiation Protocol (SBNP)*, based on the protocol proposed in [14], with a time-based concession strategy as described in [3]. This approach relies in the assumption of a monotonic.
- *Region-based Negotiation Protocol (RBNP)*, as described in [16], which is designed for non-monotonic utility spaces.
- *Complete information (CI)*: The complete agent's utility functions were passed to a multiobjective nonlinear optimizer based on genetic algorithms.

For each combination of scenario generation parameters, 10 different sets of utility functions were generated, and 10 negotiations were run for each generated function using the approaches under evaluation. The negotiation mechanisms were configured with the default values described in the referenced works. The nonlinear optimizer for the CI approach was allowed to run for the same time that the slowest of the other approaches took to complete the negotiation.



(a) $\psi_{0.7} = 5.9$ (b) utility diagram (c) $\psi_{0.7} = 1.7$

Figure 3: Scenarios with the same utility diagram and different correlation lengths.

Experiments were coded in MATLAB and run on a 2x3.2Ghz Qad-Core Intel Xeon processor with 4Gb memory using Mac OS X 10.5.4.

4.2 Experimental Results

Tables 1 and 2 show the result of the experiments for the highly-epistatic scenarios for different values of the ratio of Pareto efficient solutions ρ_{Pareto}. Each table show's the average social welfare optimality rates for the different approaches under evaluation, which are easy to compute since the maximum social welfare is known a priori (via the utility diagram). In general, the CI approach clearly outperforms the other ones, which is reasonable taking into account that it is using complete information. The results obtained by the genetic algorithm are in most cases close to the optimum regardless to the values of the different scenario parameters. However, as the ratio of Pareto-efficient solutions decrease, a significant difference can be observed when varying correlation distance and β, with the difference in β being more significant. This accounts for the fact that the crossover and mutation operators we use are based on locality, and thus for highly uncorrelated scenarios with few Pareto-efficient solutions, it is harder for the algorithm to find them. Regarding RBNP and SBNP, we can see that, again, low ratios of Pareto-efficient points make optimality values to decrease. In this case, we can see that low β values make negotiations fail, due to the fact that both approaches are driven by aspiration values, and this parameter controls the amount of potential solutions which fall above the aspiration value of both agents. However, for high concentration of Pareto efficient solutions and high β values, the optimality of both approaches tend to decrease for low values of the correlation distance. This is because in this situation the ratio of good solutions is high, but the low correlation distance makes difficult for the mechanisms to search for the optimal in an efficient manner, so solutions finally accepted are usually suboptimal. In contrast, in highly correlated scenarios, this decrease of the optimality with beta is not observed. Finally, we can see that, while region-based negotiation works better than similarity for lowly correlated scenarios, SBNP outperforms RBNP in the highly correlated ones, due to the fact that the aforementioned monotonicity assumption holds. The results for lowly-epistatic scenarios are omitted, since they did not change the general trends observed in the highly-epistatic ones, besides rising optimality rates.

Table 1: Optimality rates in highly epistatic scenarios with $\rho_{Pareto} = 0.05$.

		β					
		0.25	0.5	1	2	4	
$\psi_{0.7}$	0.01	0.0000	0.0000	0.2150	0.3350	0.9050	SBNP
		0.0000	0.0000	0.5550	0.6900	0.7950	RBNP
		0.7207	0.7483	1.0000	0.9900	1.0000	CI
	0.05	0.0000	0.3150	0.3750	0.5700	0.8450	SBNP
		0.0000	0.4550	0.5300	0.7550	0.8400	RBNP
		0.2207	0.9650	1.0000	1.0000	1.0000	CI
	0.1	0.0000	0.1150	0.6200	0.7300	0.7150	SBNP
		0.0000	0.5650	0.5950	0.8200	0.8800	RBNP
		0.5586	0.9700	0.9750	1.0000	1.0000	CI
	0.5	0.0000	0.1700	0.5800	0.8300	0.9800	SBNP
		0.0000	0.4400	0.6450	0.8800	0.9550	RBNP
		0.6000	1.0000	0.9800	1.0000	1.0000	CI

Table 2: Optimality rates in highly epistatic scenarios with $\rho_{Pareto} = 0.2$.

		β					
		0.25	0.5	1	2	4	
$\psi_{0.7}$	0.01	0.8280	0.9360	0.9310	0.0950	0.5250	SBNP
		0.8200	0.9200	0.8862	0.5100	0.6850	RBNP
		0.9880	0.9840	1.0000	1.0000	1.0000	CI
	0.5	0.8400	0.7655	0.9862	0.1600	0.5700	SBNP
		0.7120	0.8310	0.9241	0.5100	0.6550	RBNP
		0.9600	0.9724	1.0000	1.0000	1.0000	CI
	0.5	0.9040	0.8966	0.2000	0.3950	0.7350	SBNP
		0.8000	0.8379	0.5000	0.5500	0.6850	RBNP
		0.9840	0.9966	0.9550	1.0000	1.0000	CI
	0.5	0.8800	0.9207	0.2250	0.6289	0.6750	SBNP
		0.9120	0.7655	0.4100	0.6477	0.6150	RBNP
		0.9960	1.0000	1.0000	1.0000	1.0000	CI

5. CONCLUSIONS AND FUTURE WORK

One of the main problems in complex automated negotiation research is the difficulty to compare approximations from different authors, due to the vast diversity of scenarios considered by the different research groups working in this field. In this paper we present a framework for the characterization and generation of negotiation scenarios, with the aim to fill this gap. First, we provide a set of metrics to measure high-level scenario parameters, taking into account both the structural properties of the agent utility functions, and the complexity due to the relationships between the utility functions of the different agents. Then, we present a framework to generate scenarios in a parametric and reproducible way. The generator is based on the aggregation of hypervolumes to generate utility functions, and on the use of shared hypervolumes and nonlinear regression to generate negotiation scenarios from utility diagrams.

Though the experiments performed with the scenario generator yield satisfactory results, there is still plenty of research to be done in this area. We are interested in exploring new metrics, like smoothness or neutrality, and to refine the control of the generator over the current ones (e.g. fine-grined control of epistasis). We are interested in creating templates for the generation of the most usual scenarios in the literature, and in performing an exhaustive comparison of the most relevant related works in all those scenarios. Finally, we are working on a community website where generated scenarios may be stored and searched for according to their parameters, and where users of the framework can contribute to its ongoing development both with scenarios to add to the library and with extensions to the framework code.

6. ACKNOWLEDGMENTS

This work has been developed in the framework of the ITEA-2 project 2008005, "Do-it-Yourself Smart Experiences", and supported by the Spanish Ministry of Education and Science grant TIN2008-06739-C04-04 (project T2C2).

7. REFERENCES

[1] R. Buttner. A classification structure for automated negotiations. In *WI-IATW '06: Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, pages 523–530, Washington, DC, USA, 2006. IEEE Computer Society.

[2] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:2006, 2006.

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

[4] S. Fatima, M. Wooldridge, and N. R. Jennings. An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 1041–1048, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[5] K. Fujita, T. Ito, and M. Klein. An approach to scalable multi-issue negotiation: Decomposing the contract space based on issue interdependencies. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 2, pages 399 –406, 2010.

[6] K. Fujita, T. Ito, and M. Klein. Common testbed generating tool based on xml for multiple interdependent issues negotiation problems. In T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo, and H. Yamaki, editors, *Innovations in Agent-Based Complex Automated Negotiations*, volume 319 of *Studies in Computational Intelligence*, pages 89–105. Springer Berlin / Heidelberg, 2011.

[7] M. Grabisch. k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets Syst.*, 92(2):167–189, 1997.

[8] K. V. Hindriks, C. M. Jonker, and D. Tykhonov. A multi-agent environment for negotiation. In A. El Fallah Seghrouchni, J. Dix, M. Dastani, and R. H. Bordini, editors, *Multi-Agent Programming:*, pages 333–363. Springer US, 2009.

[9] T. Ito, M. Klein, and H. Hattori. A multi-issue negotiation protocol among agents with nonlinear utility functions. *Journal of Multiagent and Grid Systems*, 4(1):67–83, 2008.

[10] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: Prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

[11] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Protocols for negotiating complex contracts. *IEEE Intelligent Systems*, 18(6):32–38, 2003.

[12] S. Kraus, K. Sycara, and A. Evenchick. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 1-2:1–69, 1998.

[13] G. Lai, C. Li, K. Sycara, and J. Giampapa. Literature review on multiattribute negotiations. Technical Report CMU-RI-TR-04-66, Robotics Institute, Carnegie Mellon University, Pittsburgh, USA, December 2004.

[14] G. Lai and K. Sycara. A generic framework for automated multi-attribute negotiation. *Group Decision and Negotiation*, 18:169–187, 2009. 10.1007/s10726-008-9119-9.

[15] R. Lin, S. Kraus, D. Tykhonov, K. Hindriks, and C. Jonker. Supporting the design of general automated negotiators. In T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo, and H. Yamaki, editors, *Innovations in Agent-Based Complex Automated Negotiations*, volume 319 of *Studies in Computational Intelligence*, pages 69–87. Springer Berlin / Heidelberg, 2011.

[16] M. A. Lopez-Carmona, I. Marsa-Maestre, E. de la Hoz, and J. R. Velasco. A region-based multi-issue negotiation protocol for non-monotonic utility spaces. *Computational Intelligence*, 2011.

[17] B. Manderick, M. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 1143–150, San Diego, CA, 1991.

[18] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, New York, USA, 1995.

[19] V. Robu, D. J. A. Somefun, and J. A. La Poutré. Modeling complex multi-issue negotiations using utility graphs. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 280–287, New York, NY, USA, 2005. ACM.

[20] V. K. Vassilev, T. C. Fogarty, and J. F. Miller. *Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application*, pages 3–44. Springer-Verlag New York, Inc., New York, NY, USA, 2003.

[21] M. Ventresca and B. Ombuki-Berman. Epistasis in multi-objective evolutionary recurrent neuro-controllers. In *Artificial Life, 2007. ALIFE '07. IEEE Symposium on*, pages 77 –84, 2007.

[22] J. Zhang and P. Pu. Survey on solving multi-attribute decision problems. Technical report, EPFL, 2004.

The Effect of Grouping Issues in Multiple Interdependent Issues Negotiation between Exaggerator Agents

Katsuhide Fujita*
Center for Collective Intelligence, Sloan School of Management, MIT.
Cambridge, USA
fujita@itolab.mta.nitech.ac.jp

Takayuki Ito
Techno-Business School, Nagoya Institute of Technology, Nagoya, Japan
ito.takayuki@nitech.ac.jp

Mark Klein
Center for Collective Intelligence, Sloan School of Management, MIT.
Cambridge, USA
m_klein@mit.edu

ABSTRACT

Most real-world negotiation involves multiple interdependent issues, which makes an agent's utility functions complex. Traditional negotiation mechanisms, which were designed for linear utilities, do not fare well in nonlinear contexts. One of the main challenges in developing effective nonlinear negotiation protocols is scalability; it can be extremely difficult to find high-quality solutions when there are many issues, due to computational intractability. One reasonable approach to reducing computational cost, while maintaining good quality outcomes, is to decompose the contract space into several largely independent sub-spaces. In this paper, we propose a method for decomposing a contract space into sub-spaces based on the agent's utility functions. A mediator finds sub-contracts in each sub-space based on votes from the agents, and combines the sub-contracts to produce the final agreement. We demonstrate, experimentally, that our protocol allows high-optimality outcomes with greater scalability than previous efforts.

Any voting scheme introduces the potential for strategic non-truthful voting by the agents, and our method is no exception. For example, one of the agents may always vote truthfully, while the other exaggerates so that its votes are always "strong." It has been shown that this biases the negotiation outcomes to favor the exaggerator, at the cost of reduced social welfare. We employ the limitation of strong votes to the method of decomposing the contract space into several largely independent sub-spaces. We investigate whether and how this approach can be applied to the method of decomposing a contract space.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - Multi-agent System

General Terms

Algorithms, Design, Experimentation

Keywords

Multi-Issue Negotiation, Interdependent Issues, Multi-agent System

*Visiting from Nagoya Institute of Technology, Japan.

1. INTRODUCTION

Negotiation is an important aspect of daily life and represents an important topic in the field of multi-agent system research. There has been extensive work in the area of automated negotiation; that is, where automated agents negotiate with other agents in such contexts as e-commerce[13], large-scale deliberation[20], collaborative design, and so on. Many real-world negotiations are complex and involve interdependent issues. When designers work together to design a car, for example, the utility of a given carburetor choice is highly dependent on which engine is chosen. The key impact of such issue dependencies is that they create qualitatively more complex utility functions, with multiple optima. There has been an increasing interest in negotiation with multiple interdependent issues. [9, 17, 21, 22, 24]. To date, however, achieving high scalability in negotiations with multiple interdependent issues remains an open problem.

We propose a new protocol in which a mediator tries to reorganize a highly complex utility space with issue interdependencies into several tractable subspaces, in order to reduce the computational cost. We call these utility subspaces "Issue groups." First, the agents generate interdependency graphs which capture the relationships between the issues in their individual utility functions, and derive issue clusters from that. While others have discussed issue interdependency in utility theory[26, 2], these efforts weren't aimed at efficiently decomposing the contract space. Second, the mediator combines these issue clusters to identify aggregate issue groups. Finally, the mediator uses a nonlinear optimization protocol to find sub-agreements for each issue group based on votes from the agents, and combines them to produce the final agreement.

We also address a negotiation between Exaggerator Agents. Any voting scheme introduces the potential for strategic non-truthful voting by the agents, and our method is no exception. For example, one of the agents may always vote truthfully, while the other exaggerates so that its votes are always "strong." It has been shown that this biases the negotiation outcomes to favor the exaggerator, at the cost of reduced social welfare. We employ the limitation of strong votes to the issue-grouping method. We investigate whether this approach can be applied to the method of decomposing a contract space.

The remainder of this paper is organized as follows. We describe a model of multiple interdependent issues negotiation and the strength of interdependency between issues, and the structure of interdependency graph. Next, we present a clustering technique for finding issue sub-groups. We then propose a protocol that uses this issue group information to enable more scalable negotiations. We also describe the effect of Exaggerator Agents in multi-agent situations. We present the experimental results, demonstrating that our protocol produces more optimal outcomes than previous efforts. Finally, we describe related work and present our overall conclusions.

2. NEGOTIATION WITH NONLINEAR UTILITY FUNCTIONS

2.1 Multi-issue Negotiation Model

We consider the situation where N agents $(a_1, \ldots, a_N)$ want to reach an agreement with a mediator who manages the negotiation from a man-in-the-middle position. There are M issues $(i_1, \ldots, i_M)$ to be negotiated. The number of issues represents the number of dimensions in the utility space. The issues are shared: all agents are potentially interested in the values for all M issues. A contract is represented by a vector of values $\vec{s} = (s_1, ..., s_M)$. Each issue s_j has a value drawn from the domain of integers $[0, X]$, *i.e.*, $s_j \in \{0, 1, , \ldots, X\}(1 \le j \le M)$. [1].

An agent's utility function, in our formulation, is described in terms of constraints. There are l constraints, $c_k \in C$. Each constraint represents a volume in the contract space with one or more dimensions and an associated utility value. c_k has value $w_a(c_k, \vec{s})$ if and only if it is satisfied by contract $\vec{s}$. Function $\delta_a(c_k, i_j)$ is a region of i_j in c_k, and $\delta_a(c_k, i_j)$ is $\emptyset$ if c_k doesn't have any relationship to i_j. Every agent has its own, typically unique, set of constraints.

An agent's utility for contract $\vec{s}$ is defined as the sum of the utility for all the constraints the contract satisfies, i.e., as $u_a(\vec{s}) = \sum_{c_k \in C, \vec{s} \in x(c_k)} w_a(c_k, \vec{s})$, where $x(c_k)$ is a set of possible contracts (solutions) of c_k. This formulation produces complex utility functions with high points where many constraints are satisfied and lower regions where few or no constraints are satisfied. Many real-world utility functions are quite complex in this way, involving many issues as well as higher-order (e.g. trinary and quaternary) constraints. This represents a crucial departure from most previous efforts on multi-issue negotiation, where contract utility has been calculated as the weighted sum of the utilities for individual issues, producing utility functions shaped like hyper-planes, with a single optimum.

Figure 1 shows an example of a utility space generated via a collection of binary constraints involving Issues 1 and 2. In addition, the number of terms is two. The example, which has a value of 55, holds if the value for Issue 1 is in the range $[3, 7]$ and the value for Issue 2 is in the range $[4, 6]$. The utility function is highly nonlinear with many hills and valleys. This constraint-based utility function representation allows us to capture the issue interdependencies common in real-world negotiations. The constraint in Figure 1, for example, captures the fact that a value of 4 is desirable for issue 1 if issue 2 has the value 4, 5 or 6. Note, however, that this representation is also capable of capturing linear utility functions as a special case (they can be captured as a series of unary constraints). A negotiation protocol for complex contracts can, therefore, handle linear contract negotiations.

[1]A discrete domain can come arbitrarily close to a 'real' domain by increasing its size. As a practical matter, many real-world issues that are theoretically 'real' numbers (delivery date, cost) are discretized during negotiations.

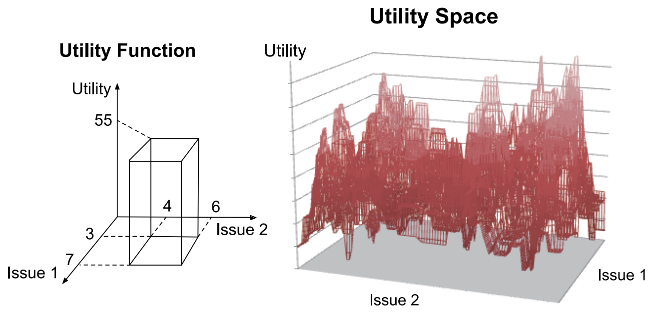


Figure 1: Example of a nonlinear utility space

This formulation was described in [9]. In [17, 21, 22], a similar formulation is presented that supports a wider range of constraint types.

The objective function for our protocol can be described as follows:

$$\arg\max_{\vec{s}} \sum_{a \in N} u_a(\vec{s}). \tag{1}$$

$$\arg\max_{\vec{s}} u_a(\vec{s}), \ (a = 1, \ldots, N). \tag{2}$$

Our protocol, in other words, tries to find contracts that maximize social welfare, *i.e.*, the summed utilities for all agents. Such contracts, by definition, will also be Pareto-optimal. At the same time, all the agent try to find contracts that maximize their own welfare.

3. OUR NEGOTIATION PROTOCOL: DECOMPOSING THE CONTRACT SPACE

It is of course theoretically possible to gather all of the individual agents' utility functions in one central place and then find all optimal contracts using such well-known nonlinear optimization techniques as simulated annealing or evolutionary algorithms. However, we do not employ such centralized methods for negotiation purposes because we assume, as is common in negotiation contexts, that agents prefer not to share their utility functions with each other, in order to preserve a competitive edge.

Our approach is described in the following sections.

3.1 Analyzing issue interdependency

The first step is for each agent to generate an interdependency graph by analyzing the issue interdependencies in its

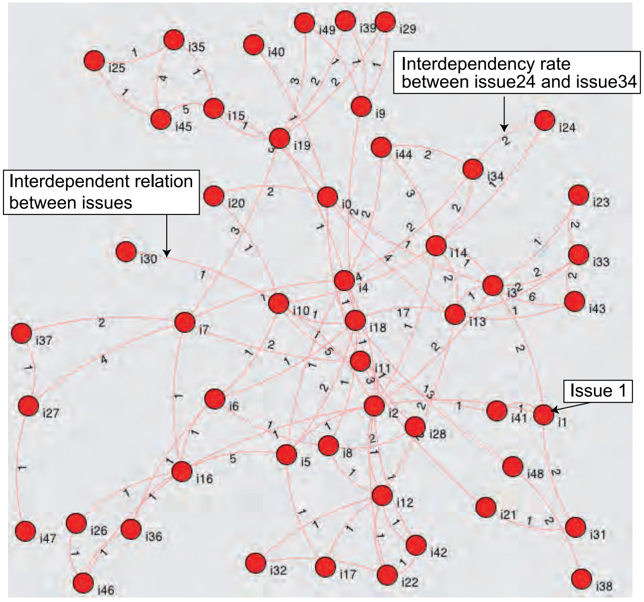


Figure 2: Interdependency Graph (50 issues)

own utility space. We define issue interdependency as follows. If there is a constraint between issue X (i_X) and issue Y (i_Y), then we assume i_X and i_Y are interdependent. If, for example, an agent has a binary constraint between issue 1 and issue 3, those issues are interdependent for that agent.

The *strength* of an issue interdependency is captured by the interdependency rate. We define the interdependency rate between two issues as the number of constraints that interrelate them. The interdependency rate between *issue* i_j and *issue* i_{jj} for agent a is thus $D_a(i_j, i_{jj}) = \sharp\{c_k | \delta_a(c_k, i_j) \neq \emptyset \wedge \delta_a(c_k, i_{jj}) \neq \emptyset\}$.

Agents capture their issue interdependency information in the form of interdependency graphs i.e. weighted non-directed graphs where a node represents an issue, an edge represents the interdependency between issues, and the weight of an edge represents the interdependency rate between those issues. An interdependency graph is thus formally defined as: $G(P, E, w) : P = \{1, 2, \ldots, |I|\}(finite\ set), E \subset \{\{x, y\}|x, y \in P\}, w : E \rightarrow R$.

Figure 2 shows an example of an interdependency graph.

3.2 Grouping issues

In this step, the mediator employs breadth-first search to combine the issue clusters submitted by each agent into a consolidated set of issue groups. For example, if agent 1 submits the clusters $\{i_1, i_2\}$, $\{i_3, i_4, i_5\}$, $\{i_0, i_6\}$ and agent 2 submits the clusters $\{i_1, i_2, i_6\}$, $\{i_3, i_4\}$, $\{i_0\}$, $\{i_5\}$, the mediator combines them to produce the issue groups $\{i_0, i_1, i_2, i_6\}$, $\{i_3, i_4, i_5\}$. In the worst case, if all the issue clusters submitted by the agents have overlapping issues, the mediator generates the union of the clusters from all the agents. The details of this algorithm are given in Algorithm1.

It is possible to gather all of the agents' interdependency graphs in one central place and then find the issue groups using standard clustering techniques. However, it is hard to determine the optimal number of issue groups or the clustering parameters in central clustering algorithms, because the basis of clustering for every agent can be different. Our approach avoids these weaknesses by requiring that each agent generates its own issue clusters. In our experiments, agents did so using the well-known Girvan-Newman algorithm[18], which computes clusters in weighted non-direct graphs. The algorithm's output can be controlled by changing the "number of edges to remove" parameter. Increasing the value of this parameter increases the number of issue dependencies ignored when calculating the issue clusters, thereby resulting in a larger number of smaller clusters. The running time of this algorithm is $O(kmn)$, where k is the number of edges to remove, m is the total number of edges, and n is the total number of vertices.

Algorithm 1 Combine_IssueGroups(G)

Ag: A set of agents, G: A set of issue-groups of each agent ($G = \{G_0, G_1, ..., G_n\}$, a set of issue-groups from agent i is $G_i = \{g_{i,0}, g_{i,1}, ..., g_{i,m_i}\}$)

$SG := G_0$, $i := 1$
while $i < |Ag|$ **do**
    $SG' := \emptyset$
    for $s \in SG$ **do**
        for $g_{i,j} \in G_i$ **do**
            $s' := s \cap g_{i,j}$
            if $s' \neq \phi$ **then**
                $SG' := s \cup g_{i,j}$
            end if
            $SG := SG'$, $i := i + 1$
        end for
    end for
end while

3.3 Finding Agreements

We use a distributed variant of simulated annealing (SA)[11] to find optimal contracts in each issue group. In each round, the mediator proposes a contract that is a random single-issue mutation of the most recently accepted contract (the accepted contract is initially generated randomly). Each agent then votes to accept(+2), weakly accept(+1), weakly reject(-1) or reject(-2) the new contract, based on whether it is better or worse than the last accepted contract for that issue group. When the mediator receives these votes, it adds them together. If the sum of the vote values from the agents is positive or zero, the proposed contract becomes the currently accepted one for that issue group. If the vote sum is negative, the mediator will accept the contract with probability $P(accept) = e^{\Delta U/T}$, where T is the mediator's virtual temperature (which declines over time) and ΔU is the utility change between the contracts. In other words, the higher the virtual temperature, and the smaller the utility decrement, the greater the probability that the inferior contract will be accepted. If the proposed contract is not accepted, a mutation of the most recently accepted contract is proposed in the next round. This continues over many rounds. This technique allows the mediator to skip past local optima in the utility functions, especially earlier on in the search process, in the pursuit of global optima.

Algorithm 2 Simulated_Annealing()

$Value(N)$: the sum of the numeric values mapped from votes to N from all agents

```
S := initial solution (set randomly)
for t = 1 to ∞ do
  T := schedule(t)
  if T = 0 then
    return current
  end if
  next := a randomly selected successor of current
  if next.Value ≥ 0 then
    ΔE := next.Value − current.Value
    if ΔE > 0 then
      current := next
    else
      current :=next only with probability e^{ΔE/T}
    end if
  end if
end for
```

3.4 Exaggerator Agents

Any voting scheme introduces the potential for strategic non-truthful voting by the agents, and our method is no exception. For example, one of the agents may always vote truthfully, while the other exaggerates so that its votes are always "strong." It has been shown that this biases the negotiation outcomes to favor the exaggerator, at the cost of reduced social welfare [12]. What we need is an enhancement of our negotiation protocol that preventing the exaggerator votes and maximizing social welfare.

We guess that simply placing a limit on the number of "strong" votes each agent can work well. If the limit is too low, we effectively lose the benefit of vote weight information and get the lower social welfare values that result. If the strong vote limit is high enough to avoid this, then all an exaggerator has to do is save all of its strong votes until the end of the negotiation, at which point it can drag the mediator towards making a series of proposals that are inequitably favorable to it. In the experiments, we demonstrate that the limit of the number of "strong" voting is efficient of finding high solutions.

4. EXPERIMENTAL RESULTS

4.1 Setting

We conducted several experiments to evaluate our approach. In each experiment, we ran 100 negotiations. The following parameters were used. The domain for the issue values was $[0, 9]$. Each agent had 10 unary constraints, 5 binary constraints, 5 trinary constraints, and so on. (a unary constraint relates to one issue, a binary constraint relates to two issues, etc). The maximum weight for a constraint was *100* $\times$ *(Number of Issues)*.

In our experiments, each agents' issues were organized into ten small clusters with strong dependencies between the issues within each cluster. We ran two conditions: "1) Sparse Connection" and "2) Dense Connection". Figure 3 gives examples, for these two cases, of interdependency graphs and the relationship between the number of issues and the sum of the connection weights between issues. As these graphs

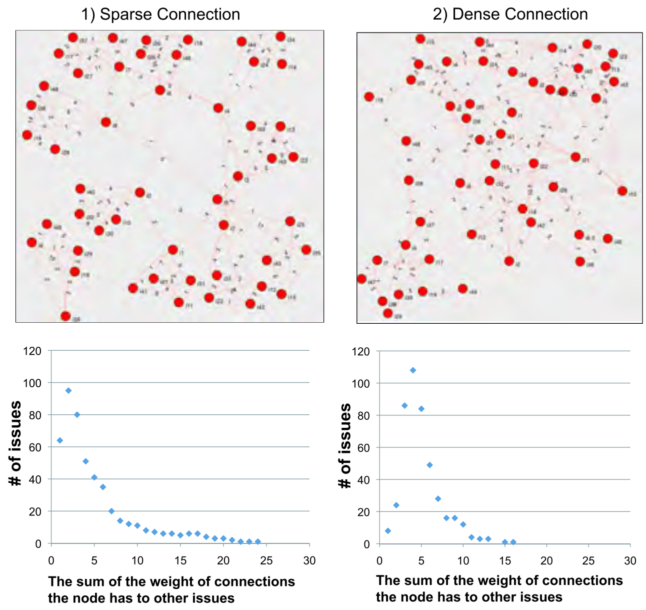


Figure 3: Issue Interdependencies

show, the "1) Sparse Connection" case is closer to a scale-free distribution, with power-law statistics, while the "2) Dense connection" is closer to a random graph.

We compared the following negotiation methods:

"(A) Issue-Grouping (True Voting)" applies the simulated annealing protocol based on the agents' votes, and performs the negotiation separately for each one of the issue groups, and combines the resulting sub-agreements to produce the final agreement. All agents tell the truth votes. "(B) Issue-Grouping (Exaggerator Agents)" applies the simulated annealing protocol based on the agents' votes with issue-grouping. "All agent" tell the exaggerator votes. "(C) Issue-Grouping (limitation)" is same situation with (B). However, the limitation of 'strong' votes is applied. The number of limitation of 'strong' votes is 250 which is the optimal number of limitations in this experiments. "(D) Without Issue-Grouping" is the method presented in Klein et.al[12], using a simulated annealing protocol based on the agents' votes without generating issue-groups.

In all these cases, the search began with a randomly generated contract, and the SA initial temperature for all these cases was 50.0 and decreased linearly to 0 over the course the negotiation. In case (D), the search process involved 500 iterations. In case (A)-(C), the search process involved 50 iterations for each issue group. Cases (A),(B),(C) and (D) thus used the same amount of computation time, and are thus directly comparable. The number of edges removed from the issue interdependency graph, when the agents were calculating their issue groups, was 6 in all cases.

We applied a centralized simulated annealing to the sum of the individual agents' utility functions to approximate the optimal social welfare for each negotiation test run. Exhaustive search was not a viable option because it becomes

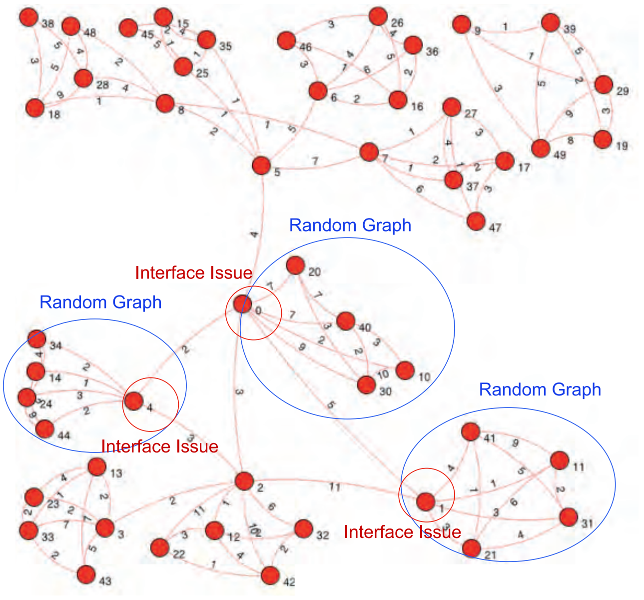


Figure 4: Method of determining interdependency graph

computationally intractable as the number of issues grows. The SA initial temperature was 50.0 and decreases linearly to 0 over the course of 2,500 iterations. The initial contract for each SA run is randomly selected. We calculated a normalized "optimality rate" for each negotiation run, defined as *(social welfare achieved by each protocol) / (optimal social welfare calculated by SA)*.

Our code was implemented in Java 2 (1.6) and was run on a core 2-duo CPU with 2.0 GB memory under Mac OS X (10.6).

4.2 Method of determining interdependency graph

Figure 4 shows what the interdependency graph consists of in an agent.

The method of determining the interdependency between issues in the experiment is as follows.

(Step 1) Small issue-groups are generated by connecting a part of the issues randomly.

(Step 2) The interface issues are decided randomly among issues in each issue-group. The interface issues are for connecting other small issue-groups. In small issue-groups, only the interface issues can connect to other issue-groups.

(Step 3) Each issue-group connects to other small issue-groups. Specifically, all combinations of each issue-group are searched for, and it is decided whether connection or disconnection according to the possibility of generating connections.

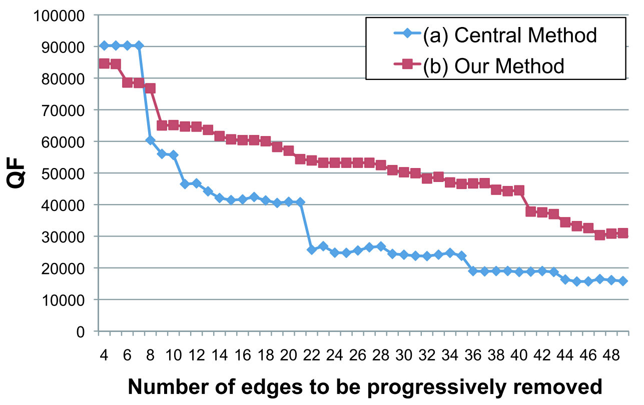


Figure 7: Number of edges to be progressively removed (Clustering parameter) v.s. QF

4.3 Experimental Results

Figure 5 and 6 compare the optimality rate in the sparse connection and dense connection cases. "(A) Issue-Grouping (True Voting)" achieved a higher optimality rate than "(D) Without Issue-Grouping" which means that the issue-grouping method produces better results for the same amount of computational effort. The optimality rate of the "(A) Issue-Grouping (True Voting)" condition decreased as the number of issues (and therefore the size of the search space) increased. "(B) Issue-Grouping (Exaggerator Agents)" is worse than "(A) Issue-Grouping (True Voting)" because the exaggerator agents generate reduced social welfares in multi-agents situations. However, "(C) Issue-Grouping (limitation)" outperforms "(B) Issue-Grouping (Exaggerator Agents)", therefore, the limitation of 'strong' votes is effective of improving the social welfare reduced by the Exaggerator Agents.

The optimality rates for all methods are almost unaffected by the number of agents, as Figure 6 shows. The optimality rate for (A) is higher than (D) in the "1) Sparse Connections" case than the "2) Dense Connections" case. This is because the issue grouping method proposed in this paper can achieve high optimality if the number of ignored interdependencies is low, which is more likely to be true in the "1) Sparse Connections" case. Many real-world negotiations are, we believe, characterized by sparse issue inter-dependencies.

We also assessed a quality factor measure *QF = (Sum of internal weights of edges in each issue-group) / (Sum of external weights of edges in each issue-group)* to assess the quality of the issue groups, i.e. the extent to which issue dependencies occurred only between issues in the same clusters, rather than between issues in different groups. Higher quality factors should, we predict, increase the advantage of the issue grouping protocols, because that means fewer dependencies are ignored when negotiation is done separately for each issue group. Figure 7 shows the quality factors when the number of agents is 3 and 20, as a function of the number of edges to be removed (which is the key parameter in the clustering algorithm we used).The number of issues is 50 in the "1) sparse connection" case. "(a) Central Method" is to gather all of the agents' interdependency graphs in one central place and then find the issue groups

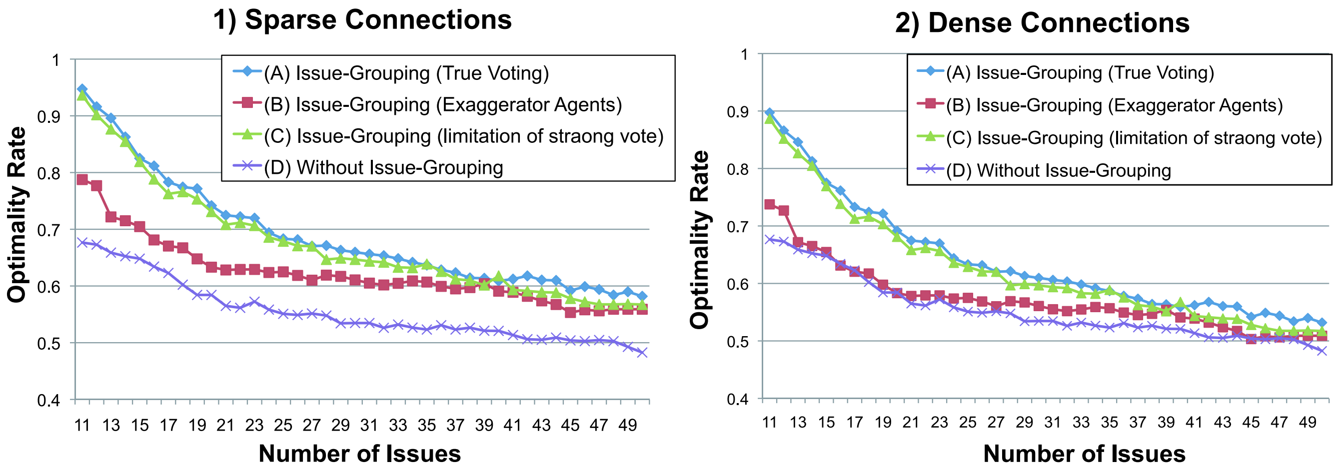


Figure 5: Comparison of optimality when the number of issues changes

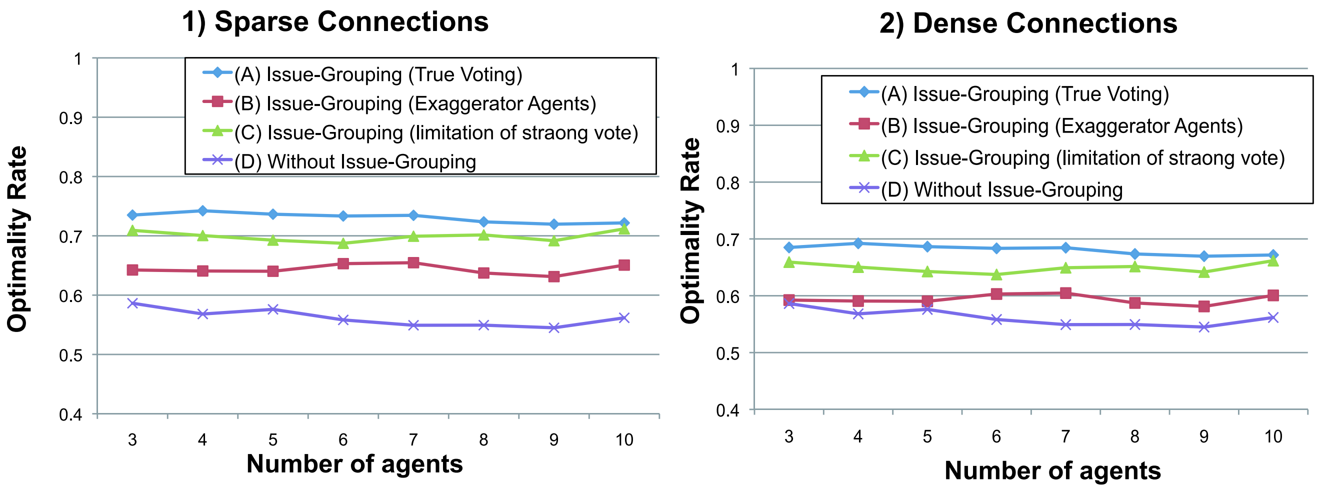


Figure 6: Comparison of optimality when the number of agents changes

using the well-known Girvan-Newman algorithm[18]. "(b) Our method" employs breadth-first search to combine the issue clusters submitted by each agent into a consolidated set of issue groups.

Comparing (a) with (b) in Figure 7, (b) proposed in this paper outperforms (a). This is because that our method is reflected by the idea of all agents to final issue-grouping without fixing the clustering parameter as Figure8 showing. QF becomes smaller when the number of edges to be progressively removed is larger. This is because the number of issue-groups generated by each agent is higher as the number of edges to be progressively removed becomes larger. The rapid decrease sometimes happens as the number of edges to be progressively removed increases. These points are good parameters for decomposing the issue-groups. In real life, the utility of agents contains an adequate idea of issue-groups, and agents can determine the optimal idea of issue-groups by analyzing the utility spaces.

5. RELATED WORK

Even though negotiation seems to involve a straightforward distributed constraint optimization problem [7, 19], we have been unable to exploit existing work on high-efficiency constraint optimizers. Such solvers attempt to find the solutions that maximize the weights of the satisfied constraints, but do not account for the fact that the final solution must satisfy at least one constraint *from every agent*.

Lin et al.[16] explored a range of protocols based on mutation and selection on binary contracts. This paper does not describe what kind of utility function is used, nor does it present any experimental analyses, so it remains unclear whether this strategy enables sufficient exploration of utility space.

Klein et al.[12] presented a protocol applied with near optimal results to medium-sized bilateral negotiations with binary dependencies, but was not applied to multilateral negotiations and higher order dependencies.

A bidding-based protocol was proposed by Ito et al.[9]. Agents

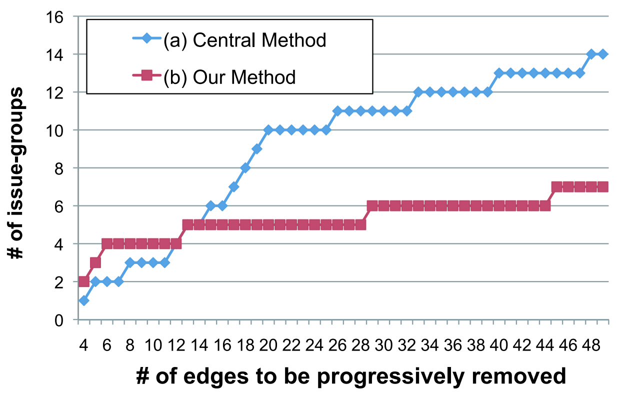


Figure 8: Number of edges to be progressively removed (Clustering parameter) v.s. The number of issue-groups

generate bids by finding high regions in their own utility functions, and the mediator finds the optimum combination of submitted bids from the agents. However, the scalability of this protocol is limited, and the failure rate of making agreements is too high. By Fujita et al.[5], a representative-based protocol for reducing the computational cost was proposed based on the bidding-based protocol. In this method, the scalability of agents was improved; however, the scalability of issues was not sufficient. Fujita et.al[6] also focused on the decomposing the contract space for highly scalable negotiation, but the negotiation protocol and experimental results are completely different.

Hindriks et al.[8] proposed an approach based on a weighted approximation technique to simplify the utility space. The resulting approximated utility function without dependencies can be handled by negotiation algorithms that can efficiently deal with independent multiple issues, and has a polynomial time complexity. Our protocol can find an optimal agreement point if agents don't have in common the expected negotiation outcome.

Fatima et al.[3, 4] proposed bilateral multi-issue negotiations with time constraints. This method can find approximate equilibrium in polynomial time where the utility function is nonlinear. However, this paper focused on bilateral multi-issue negotiations. Our protocol focuses on multilateral negotiations.

Zhang[27] presents an axiomatic analysis of negotiation problems within task-oriented domains (TOD). In this paper, three classical bargaining solutions (Nash solution, Egalitarian solution, Kalai-Smorodinsky solution) coincide when they are applied to a TOD with mixed deals but diverge if their outcomes are restricted to pure deals.

Maestre et al.[21, 22, 23] proposed an auction-based protocol for nonlinear utility spaces generated using weighted constraints, and proposed a set of decision mechanisms for the bidding and deal identification steps of the protocol. They proposed the use of a quality factor to balance utility and deal probability in the negotiation process. This quality factor is used to bias bid generation and deal identification, taking into account the agents' attitudes toward risk. The scalability of the number of issues is still a problem in these works.

Jonker et al.[10] proposed a negotiation model called ABMP that can be characterized as cooperative one-to-one multi-criteria negotiation in which the privacy of both parties is protected as much as desired.

By Robu et al.[24], utility graphs were used to model issue dependencies for binary-valued issues. Our utility model is more general.

Bo et al.[1] proposed the design and implementation of a negotiation mechanism for dynamic resource allocation problem in cloud computing. Multiple buyers and sellers are allowed to negotiate with each other concurrently and an agent is allowed to decommitment from an agreement at the cost of paying a penalty.

Lin et al. [14, 15] focus on the Expert Designed Negotiators (EDN) which is the negotiations between humans and automated agents in real-life. In addition, the tools for evaluating automatic agents that negotiate with people were proposed. These studies include some efficient results from extensive experiments involving many human subjects and PDAs.

6. CONCLUSION

In this paper, we proposed a new negotiation protocol, based on grouping issues, which can find high-quality agreements in interdependent issue negotiation. In this protocol, agents generate their private issue interdependency graphs and use these to generate issue clusters. The mediator consolidates these clusters to define aggregate issue groups, and independent negotiations proceed for each group. We analyzed the negotiation that one of agents may always vote truthfully, while the other exaggerates so that its votes are always "strong." We demonstrated that our proposed protocol results in a higher optimality rate than methods that don't use issue grouping, especially when the issue interdependencies are relatively sparse. In addition, the limitation of "strong" votes is effective of improving the reduced social welfare in multi-agent negotiations between exaggerators.

In future work, we will conduct additional negotiation, after the concurrent sub-contract negotiations, to try to increase the satisfaction of constraints that crossed issue group boundaries and were thus ignored in our issue grouping approach. In the bilateral case, we found this can be done using a kind of Clarke tax [25], wherein each agent has a limited budget from which it has to pay other agents before the mediator will accept a contract that favors that agent but reduces utility for the others. We investigate whether and how this approach can be applied to the multilateral case.

Acknowledgement

This research is partially supported by PREST, JST.

7. REFERENCES

[1] B. An, V. R. Lesser, D. Irwin, and M. Zink. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In *Proc. of the Ninth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2010)*, pages 981–988, 2010.

[2] S. D. Eppinger. Model-based approaches to managing concurrent engineering. *Journal of Engineering Design*, 2(4):283–190, 1991.

[3] S. S. Fatima, M. Wooldridge, and N. R. Jennings. An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In *Proc. of the Eighth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2009)*, pages 1041–1048, 2007.

[4] S. S. Fatima, M. Wooldridge, and N. R. Jennings. Approximate and online multi-issue negotiation. In *Proc. of the Sixth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2007)*, pages 947–954, 2007.

[5] K. Fujita, T. Ito, and M. Klein. A representative-based multi-round protocol for multi-issue negotiations. In *Proc. of the Seventh Inernational Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2008)*, pages 1573–1576, 2008.

[6] K. Fujita, T. Ito, and M. Klein. An approach to scalable multi-issue negotiation: Decomposing the contract space based on issue interdependencies. In *Proc. of the 2010 International Joint Conference on Intelligent Agent Technology (IAT-2010)*, 2010.

[7] R. Greenstadt, J. Pearce, and M. Tambe. Analysis of privacy loss in distributed constraint optimization. In *Proc. of the 21th Association for the Advancement of Artificial Intelligence (AAAI-2006)*, pages 647–653, 2006.

[8] K. Hindriks, C. Jonker, and D. Tykhonov. Eliminating interdependencies between issues for multi-issue negotiation. *Cooperative Information Agents X, Lecture Notes in Computer Science*, 4149:301–316, 2006.

[9] T. Ito, H. Hattori, and M. Klein. Multi-issue negotiation protocol for agents : Exploring nonlinear utility spaces. In *Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pages 1347–1352, 2007.

[10] C. M. Jonker, V. Robu, and J. Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 15:221–252, 2007.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[12] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation*, 12(2):58–73, 2003.

[13] S. Kraus. *Strategic Negotiation in Multiagent Environments.* Cambridge University Press, 2001.

[14] R. Lin and S. Kraus. Can automated agents proficiently negotiate with humans? *Commun. ACM*, 53(1):78–88, 2010.

[15] R. Lin, S. Kraus, Y. Oshrat, and Y. K. Gal. Facilitating the evaluation of automated negotiators using peer designed agents. In *Proc. of The 24th Association for the Advancement of Artificial Intelligence (AAAI-2010)*, 2010.

[16] R. J. Lin and S. T. Chou. Bilateral multi-issue negotiations in a dynamic environment. In *Proc. of the AAMAS Workshop on Agent Mediated Electronic Commerce (AMEC-2003)*, 2003.

[17] M. Lopez-Carmona, I. Marsa-Maestre, M. Klein, and T. Ito. Addressing stability issues in mediated complex contract negotiations for constraint-based, non-monotonic utility spaces. *Autonomous Agents and Multi-Agent Systems*, pages 1–51, 2010.

[18] G. M. and N. M. E. J. Community structure in social and biological networks. *National Academy of Sciences of USA (PNAS)*, 99(12):7821–7826, 2002.

[19] R. T. Maheswaran, J. P. Pearce, P. Varakantham, and E. Bowring. Valuations of possible states (vps):a quantitative framework for analysis of privacy loss among collaborative personal assistant agents. In *Proc. of the Forth Inernational Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2005)*, pages 1030–1037, 2005.

[20] T. W. Malone and M. Klein. Harnessing collective intelligence to address global climate change. *Innovations Journal*, 2(3):15–26, 2007.

[21] I. Marsa-Maestre, M. A. Lopez-Carmona, J. R. Velasco, and E. de la Hoz. Effective bidding and deal identification for negotiations in highly nonlinear scenarios. In *Proc. of the Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2009)*, pages 1057–1064, 2009.

[22] I. Marsá-Maestre, M. A. López-Carmona, J. R. Velasco, and E. de la Hoz. Avoiding the prisoner's dilemma in auction-based negotiations for highly rugged utility spaces. In *Proc. of the Ninth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-2010)*, pages 425–432, 2010.

[23] I. Marsa-Maestre, M. A. Lopez-Carmona, J. R. Velasco, T. Ito, M. Klein, and K. Fujita. Balancing utility and deal probability for negotiations in highly nonlinear utility spaces. In *Proc. of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-2009)*, pages 214–219, 2009.

[24] V. Robu, D. J. A. Somefun, and J. L. Poutre. Modeling complex multi-issue negotiations using utility graphs. In *Proc. of the Forth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005)*, pages 280–287, 2005.

[25] T. W. Sandholm. *Distributed Rational Decision Making.* Multi-Agent Systems. G. Weiss, Editor, 1998.

[26] H. Tamura and Y. Nakamura. Decompositions of multiattribute utility functions based on convex dependence. *Operations Research*, 31(3):488–506, 1983.

[27] D. Zhang. Axiomatic characterization of task oriented negotiation. In *Proc. of Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-2009)*, pages 367–372, 2009.

Efficient Deal Identification For the Constraints Based Utility Space Model

Raiye Hailu
School of Techno-Business Administration,Nagoya Institute of Technology.
raiye@itolab.mta.nitech.ac.jp

Takayuki Ito
School of Techno-Business Administration/ Dept. of Computer Science, Nagoya Institute of Technology.
Todai Policy Alternatives Research Institute, University of Tokyo.
PREST, Japan Science and Technology Agency (JST).
ito.takayuki@nitech.ac.jp

ABSTRACT
We propose correct and efficient algorithms for locating the optimal contract of negotiating agents that represent their utility space with the constraints based utility space model. It is argued that the agents that use the model can be classified in to two extreme kinds: sensitive and insensitive. When the negotiation is between a sensitive agent and many insensitive agents, the optimal contract can be computed correctly and efficiently by avoiding Exhaustive Matching.

General Terms
Automated Negotiations

Keywords
Utility models, Multi-Issue Negotiations

1. INTRODUCTION
Automating negotiations over multiple and interdependent issues is potentially an important line of research since most negotiations in the real world have interdependent issues. When a service provider negotiates on "When"to provide its service, its utility for a certain time period (e.g. T1=8a.m-10a.m) is dependent on the day of the week (Monday-Sunday). It might have high utility for T1 on Mondays, but low utility for T1 on Sundays. The issues, time of the meeting and day of the meeting cannot be negotiated independently.

We propose correct and efficient algorithm for locating the optimal contract of negotiating agents that represent their utility space with the constraints based utility space model proposed in [4]. The model is used to represent utility space of agents negotiating over multiple and interdependent issues. Some researchers [1, 2, 3, 5] have proposed algorithms(protocols) for locating the optimal contract. The proposed algorithms have their own merits, but they all fall under the classification of heuristic algorithms when evaluated solely from the view point of locating the optimal contract correctly. The optimal contract is the contract that has the maximum total utility. Total utility for a contract is the sum of the utility of each agent for the contract.

Exhaustively Matching (EM) the entire utility space of the agents is the only correct method of searching the optimal contract. If the utility space of agents is assumed to be generated randomly, then there is no method of making EM efficient (faster) and still guarantee correctness. Therefore we make intuitive assumptions about utility space of agents that can be readily implemented by the basic building block of the model - integer interval.

1.1 Constraints Based Utility Space Model
In the model, for agents negotiating on I number of issues, an I dimensional coordinate system is created. An axis is assigned to each issue. Each issue will have up to V number of issue values. We represent these values by integers ranging from 0 to V-1. Since the issues are interdependent, we will have V^I number of possible issue value combinations which are called contracts. An example of a contract is [0,2,4]. 0 is the issue value for I1(Issue 1) , 2 is the issue value for I2 etc.

The utility of a contract is the sum of the weights of all constraints satisfied at it. The constraint in Figure 1 has a weight of 55. Contracts that have the values 4 and 5 for issue 1, and the values 3, 4, 5 and 6 for issue 2 satisfy this constraint. An agent creates its utility space by defining multiple such constraints. Figure 2 shows a utility space created by using more than 100 constraints.

2. BIDDING BASED ALGORITHM
Most previous works that used the constraints based utility space model use the bidding based deal identification method. Bids are high utility regions of the utility space of an agent. In a nutshell, bidding is the process of identifying and then submitting these high utility regions to a mediator agent. The mediator agent matches the bids to find those that have intersections and maximize the total utility. It was first proposed in [4]. Since then, some researchers have

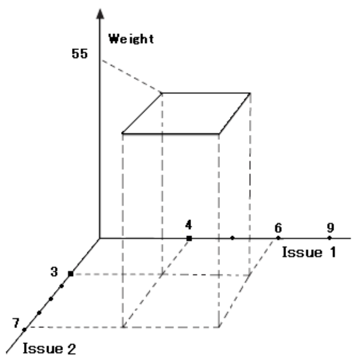


Figure 1: A 2 issue Constraint

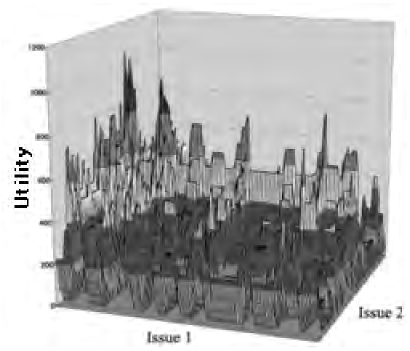


Figure 2: 2 issue utility space

improved the method to address various concerns.

The threshold adjusting algorithm [1] makes agents bid in multiple rounds rather than once. In each round the threshold value is lowered. The threshold value is the minimum allowable utility value of a bid. The bidding is stopped at the round a deal is found. This has the advantage of limiting the amount private information revealed to a third party.

The representative based algorithm [2] improves scalability of the bidding based algorithm by making only few agents called representatives participate in the bidding process. Scalability refers to the number agents that can be supported by the negotiation system. When the number of issues increases, the number of bids each agent has to make in order to effectively sample its utility space also increases. This in turn increases the time taken by the mediator to search an intersection of the bids that maximizes the total utility. If only the representatives are allowed to participate in the bidding process, then negotiations with large number of agents can be supported.

When the contract space is large, the failure rate (when no bids from agents intersect) of a negotiation increases. The iterative narrowing protocol [3] reduces failure rates by narrowing down the region of the contract space that the agents generate their bids from. It is especially effective when the constraints of each agent are found being clustered in some of regions of the contract space, rather than being scattered all over the contract space.

Measures that reduce high failure rates that arise when agents use narrow constraints was discussed in [5]. The product of a bid's utility and its volume was used as a criteria to select it to be submitted to the mediator or not. Usually high utility valued bids tend to be small in volume and therefore the chance that they will intersect with other agents'bids is minimal. Adding the volume criteria for selecting a bid for submission makes the deal identification process more effective.

The problem is that the bid that contains the optimal contract may not be submitted by at least one of the agents. This might be because either that bid has low utility for that agent, or the bid generation mechanism "missed"it. Hence, there is always the chance that the optimal contract is not found.

3. EXHAUSTIVE MATCHING

The only way we can guarantee that the optimal contract is computed correctly is by making the agents submit their entire utility space to the mediator. Then the mediator Exhaustively Matches(EM) the utility spaces. The problem is that the computational time cost of this algorithm grows exponentially. If the number of issues of a negotiation grows from I to $I+1$, then the contract space grows from V^I to $V^{\mathrm{I}+1}$.

To reduce the time required to search for the optimal contract, we have to look for patterns in the utility space of agents that could be exploited to avoid EM. But observing Figure 1 and Figure 2 reveals that based on the number of constraints, their positioning and weight, utility spaces can be of various types and very unpredictable. The only predictable nature of them is that they are all based on constraints. Not just any constraint but integer interval based constraints.

3.1 Single Issue Version of The Model

The constraint in Figure 1 is a two dimensional integer interval of [4..5]x[3..6]. An example of a constraint in a negotiation over three issues would be [2..5]x[1..3]x[6..9]. If we were to define a single issue version of the model , then an example of a constraint would be [1..3].

Since the single issue version is easy to understand we will use it for analysis and experiments from here on wards. Since integer intervals are the basic building unit of the model we expect lessons learned from studying the single issue version of the model will be applicable for the multi issue version of it.

Figure 3 shows an agent that has 3 constraints :(C1, C2, C3). Its utility for the issue value 5 is: Weight (C2) + Weight (C3) = 10+20 = 30. Figure 4 is Figure 3 redrawn by summing the weights of each constraint. S0, S1...,S6 are called Steps of the utility function. Notice that Steps are also integer intervals. Also notice that, in a one issue utility space the issue values themselves are contracts of the negotiation. For example, in Figure 4, Step 4(S4) contains the contracts 4 and 5.

To avoid EM, we have to make assumptions about utility space of agents. To do that we still focus on integer intervals. This time the Steps are considered.

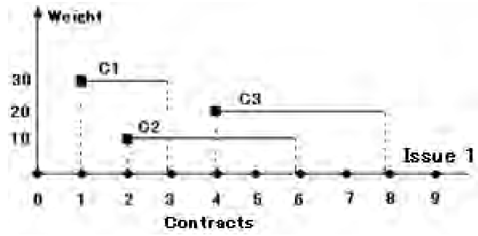


Figure 3: Many single issue constraints

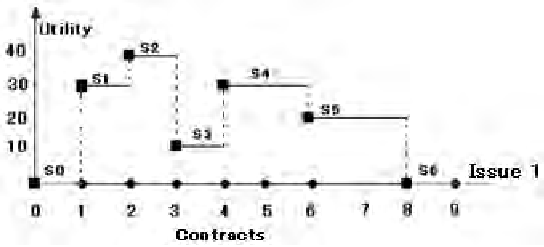


Figure 4: Single issue utility space

3.2 Sensitive and Insensitive Agents

By focusing on the width of the Steps in the utility space of an agent, we can ask some interesting questions. If an agent's utility space is dominated by Steps that are wide, what does that say about the agent? What about when an agent's utility space is dominated by Steps that are narrow?

A Step contains consecutive contracts that the agent has equal utility for. Let's assume that consecutive contracts are more similar to one other than contracts that are far apart. Then, the fact that the agent has equal utility for some consecutive contracts indicates that, the agent neglects the small difference between the contracts. Based on this, we can classify agents to two extreme kinds: sensitive and insensitive. Here, the word, sensitive is used as it would be used for a sensor. A sensitive sensor is capable of registering small differences of the sensed signal.

Let's define a branch to be a portion of the contract space . For example, part of the contract space in Figure 4 containing the contracts 0 to 3 ([0..3]). In a branch, a sensitive agent will have four Steps. One for each contract. An insensitive agent will have one Step that contains all the contracts.(Currently we assume that the end points of the branches of all agents are the same and known).

Consider negotiation for scheduling a meeting of 30 minutes duration. A busy person is sensitive about every 30 minute interval. While he is relatively free at 10:30 a.m., he might have very important meeting at 11:00 a.m.. Therefore, he would not like to have the meeting at 11:00 a.m. (Figure 5). Hence, a busy person's utility space will be made of narrow width Steps. A free (not busy) person groups his time with large intervals (Figure 6). Hence, his utility space will be made up of wide Steps.

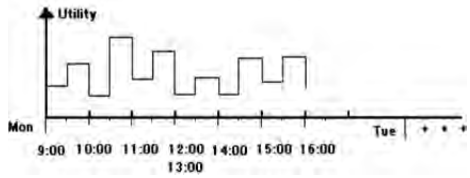


Figure 5: A busy person

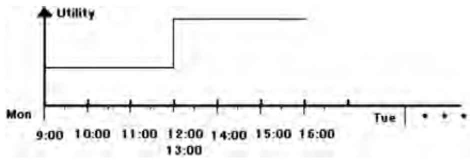


Figure 6: A free(not busy) person

4. COPE ALGORITHM

The COPE algorithm can locate the optimal contract more efficiently than EM when the COPE condition is satisfied. In Figure 7, a branch of a utility space is shown for four agents (Ag. h, i ,j and k). The optimal contract could be found by taking Step C of Ag. h (the Step with the highest utility) and matching it with the steps of Agents i,j and k. We call this method of computing the optimal contract COPE. Since the agents i,j and k have just one Step in the branch, just using the maximum Step of Ag. h is sufficient to correctly compute the optimal contract. For a branch the COPE condition is satisfied if,

1. Only The first agent in the matching lineup is sensitive; that is, it has many narrow width Steps.
2. The rest agents in the matching lineup have one wide Step which contains all the contracts in the branch.

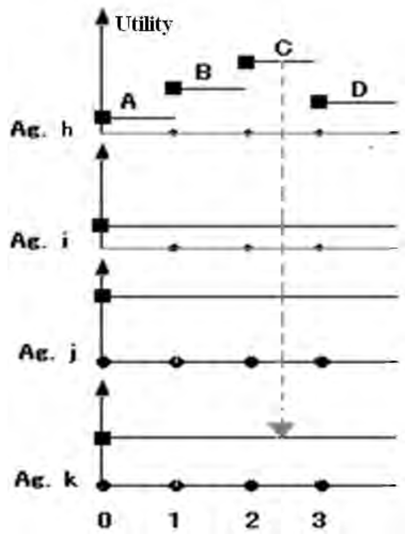


Figure 7: COPE Algorithm

5. FASTCOPE ALGORITHM

The COPE condition imposes stringent requirements on utility spaces of agents. One that could be relaxed is the requirement that the sensitive agent has to be the first in the matching line up. FASTCOPE algorithm is designed to compute the optimal contract efficiently even when the position of the sensitive agent is not known before hand. FASTCOPE algorithm extends COPE by rearranging the agents so that COPE condition is created before matching. The steps in the algorithm are:

- Step 1: Identify the sensitive agent.
- Step 2: Rearrange the agents. That is, place the sensitive agent in the first position of the matching lineup.
- Step 3: Execute COPE on the rearranged agents.

To identify the sensitive agent, FASTCOPE samples the first Step of each agent for the branch and reads its width. The Step from the sensitive agent will have narrower width than the insensitive agents.

6. EM VS COPE VS FASTCOPE

We compared the efficiency of EM, COPE and FASTCOPE experimentally. The result is shown in Figure 8. As expected COPE and FASTCOPE have higher efficiency than EM. COPE (20%) means, 20% percent of the branches satisfy the COPE condition. The rest violate it by not having the first agent as the sensitive one. When COPE is applied on branches that do not satisfy the condition, it makes no efficiency improvement. FASTCOPE rearranges the agents and applies COPE to compute the optimal contract for the branch.

The experiments were done at sensitivity ratios of 1:1000, 1:100, 1:10 and 1:5. For example sensitivity ratio of 1:5 means, the entire contract space is divided into branches that contain 5 contracts each. In a branch only one agent is sensitive and it will have 5 narrow width Steps. Each of the remaining agents will have one wide Step. When the total number of the contracts in the negotiation is 10000, there will be 10000/5 , 2000 branches. In figure Figure 8, for each algorithm, the average of the running time costs of the algorithm at the four sensitivity ratios is shown. The number of agents in the negotiation was 4.

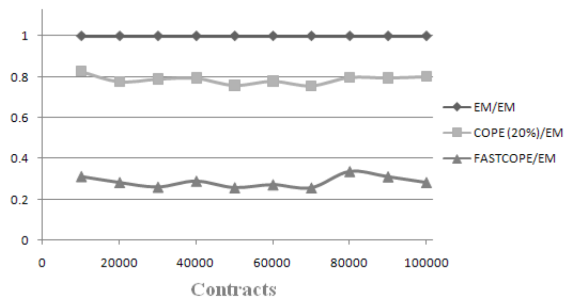


Figure 8: EMvsCOPEvsFASTCOPE

7. CONCLUSION AND FUTURE WORKS

This paper reports a preliminary work for designing efficient algorithm that compute the optimal contract correctly for agents that use the constraints based utility space model. The integer interval was identified to be the basic building unit of the model, and it was used to define the single issue version of it. It was argued that , the agents that use this model can be classified to two extreme kinds:sensitive and insensitive. COPE; an algorithm that computes the optimal contract for a branch correctly and efficiently when the first agent is sensitive and the others are insensitive is proposed. FASTCOPE extends COPE by relaxing the requirement that the sensitive agent has to be the first agent in the matching lineup.

Although FASTCOPE is efficient it imposes stringent requirements on the utility of space of agents. We aim to relax these requirements and increase the applicability of the algorithm. These include:In a branch, allowing more than one agent to be sensitive. Allowing some insensitive agents to have exceptional narrow width Steps. Allowing agents to independently branch their utility space. That is handling the case where the end points of the branches from each agent are not exactly the same (overlap).

Another future work is to extend the algorithm developed for the single issue version of the model to work for multiple issue version of it.

8. REFERENCES

[1] K. Fujita, T. Ito, H. Hattori, and M. Klein. An approach to implementing a threshold adjusting mechanism in very complex negotiations a preliminary result. *International Conference on Knowledge, Information and Creativity Support Systems(KICSS)*, pages 185–192, November 2007.

[2] K. Fujita, T. Ito, H. Hattori, and M. Klein. Effects of revealed area based selection method for representative-based protocol. *Agent-based Complex Automated Negotiations(ACAN)*, May 2008.

[3] H. Hattori, M. Klein, and T. Ito. Using iterative narrowing to enable multi-party negotiations with multiple interdependent issues. *In Proc. of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems,AAMAS*, pages 1043–1045, May 2007.

[4] T. Ito, H. Hattori, and M. Klein. Multi-issue negotiation protocol for agents exploring nonlinear utility spaces. *International Joint Conference on Artificial Intelligence(IJCAI)*, pages 1347–1352, January 2007.

[5] I. Marsa-Maestre, M. A.Lopez-Carmona, J. R. Velasco, and E. de la hoz. Effective bidding and deal identification for negotiations in highly nonlinear scenarios. *Proc. of 8th Int. conf. on Autonomous Agents and Multiagent Systems(AAMAS)*, pages 1057–1064, May 2009.

An Extension of Private Policy Matching for Bidirectional Automated Trust Negotiation

Fumihiro Mori
Graduate School of Information Science
Nagoya University
Furo-cho, Chikusa-ku, Nagoya
464-8601
mori@net.itc.nagoya-u.ac.jp

Hirofumi Yamaki
Information Technology Center
Nagoya University
Furo-cho, Chikusa-ku, Nagoya
464-8601
yamaki@itc.nagoya-u.ac.jp

Momoko Aoyama
School of Engineering
Nagoya University
Furo-cho, Chikusa-ku, Nagoya
464-8601
aoyama@net.itc.nagoya-u.ac.jp

ABSTRACT

Automated Trust Negotiation (ATN) is a mechanism to establish mutual trust between service providers and users in an open network environment like the Internet. In this paper, we propose Bidirectional Private Policy Matching based on Additively Homomorphic Encryption Systems (BPPM/AHES) as an ATN negotiation protocol where unidirectional private policy matching based on additively homomorphic encryption systems is repeated. In this protocol, the problems of existing ATN protocols such as unnecessary disclosure of credentials or that of policies before the negotiation succeeds are solved.

Keywords

trust, Automated Trust Negotiation (ATN), additively homomorphic encryption, private policy matching

1. INTRODUCTION

In an open network environment like the Internet, service users and providers are unknown to each other. Before using or providing services, the users will determine whether the providers are trustworthy, and the providers also wish to restrict their services only to trustworthy users. In this case, it is not easy for the users and the providers to negotiate for establishing mutual trust, because both of them want to disclose their own information only to the trust parties. Thus, the negotiations may fail unless they communicate successfully. Also, since the number of services has become enormous, it is very costly to establish mutual trust every time they encounter. To tackle this issue, Automated Trust Negotiation (ATN) [4] has been proposed to establish mutual trust between strangers. ATN is a process to automatically obtain the sequence to exchange credentials without violating disclosure policies that each party has. In existing ATN protocols [4], there is some problem such as unnecessary disclosure of credentials or that of policies before the negotiation succeeds.

2. ATN BASED ON ENCRYPTION SYSTEMS

2.1 Automated Trust Negotiation

We consider a situation where a user is trying to use a service, where we have two parties, a service user and a service provider. We refer them as a *client* and a *server* respectively in this paper. Both of them have their own digital *credentials* and *policies*. Credentials are digital data that contain such information as identi ers, names, contacts, af- liations, etc., and that are certi ed by trusted third parties. Policies are rules that de ne the condition on which the client and the server disclose their credentials to the counterpart. The server also has service-governing policies (SGP's) which gives on what condition the client is allowed to use the service. We use the same notation presented in [5] to describe these policies. We denote the service by R, credentials of the client by $C_1, \ldots, C_{n_c}$, credentials of the server by $S_1, \ldots, S_{n_s}$, where n_c and n_s are the numbers of credentials possessed by the client and the server respectively. The policy for disclosing the client's credential C is denoted by $C \leftarrow F_C(S_1, \ldots, S_{n_s})$, where $F_C(S_1, \ldots, S_{n_s})$ is a Boolean expression with the server's credentials $S_1, \ldots, S_{n_s}$. Similarly, SGP is denoted by $R \leftarrow F_S(C_1, \ldots, C_{n_c})$. The policy for disclosing the server's credential S is denoted by $S \leftarrow F_S(C_1, \ldots, C_{n_c})$, where $F_S(C_1, \ldots, C_{n_c})$ is a Boolean expression with the client's credentials $C_1, \ldots, C_{n_c}$. The policy of credential C will be $satisfied$, if the logical expression $F_C(S_1, \ldots, S_{n_s})$ evaluates to $true$, after substituting propositional symbols of already disclosed credentials by the other party with $true$ in the logical expression $F_C(S_1, \ldots, S_{n_s})$. If the policy of credential C is satis ed, it can be disclosed to the other party. Table 1 is an example of policies. If credential C can be disclosed without any credentials from the other party, such a policy is denoted by $C \leftarrow true$, and C is called an *unprotected* credential. On the other hand, if credential C cannot be disclosed under any circumstances, such a policy is denoted by $C \leftarrow false$. If SGP is satis ed as the result of negotiation, the service becomes available to the client.

The aim to perform ATN is to automatically obtain the sequence to exchange credentials without violating the policies in order to establish mutual trust between the client and the

Table 1: Example of Policies

client's policies	server's policies
$C_1 \leftarrow true$	$R \leftarrow (C_3 \wedge C_4) \vee C_6$
$C_2 \leftarrow true$	$S_1 \leftarrow true$
$C_3 \leftarrow S_1 \wedge S_2$	$S_2 \leftarrow (C_1 \wedge C_2) \vee C_3$
$C_4 \leftarrow S_3 \vee S_4$	$S_3 \leftarrow C_3 \vee C_4$
$C_5 \leftarrow S_2 \vee S_3$	$S_4 \leftarrow C_4$
$C_6 \leftarrow false$	$S_5 \leftarrow C_1 \wedge C_5$

Table 2: Example of Negotiation Table

client's policy			DF	MP
C_1	$\leftarrow$	$true$	0	$true$
C_2	$\leftarrow$	$true$	0	$true$
C_3	$\leftarrow$	$S_1 \wedge S_2$	0	
C_4	$\leftarrow$	$S_3 \vee S_4$	0	
C_5	$\leftarrow$	$S_2 \vee S_3$	0	
C_6	$\leftarrow$	$false$	0	

server. Various protocols and strategies have been proposed to achieve this [3, 4, 5]. Below, we brie y describe two basic strategies presented in [4], i.e., Eager Strategy and Parsimonious Strategy.

2.1.1 Eager Strategy

In Eager Strategy [4], the client and the server in turns exchange all the currently unlocked credentials. As credentials are exchanged in the negotiation, more credentials become unlocked. The negotiation succeeds when SGP is satis ed by the credentials disclosed by the client, and fails when the client terminates the negotiation because either of the negotiating parties has no credential to newly disclose. The negotiation process of Eager Strategy is very simple, and none of the server's and the client's policies is directly disclosed. The weakness of Eager Strategy is in that credentials are disclosed regardless of their contribution to the success of the negotiation, i.e., some of them may be unnecessarily disclosed. They are disclosed even if the negotiation fails.

2.1.2 Parsimonious Strategy

Eager Strategy is an approach to disclose all the credentials that can be disclosed, and no party sends requests for credentials to the other. In contrast, in Parsimonious Strategy [4], each party rst repeats sending requests for credentials to the counterpart, and discloses their credentials only after nding the sequence to exchange them for satisfying SGP. There is no disclosure of unnecessary credentials, but the existence of some policies that are not related to the sequence to exchange credential may be known to the counterpart. Most of protocols in the studies of ATN are extensions of Parsimonious Strategy, and su ers from the same weakness.

2.2 Private Policy Matching

Our goal is to create a new protocol that discloses neither credentials nor policies that do not contribute to the successful trust negotiation. In this section, we explain private policy matching [2] which is used in the process of BPPM/AHES. Kursawe et al. [2] proposed private policy matching based on the ElGamal cryptosystem [1]. It is a process to nd out whether a match exists between the credentials that a client can disclose (client's preference) and those requested by a server (server's preference). A set of matching credentials is called a *matching policy*. In private policy matching, both of parties encrypts their preferences, and from it they calculate a matching policy using the additively homomorphic property of the ElGamal cryptosystem. Because of this, they cannot acquire additional knowledge about the preference of the counterpart. Private policy matching enables us to derive a minimal set of credentials for the client to disclose, when only the server has policies and the client's credentials are unprotected. In the following section, we extend this to be applied in such bidirectional scenarious as in ATN, where both of the server and the client have policies.

3. BIDIRECTIONAL PRIVATE POLICY MATCHING

Our new protocol is derived by repeating private policy matching described in the previous section. Below, we call the original private policy matching as the *server-side policy matching*, where the server's policies are tested against the client's credentials. On the other hand, we call the opposite where the client's policies are examined as the *client-side policy matching*. By repeating the server-side and the client-side policy matchings alternately, the information exchange needed in ATN is achieved. We call our new protocol as Bidirectional Private Policy Matching based on Additively Homomorphic Encryption Systems (BPPM/AHES).

3.1 Negotiation Tables

This protocol takes the policies of the client and the server as its input, and outputs the sequence to exchange credentials. The process fails when no sequence is feasible. Both the client and the server maintain *negotiation tables* as exempli ed in Tables 2. These tables are updated through the negotiation process. A negotiation table has three columns, Policy, DF and MP. DF and MP stand for disclosing ag (DF) and matching policy (MP) respectively. A value in DF means whether the credential is disclosed (DF= 1) or not (DF= 0) in the negotiation process. The values in DF are initialized to 0, which means that all credentials are not disclosed before the negotiation. MP is for storing encrypted matching policies found in the negotiation. For a policy the right-hand side of which is *true*, its MP column is initialized to "unprotected."

3.2 Protocol

The BPPM/AHES protocol consists of two stages, policy negotiation and credential exchange. Below, we explain each of them.

3.2.1 Policy Negotiation

In policy negotiation, the client and the server repeat the server-side and the client-side policy matchings alternately until they know that whether there is a sequence to exchange credentials which satis es SGP. If there is such a sequence,

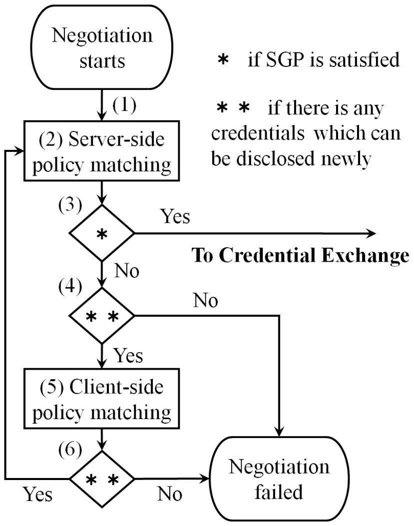


Figure 1: Policy Negotiation

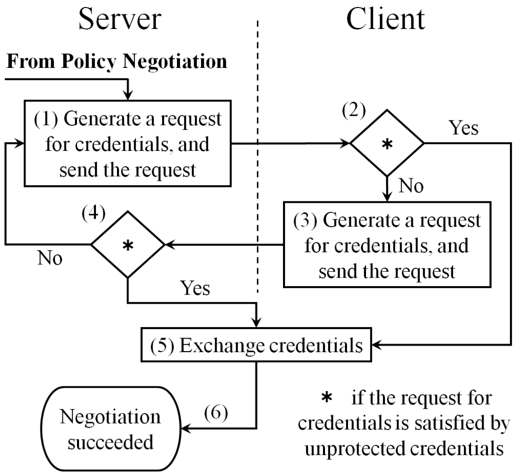


Figure 2: Credential Exchange

they move to credential exchange. If there is not, the negotiation process is terminated. The owchart of policy negotiation is depicted in Figure 1.

(1) First, the client requests a service to the server. The server sends a message that indicates the start of the server-side policy matching as reply.

(2) The client sets the value in DF of the credential that can be disclosed at that time to 1. The client and the server perform the server-side policy matchings in parallel for each of the server's policies. If a matching policy exists, the server sets the policy of the credential to *true*, and writes the matching policy in the column of MP of the credential.

(3) The server determines if SGP is satis ed. If it is satis ed, which means that the negotiation succeeds, the negotiating party move to credential exchange described in the next section. If it is not satis ed, move to (4).

(4) The server determines if there is any credential which can be disclosed and whose DF is 0. If there is such a credential, move to (5). If there is no such a credential, the negotiation fails and is terminated.

(5) The server sets the value in DF of the credential that can be disclosed at that time to 1. The client and the server perform the client-side policy matchings in parallel for each of the client's policies. If a matching policy exists, the client sets the policy of the credential to *true*, and writes the matching policy in the column of MP of the credential.

(6) The client determines if there is any credential which can be disclosed and whose DF is 0. If there is such a credential, move to (2). If there is no such a credential, the negotiation fails and is terminated.

3.2.2 Credential Exchange

After policy negotiation, the client and the server rst nd the sequence to exchange credentials and then exchange credentials according to it. The owchart of credential exchange is given in Figure 2.

(1) The server generates a request for credentials which must be disclosed by the client, for the server to satisfy the client's request for the server's credentials or service, and sends it to the client. A request for credentials is generated by matching policies stored in the columns of MP in the negotiation table. The matching policies are decrypted by the client and the server if needed to generate a request for credentials.

(2) The client determines if the request from the server is satis ed by the client's unprotected credentials. If it is satis ed, move to (5). If it is not satis ed, move to (3).

(3) The client generates a request for credentials which must be disclosed by the server, for the client to satisfy the server's request for the client's credentials, and sends it to the server.

(4) The server determines if the request from the client is satis ed by the server's unprotected credentials. If it is satis ed, move to (5). If it is not satis ed, move to (1).

(5) The client and the server nd the sequence to exchange credentials by reversing previous requests for credentials and exchange credentials according to it.

(6) When SGP is satis ed by the credentials which is disclosed last, the negotiation nishes in success.

3.3 Example of the Negotiation

In this section, we explain an example of negotiation in BPPM/AHES by showing the changes of the negotiation tables using the policies given in Table 1. The negotiation

Table 3: The Negotiation Table of the Client after the First Server-side and Client-side Policy Matchings

client's policy			DF	MP
C_1	$\leftarrow$	$true$	1	$true$
C_2	$\leftarrow$	$true$	1	$true$
C_3	$\leftarrow$	$\cancel{S_1 \wedge S_2}\ true$	0	E_{C_3}
C_4	$\leftarrow$	$S_3 \vee S_4$	0	
C_5	$\leftarrow$	$\cancel{S_2 \vee S_3}\ true$	0	E_{C_5}
C_6	$\leftarrow$	$false$	0	

Table 4: The Negotiation Table of the Server after the First Sever-side and Client-side Policy Matchings

server's policy			DF	MP
R	$\leftarrow$	$(C_3 \wedge C_4) \vee C_6$		
S_1	$\leftarrow$	$true$	1	$true$
S_2	$\leftarrow$	$\cancel{(C_1 \wedge C_2) \vee C_3}\ true$	1	E_{S_2}
S_3	$\leftarrow$	$C_3 \vee C_4$	0	
S_4	$\leftarrow$	C_4	0	
S_5	$\leftarrow$	$C_1 \wedge C_5$	0	

tables of the client and the server after the first server-side and client-side policy matchings are shown in Table 3 and 4.

At the first server-side policy matchings, since the client can disclose the credentials C_1 and C_2, the values in DF of credentials C_1 and C_2 in Table 3 are set to 1. When the first server-side policy matchings are performed, the server knows that there is a matching policy that satisfied the policy of server's credential S_2. The server sets the policy of the credential S_2 to $true$, and writes a matching policy E_{S_2} in the column of MP of the credential S_2. After that, the server starts the first client-side policy matchings with the client, because SGP is not satisfied and there are credentials S_1 and S_2 which can be disclosed newly. At the first client-side policy matchings, since the server can disclose the credentials S_1 and S_2, the values in DF of credentials S_1 and S_2 in Table 4 are set to 1. When the first client-side policy matchings are performed, the client know that there are matching policies that satisfied the policy of client's credentials C_3 and C_5 respectively. The client sets the policies of the credentials C_3 and C_5 to $true$, and writes matching policies E_{C_3} and E_{C_5} in the columns of MP of the credentials C_3 and C_5 respectively. After that, the client starts the second server-side policy matchings with the server, because there are credentials C_3 and C_5 which can be disclosed newly. Similarly, the second server-side and client-side policy matchings are performed. The negotiation tables of the server after the third server-side policy matchings are given in Table 5. Since SGP is satisfied at the third server-side policy matchings, they move to credential exchange.

In credential exchange, the server first decrypts E_R cooperating with the client, and obtains $E_R = \{C_3, C_4\}$. After that, the server generates a request for credentials $C_3 \wedge C_4$, and sends it to the client. Since the request is not satisfied by the client's unprotected credentials, the client decrypts E_{C_3} and E_{C_4} cooperating with the server, and obtains $E_{C_3} = \{S_1, S_2\}$ and $E_{C_4} = \{S_3\}$ respectively. Then, the client generates a request for credentials $S_1 \wedge S_2 \wedge S_3$, and sends it to the server. Similarly, the server and the client repeat requests for credentials each other. When the client received a request for credentials $C_1 \wedge C_2$, the client terminates a request because the server's request is satisfied by the client's unprotected credentials. The client and the server find the sequence to exchange credentials C_1, $C_2 \to S_1$, $S_2 \to C_3 \to S_3 \to C_4 \to R$ by reversing previous requests for credentials and exchange credentials according to it. When SGP is satisfied by the exchanged credentials, the negotiation finishes in success.

Table 5: The Negotiation Table of the Server after Third Server-side Policy Matchings

server's policy			DF	MP
R	$\leftarrow$	$\cancel{(C_3 \wedge C_4) \vee C_6}\ true$		E_R
S_1	$\leftarrow$	$true$	1	$true$
S_2	$\leftarrow$	$\cancel{(C_1 \wedge C_2) \vee C_3}\ true$	1	E_{S_2}
S_3	$\leftarrow$	$\cancel{C_3 \vee C_4}\ true$	1	E_{S_3}
S_4	$\leftarrow$	$\cancel{C_4}\ true$	0	E_{S_4}
S_5	$\leftarrow$	$\cancel{C_1 \wedge C_5}\ true$	1	E_{S_5}

4. CONCLUSION

In this paper, we proposed BPPM/AHES as an ATN negotiation protocol. We extended private policy matching proposed in a preceding work, and defined the server-side policy matching and the client-side policy matching. In each policy matching, calculations are performed using additively homomorphic properties of the ElGamal cryptosystem. The negotiation process proceeds by repeating the server-side and the client-side policy matchings alternately, until the sequence to exchange credentials without violating policies is found. In this protocol, the problems of existing ATN protocols are solved, and there is no disclosure of credentials and policies before the negotiation succeeds.

5. REFERENCES

[1] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory, 31(4):469–472, 1985.

[2] K. Kursawe, G. Neven, and P. Tuyls. Private policy negotiation. LECTURE NOTES IN COMPUTER SCIENCE, 4107:81, 2006.

[3] K. Nakatsuka and T. Ishida. Distributed AND/OR Search for Automated Trust Negotiation. Transactions, 47(8):2454–2463, 2006.

[4] W. Winsborough, K. Seamons, and V. Jones. Automated trust negotiation. In DARPA Information Survivability Conference and Exposition, volume 1, pages 88–102, 2000.

[5] T. Yu, X. Ma, and M. Winslett. PRUNES: an efficient and complete strategy for automated trust negotiation over the Internet. In Proceedings of the 7th ACM conference on Computer and communications security, pages 210–219. ACM, 2000.

Agreement among Agents based on Decisional Structures and its Application to Group Formation

Rafik Hedfi
Department of Computer Science, School of Techno-Business Administration,
Nagoya Institute of Technology.
rafik@itolab.mta.nitech.ac.jp

Takayuki Ito
Department of Computer Science,
School of Techno-Business Administration,
Nagoya Institute of Technology.
ito.takayuki@nitech.ac.jp

ABSTRACT
There has been an increased interest in automated negotiation systems for their capabilities in reaching an agreement through negotiation among autonomous software agents. In real life problems, the negotiated contracts consist of multiple and interdependent issues which tend to make the negotiation more complex. In this paper, we propose to define a set of similarity measures used to compare the agents' constraints, their utilities as well as their certainties over their possible outcomes. Precisely, we define a decision value-structure which gives a reasonable condition under which agents having similar decision structures can form a group. We think that a collaborative approach is an efficient way to reason about agents having complex decisional settings, but show similarities in their constraints, preferences or beliefs. Agents will tend to collaborate with agents having the same decisional settings instead of acting selfishly in a highly complex and competitive environment. Therefore, formed groups will benefit from the cooperation of its members by satisfying their constraints as well as maximizing their payoffs. Under such criterion, the agents can reach an agreement point more optimally and in a collaborative way. Experiments have been performed to test the existence of the decision value-structure as well as its capability to describe an agent's decision structure. Moreover, the decision value-structure was used for group formation based on measuring the agents similarities.

Keywords
Multi-attribute Utility, Decision Theory, Multi Objective Optimization, Uncertainty, Group Formation, Collaboration.

1. INTRODUCTION
Automated negotiation is a process by which a group of autonomous agents interact to achieve their design objectives. The agents will attempt to reach an agreement and satisfy their contradictory demands through a bargaining process. In an agent-mediated system, an important aspect of the solution is the way in which the agents negotiate to propose contracts to each other, under specific requirements and constraints. In real life situations, agents have to take into consideration multiple attributes simultaneously during the bargaining process, such as the quality, quantity, delivery time, etc. ([7]). In this paper, we propose to define a new approach to tackle the complexity of utilities with interdependent attributes by providing a new model for multi-attribute utility representation, which takes into consideration the possible interdependencies between attributes. In the real world, we believe that people who have similar decisional structures could reach an agreement more smoothly. In this paper, we propose also a new criterion for potential consensus under a number of assumptions, related to the decisional structure of the agent, defined as a Constraint-Utility-Belief space. In fact, adopting a cooperative behavior during the negotiation process may improve the performance of the individual agents, as well as the overall behavior of the system they form, by achieving their own goals as a joint decision [6]. To put this straightforward, we assume that our model is based on the following assumptions. In real life, we believe that people who have similar beliefs (certainties) relative to a specific situation, as well as the same preferences (utilities) over the same common outcomes (attributes), could reach a reasonable agreement more optimally and smoothly, than if they had different certainties or preferences over different outcomes. To support this claim, we first describe the different aspects of the decisional structure of an agent as a Constraint-Utility-Belief space. Most importantly, we define a unique decision value-structure for each agent, which gives a reasonable criterion, under which agents' decisional structures can be compared. We point out that in the case of similar decision value-structures, the agents can form groups, as an initial step before making coalitions which satisfy their constraints and maximizes their payoffs. Therefore, the agents can reach an agreement point more efficiently and in a collaborative way. We argue that the advantage of such approach is that the agents having strongly different decisional structures *i.e.* different decisional value-structures, do not need to cooperate. Instead, they can find agents having similar settings, and form groups.

At this end, in the case of multi-attribute negotiation we must define the main components needed by an agent to make decisions.

There have been several works in the context of multi-attribute negotiation for its importance in commerce as well as in social interactions. Different approaches and methods were proposed to analyze multi-attribute utilities for contracts construction. [12] presented the notion of convex dependence between the attributes as a way to decompose utility functions. [9] proposed an approach based on utility graphs for negotiation with multiple binary issues. [2] proposed also a model inspired from Bayesian and Markov models, through a probabilistic analogy while representing multi-attribute utilities. The same idea was firstly introduced by [11] through the notion of utility distribution, in which utilities have the structure of probabilities. Most importantly, a symmetric struc-

ture that includes both probability distributions and utility distributions was developed. In another work by [8], a similar concept was introduced by the notion of Expected Utility Networks which includes both utilities and probabilities. [3] proposed a model which takes into consideration the uncertainties over the utility functions by considering a person's utility function as a random variable, with a density function over the possible outcomes.
The remainder of the paper is structured as follows. Section 2 provides a formal definition of our model based on the notion of Decisional Structure of an agent with all its components. Section 3 describes a method used by the agent to construct his proposals or contracts, based on his decisional structure. In section 4, we elaborate a possible usage of the decisional structure as a group formation criterion through a set of similarity metrics. In section 5, we generalize the use of those metrics by the Decisional Value-Structure function as a method to compare agents' decisional structures. The experiment and the analysis of the model are described in section 6. In section 7 we present the conclusions and outline the future work.

2. DECISIONAL STRUCTURE

In the following section we will provide an overview of our theoretical model used for the representation of an agent's decisional structure. In fact, by decisional structure, we refer to the overall settings or information used by the decision maker *i.e.* the agent, to elaborate his strategies and make his decisions. In other words, the decisional structure of an agent can be considered as the decision space of the agent representing all his possibilities. Therefore, we will initially focus on a microscopic representation of an agent i regardless from his environment or the other agents. The macroscopic view will be developed in the next sections in the case of group formation. An agent i will define a unique tuple (1) representing his decisional structure.

$$Agent\ i \longmapsto (G_i, U_i, B_i) \tag{1}$$

This tuple will be characterized by the attributes and constraints of the agent i, represented by a Directed Acyclic Graph G_i [2] . The preferences of the agents will be represented by the utilities U_i of the agent. The agent's beliefs or certainties will be represented by the probability distributions B_i. The tuple can be described in the equations (2).

$$G_i = (V_i, E_i) \tag{2a}$$

$$V_i = \{v_j^i \sim a_j^i\}_{j=1}^n,\ a_j^i = (x_1, ...x_{m_j}) \tag{2b}$$

$$E_i = V_i \times V_i = \ \{d_j\}_{j=1}^{m_d} \tag{2c}$$

$$U_i = \ \{u_j^i\}_{j=1}^n \tag{2d}$$

$$B_i = \ell_i = \ \{\ell_j^i\}_{j=1}^n \tag{2e}$$

$$= \ \{\ \ell_j^i\ [p_{i,j,1} : x_{i,j,1},\ \ldots p_{i,j,m} : x_{i,j,m_j}]\ \}_{j=1}^n \tag{2f}$$

The static structure of the agent in (2a), defines the attributes (2b) and the dependencies (2c) between them, represented as a Directed Acyclic Graph G_i. In (2b), each vertexv_j^i of the graph corresponds to an attribute a_j^i *i.e.* an outcome or a prospect. An attribute a_j^i is defined as a vector of the possible values that can be taken by a_j^i. In the discrete case a_j^i (2b) and in the continuous case $a_j^i \in [x_1,\ x_{m_j}]$. In (2c), constraints are represented by the arcs $\{d_j\}_{j=1}^{m_d} \subset G_i$, and connect the vertices representing dependent attributes. But, it can be used to compute the utilities by mirroring the same dependence structure as a conditional dependence between the utilities [11]. This dependence structure could be updated dynamically during a negotiation process when the agents are collaborative. In (2d), utility functions U_i of the agent i represented as a function-vector $\{u_j^i\}_{j=1}^n$. In our model, we assume that the decision maker *i.e.* the agent follows the axioms of normative utility functions ($\sum_j u_j^i = 1$) [13]. Furthermore, we assume that the used utility functions have the properties of non-satiation (${u'}_j^i(x) > 0$) and risk aversion (${u''}_j^i(x) < 0$) [5]. Each utility function u_j^i is defined over a domain D_j related to the possible values taken by the attribute a_j as in (3).

$$u_j^i : D_j \rightarrow [0, 1] \tag{3}$$

Another important aspect of our utility functions is that they are defined in term of dependencies as conditional utilities, and therefore embody the notions of conditional probabilities and probability independence [11]. In our model, we use this representation for the computation of the utilities in respect to the functional dependencies. We refer the reader to the work proposed in [2] and related to conditional utilities and the conditional independence. In (2e), the belief or the certainty structure B_i of an agent i characterized by all the lotteries $\{\ell_j^i\}_{j=1}^n$ (2f) where each lottery ℓ_j^i is associated to the attribute a_j^i, according to the probability distribution $p_{i,j}$ over the outcomes $x_{i,j,k} \in a_j^i$ with $\sum_{k=1}^{n_j} p_{i,j,k} = 1$. The lotteries of an agent i over the set of attributes a_j^i can be represented by the lottery (4).

$$\ell_j^i\ [p_{i,j,1} : x_{i,j,1},\ \ldots, p_{i,j,n} : x_{i,j,n}] \tag{4}$$

The probabilities $p_{i,j}$ are the subjective probabilities [1] of the agent i and represent his certainties about the possible outcomes. Each probability associated to an attribute, can be seen as a random variable over the possible values of an attribute [3].

3. UTILITY MAXIMIZATION

3.1 Contract Representation

An agent i will represent a contract $\vec{C^i}$ as a vector of attributes $\vec{C^i} = (\ a_1^i, \ldots, a_j^i, \ldots, a_n^i\ )$, where each attribute corresponds to a vertex $v_j^i \in V_i$ as we mentioned in (2b). Therefore, finding the optimal contract $\vec{C^*}$ having the highest utility among the contracts $\vec{C}_{i \in N}$, corresponds to solving the objective function (5) [4].

$$\vec{C^*}\ =\ \arg\max_{\vec{C}} \sum_{i \in \mathbb{N}} u_i(\vec{C_i}) \tag{5}$$

However, we assume the existence of a number of constraints, describing the relations or interdependencies (2c) between the attributes [2] . In other words, to compute the utility of a single attribute, we must take into consideration the other attributes. Meanwhile, we will associate a specific utility function u_i to each attribute a_i, with i as an attribute index. The overall utility of a contract $\vec{C}$ can be represented in the equation (6).

$$u(\vec{C})\ =\ \sum_{a_i \in \vec{C}} u_i(a_i / \{a_{j \neq i}\}) \tag{6}$$

It is obvious that none of the overall attributes are needed to compute the utility of a single attribute. It means that based on a graphical representation of the interdependencies (2c), we will only use the connected attributes. The edges d_i representing the constraints or dependencies between attributes. Since the dependencies will exist only between the connected vertices, each vertex a_i will de-

Table 1: Conditional Utility functions

Utility u_i	Conditional Utility $u_i/\{u_j\}_{j=1}^{7}$
u_1	u_1
u_2	u_2
u_3	$u_3/\{u_1, u_2\}$
u_4	u_4
u_5	$u_5/\{u_3, u_4\}$
u_6	u_6
u_7	$u_7/\{u_4\}$

pend on its parent vertices giving the equation (7).

$$u(\vec{C}) \;=\; \sum_{a_i \in \vec{C}} u_i(a_i/\pi(a_i)) \tag{7}$$

Where $\pi(a_i)$ is the set of all the parents of the vertex a_i. This representation means that in order to compute the utility of the attribute a_i we need to use the attributes $\pi(a_i)$ and their corresponding utility functions. Therefore, the objective function (5) can be written as $\vec{C^*} \;=\; \arg\max_{\vec{C}} u(\vec{C})$. The final equation is described as in (8)

$$\vec{C^*} \;=\; \arg\max_{\vec{C}} \sum_{a_i \in \vec{C}} u_i(a_i/\pi(a_i)) \tag{8}$$

3.2 Example of Contract Construction

Suppose we are dealing with contracts with a number of attributes equal to 7. The goal is to find the optimal contract $\vec{C^*}$ satisfying the interdependencies between the attributes. Each agents will organize his attributes and constraints in a specific way defined by the Directed Acyclic Graph in **Figure 1**.

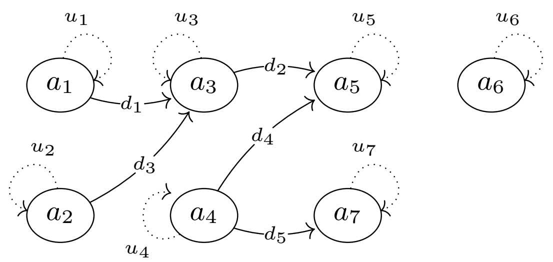


Figure 1: Constrained attributes

As we can see in **Figure 1**, the DAG will represent the contract from a statical viewpoint *i.e.* the structure and the interdependencies between the attributes. Moreover, a utility function u_i has to be associated to each vertex v_i, in order to compute the utility of the corresponding attribute a_i. Based on the graph in **Figure 1**, the interdependency relations between attributes will yield the same dependencies among the utility functions as shown in **Table 1**. In the concrete case, an attribute a_j can have different values and therefore will be represented by a vector $a_i \;=\; \{x_j \in D_i\}_{j=1}^{m_j}$ Maximizing an utility function u_i is finding the value $x^* \in D_i$ representing the maximal extrema of u_i such as in (9).

$$u_i(x^*) \;\geq\; u_i(x_k)\, \forall k \in [1, m_j] \tag{9}$$

Thus, we are interested in maximizing the sum of the increasing functions U_i. Therefore, the optimal contract can be written as a vector $\vec{C^*} \;=\; (\; a_1^*, \ldots, a_i^*, \ldots, a_n^* \;)$, where a_i^* is the maxima of u_i. The optimal contract's utility is computed according to the equation (10).

$$u(\vec{C^*}) \;=\; \sum_{i \in \mathbb{N}} u_i(a_i^*/\pi(a_i^*)) \tag{10}$$

3.3 Agent's Optimal Contract

The algorithm **Optimal_Contract** is used to find the optimal contract based on the attributes (2b), the utilities (2d), and the interdependencies among the attributes (2c).

Algorithm: Optimal_Contract
Input: DAG G_i of the Agent i
Output: Optimal Contract C^*

begin
  Topologic ordering of a_i according to $\pi(a_i)$;
  for $k \leftarrow |\pi(a_i)|_{min}$ **to** $|\pi(a_i)|_{max}$ **do**
    foreach a_i *satisfying* $|\pi(a_i)| = k$ **do**
      Find a_i^* satisfying
      $u_i(a_i^*) \geq u_i(x_j),\ j \in [1, m_i],\ x_j \in D_i$;
    end
  end
  $C^* \leftarrow (\; a_1^*, a_2^*, a_3^*, \ldots, a_i^*, \ldots, a_n^* \;)$;
  return C^*
end

Algorithm 1: Optimal contracts finding

Based on our example in **Figure 1**, the vertices a_i will be sorted according to the number of parents *i.e.* the in-degree $deg^-(a_i)$, which will describe the number of constraints of the related attribute.
An attribute a_i with $deg^-(a_i) = 0$ is called a *free attribute*, as the corresponding utility is computed only by using the attribute a_i's utility function u_i without any reference to other utility functions or other attributes. Similarly, an attribute with $deg^+(a_i) > 0$ is a *non-free attribute* or *dependent* and is subject to $deg^+(a_i)$ constraints. The topological sort of the attributes a_i within G_i is based on the $deg^-(a_i)$.

4. GROUP FORMATION

4.1 Group formation metrics

The nonlinearity and the complexity of the agents preferences is basically due to the different constraints they are trying to satisfy, as well as their utilities and the way probabilities are affected. Generally, our approach tends to capture and analyze the similarities between the agents constraints, utilities and beliefs. Being part of the same group means that all its members have close constraints, utilities and certainties. Therefore, it is important to define the similarity functions, to be able to compare between two agents' decisional spaces and decide whether they can be part of the same group or not.

4.2 Metric related to the Graph

We define the measure sim as the degree of similarity between two graphs G_1 and G_2. In other words, how much the agents whose graphs G_1 and G_2 share constraints and how close they are in term of vertices and edges. The similarity measure is calculated by multiplying the *Jaccard* indexes relative to the vertices and the edges sets.

This similarity measure can be definied by (11).

$$sim : G \times G \rightarrow [0,1] \tag{11a}$$
$$sim(G_1, G_2) = J_V(V_1, V_2) \times J_E(E_1, E_2) \tag{11b}$$
$$= \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|} \times \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|} \tag{11c}$$

The extreme value $sim(G_1, G_2) = 0$ means that the agent 1 and the agent 2 do not have the same attributes nor share common constraints, whereas $sim(G_1, G_2) = 1$ means that they have exactly the same attributes and the same constraints. Therefore, it might be interesting to consider these similarities' measures between agents' DAGs as a way to form groups and maybe think of potential coalitions. Under these hypothesis, each agent i has a vector $SG_i = \{sim(G_i, G_k)\}_{k \neq i}$ containing all the similarity values between his graph G_i and the other agents' graphs G_k. Using this vector, the agent can selected the set of agents having similar structures (attributes, constraints). This can be a first step for a future collaboration between the agents being part of the same group.

4.3 Metric related to the Utilities

As mentioned in **2**., the utility functions have the properties of *non-satiation* and are *risk aversion* . Under these hypothesis, we assume that the behavior of these functions can be used to compare the utilities of two agents. Let's consider two utility functions $u_i : D_i \rightarrow [0,1]$, $u_j : D_j \rightarrow [0,1]$ and the domain $D = D_i \cap D_j$. If we suppose that u_i and u_j are similar ($u_i \sim u_j$), then (12) holds.

$$u_i \sim u_j \Longrightarrow \forall x \in D, \exists \epsilon, |u_i(x) - u_j(x)| \leq \epsilon \tag{12}$$

The main purpose of comparing utility functions is finding a similarity measure enabling us to say whether two agents have the same preferences over the same outcome (attribute) or not. We can propose a way to compare two agents' utilities by comparing their accumulated wealth for the same outcome x. In this case, we have to consider the utility value as if it was a cumulative distribution function. Comparing two agents' utilities u_i and u_j is comparing their integrations from the last preferred outcome x_{min} up to the outcome x. Therefore (13) holds.

$$u_i \sim u_j \Longrightarrow \int_{x_{min}}^{x} (u_i(x) - u_j(x))\,\mathrm{d}x \simeq 0 \tag{13}$$

The comparison measure of two utility functions u_i and u_j up to an outcome x will be defined as in (14).

$$sim(u_i, u_j) = \int_{x_{min}}^{x} (u_i(x) - u_j(x))\,\mathrm{d}x \tag{14}$$

We notice that both utilities have the same type *i.e.* correspond to the same outcome (domain). Therefore comparing the overall n utilities U_i and U_j of two agents i and j can be determined as in (15).

$$sim(U_i, U_j) = \prod_{k=1}^{n} sim(u_k^i, u_k^j) \tag{15}$$

4.4 Metric related to Beliefs

The agents have different certainties when it comes to decide about the outcomes and their related preferences. Therefore, we think about a way to compare these certainties defined as lotteries. Two agent i and j will share the same certainties (beliefs) for an outcome a_k, if their respective probability distributions p_k^i and p_k^j over a_k are close or similar. A possible way to consider this similarity is to use the cross entropy. Assuming that for a certain attribute $a_k = (x_1, ...x_{m_k})$ and for two lotteries ℓ_k^i and ℓ_k^j relative to two agents i and j, each lottery will correspond respectively to a probability distributions p_k^i and p_k^j over a_k. Therefore, we can define the cross entropy of p_k^i and p_k^j as in (16).

$$sim(p_k^i, p_k^j) = \sum_{l=1}^{m_k} p_k^i(x_l) \log[p_k^j(x_l)] \tag{16}$$

Generally, each agents i has a vector of lotteries ℓ_i over the n attributes and defined as his certainty structure B_i as in (2e) and (2f). We can define a similarity measure comparing two agent's certainty structures B_i and B_j as in (17).

$$sim(B_i, B_j) = \sum_{k=1}^{n} sim(p_k^i, p_k^j) \tag{17}$$

5. DECISIONAL STRUCTURE VALUE FUNCTION

After defining the agent's metrics we will focus on how to exploit them in order to satisfy the common constraints as well as the possible similarities between the agents's belief and utilities. For example, the agents sharing the same constraints (same graphs structure) and having the same beliefs (same probability distributions over the outcomes) could form groups by opening and sharing their utility functions according to a specific strategy. As in (1), the tuple (G_i, U_i, B_i) of an agent i describes his constraints, preferences and beliefs in a way that identifies the agent from the other agents' configurations. However, if the values G_i, U_i and B_i represent in a unique way their corresponding agent, it is possible to construct a bijective function f which maps each agents tuple (G_i, U_i, B_i) to a unique real value $dsv_i \in [0,1]$ identifying the agent in a unique way. This function can be assimilated to an *Hilbert Space Filling Curve* [10] or can be constructed by a binary expansion of real numbers. This function can be described by the definition (18).

$$f : D_J \times D_U \times D_P \rightarrow [0,1] \tag{18a}$$
$$f(g_i, u_i, p_i) = dsv_i \tag{18b}$$

The domains D_J, D_U and D_P of f are equal to [0,1]. We will develop in the next section the proper use of this function f in the context of group formation and agents clustering. The function f must be injective $i.e.$ for two agents i and j having different settings (g_i, u_i, b_i) and (g_j, u_j, b_j) we will have (19).

$$(g_i, u_i, b_i) \neq (g_j, u_j, b_j) \Longrightarrow f(g_i, u_i, b_i) \neq f(g_j, u_j, b_j) \tag{19}$$

It is possible to prove not only the existence of an injection from $[0,1]^3$ to $[0,1]$ but also a bijection. In fact, that bijection exists and it can be proven using the *Cantor-Bernstein-Schroeder* theorem as following :

i. There is an injection g satisfying (20).

$$g : [0,1] \rightarrow [0,1]^3 \tag{20a}$$
$$g(x) = (x, 0, 0) \tag{20b}$$

ii. It is possible to define an injection $h : [0,1]^3 \rightarrow [0,1]$ given by representing the tuple (x, y, z) in binary and then interlacing the digits before interpreting the result in base 10, yielding the image of (x, y, z). Using binary for the representation of the strings is a way to avoid the 9's with the dual representation in base 10 and therefore, preserving the injection.

iii. Based on $i.$ and $ii.$, we can apply the *Cantor-Bernstein-Schroeder* theorem, which states that if there are two injections g and h as in (21a) and (21b),

$$g : A \rightarrow B \tag{21a}$$
$$h : B \rightarrow A \tag{21b}$$

Then there is a bijection f between A and B. Hence, it is possible to find f satisfying the condition (19).

An interesting usage of the function f is in a mediated negotiation where a mediator is gathering bids from the agents and trying to find the optimal contract. In fact, f provides to the mediator a way to group the agents based on their similarities without the need for the agents to open their utility spaces or their constraints. In this situation, the mediator can establish a feedback mechanism to update his constraints according to the settings of the agents. The convergence to the optimal solutions, ensuring social welfare, will be based upon the agents' feedback as well as the initially established mediator's constraints. Each agent i has only to provide the decisional structure value (dsv) which can be seen as a fuzzy indicator about the agent's Constraint-Utility-Belief Space ($[0, 1]^3$). Once these values are collected, the mediator can analyze and predict the possibilities of consensus reaching and the convergence to final contract. This is done before starting any utility space sampling or any computationally consuming task, used for example in [4].
The main advantage of using the dsv is to avoid bidding when the bids are likely to yield a complex and nonlinear utility space. Furthermore, having nonlinear space tends to make the consensus finding process complex, especially when there is a mediator. In fact, the mediator has to collect the bids and explore a highly nonlinear utility space in order to find the Pareto optimal contracts [4]. Instead, we can find an appropriate grouping of the bids based on certain criteria (including similarity measures) defined by the decisional structure values of the agents.
As we mentioned above, f is bijective, as the agents do not need to open their utilities nor their belief nor their constrains. Instead, they can know exactly how close and how similar their decision structures are and hence to decide whether to go for a collaborative strategy or act regardless from the others. The closeness degree between two agents stands upon the monotonicity of f when mapping to [0,1]. The function f can capture enough information that allows a meaningful clustering of agents based on their common interests : Constraints, Attributes, Utilities, Belief, Certainty, etc.

6. EXPERIMENTAL ANALYSIS

In the following experiments, we provide a method for group formation based on the similarity between the decisional values of the agents. We also provide an application of the decisional structure in the design of vectors called *vectorial design*.
Given the set $C = \{d_i\}_{i=1}^{N}$ of all the decisional structure values (dsv) of the agents, we propose to partition C into k disjoint clusters using the *K-Means* algorithm. Finding the optimal partitioning of C corresponds to finding the k clusters as in (22).

$$C^* = \arg\min_C \sum_{i=1}^{k} \sum_{d_j \in C_i} \|d_j - \delta_i\|^2 \tag{22}$$

Each cluster or group C_i is centered around a specific structure value δ_i which refers to the agent having the decisional structure that is more likely to describe the common features of the group C_i.

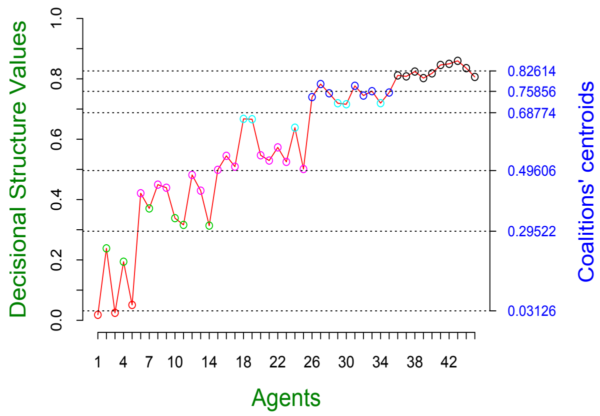


Figure 2: Agents' *dsv* values

Figure 2 illustrates a process of grouping of 45 agents, based on their decisional structure values *i.e.* d_j. We propose to partition these agents into 6 groups each of which is characterized by a group centroid δ_i. The resulting groups can be described by their corresponding centroids which are represented in **Figure 2** in blue, on the right axis.

The decisional structure values δ_i were generated based on the function f defined in **5.**, which was applied on the G_i, U_i and B_i variables of the 45 agents. The corresponding DSVs must be unique for each agent. Under such hypothesis, the injectivity of the function f will stand and there will be no risk for collisions *i.e.* two different agents, having different decisional structures but having the same DSV. Based on the original tuples G_i, U_i and B_i, we found that the agents being part of a group (C_j, δ_j) had close constraints, utilities and probabilities. This result was evaluated firstly by comparing the similarities between two agents decisional structure values dsv_i and dsv_j based on the distance $d = |dsv_i - dsv_j|$. Secondly, we measured the distances $d_g = sim(G_i, G_j)$, $d_u = sim(U_i, U_j)$ and $d_b = sim(B_i, B_j)$, defined in **4**. We found that the distance d is related to the distances d_g, d_u and d_b. The result confirms the characteristics of the bijective function f defined in **5.**, and its ability to describe uniquely an agent's decisional structure.
In **Figure 3**, we can see that there is a number of agents grouped around the same dsv value. In this case, the agents 2, 3, 4, 5 and 9 can be grouped into a cluster G based on the assumption that they have common decisional structures. According to this information, and whenever its shared to the overall agents (1 to 10), the agents not being part of G can choose to join this group or not. In case they accept to join, it is probable that they should start adapting and updating their constraints, preferences and beliefs similarity to the initial agents of G.

Generally, The decisional value structures are constructed based on the graphical constraints, utilities and beliefs. As we can see in **Figure 4**, the red curve represents the graphical constraints values, the utilities are represented by the green curve, and the blue values represent the beliefs. The overall similarity is represented by the black curve. For example, we can see that the agent 1 and

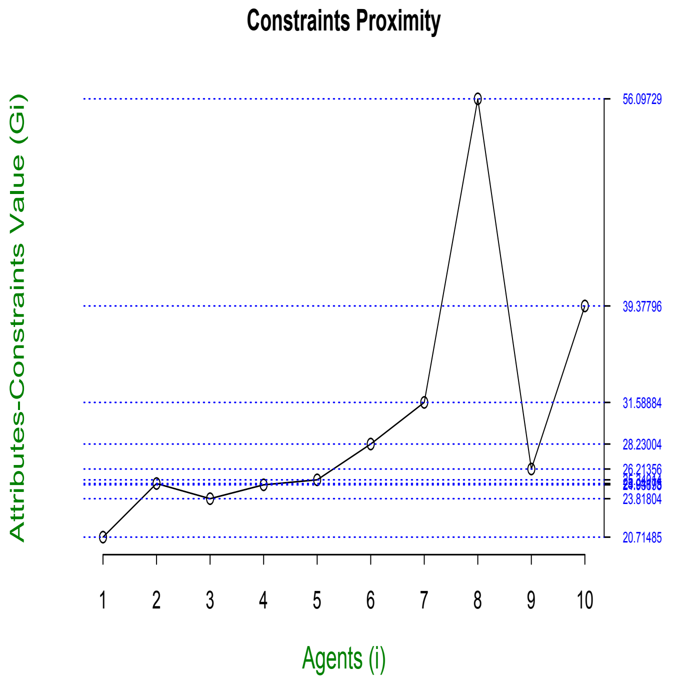


Figure 3: Dominant Group

Table 2: Agents' vectors values

$Agents$	x_1	x_2	$Values$
A_1	0.12	0.96	0.05683
A_2	1.87	1.83	0.68083
A_3	1.34	1.45	0.38637
A_4	1.41	1.57	0.44097
A_5	2.32	2.92	1.36735
A_6	2.39	3.01	1.4523

the agent 6 have close DSVs, and this can be seen based on the closeness in the red, green and blue curves *i.e.* the graphical constraints points, utilities and belief points.

A concrete application of such method of comparison is the case of *vectorial design*, where a user designs graphically a vector. A vector can represent an object, a product, or more generally a multi-attribute contract. As an example, 6 agents are designing 6 different vectors. For the sake of simplicity, we can think about the vector as a 2-points vector with components x_1 and x_2. In **Table 2** we can see that for each two values x_1 and x_2 we can represent the design vector by a unique value, locating the agents design in the overall designed vectors. This will give an idea about the degree of closeness between the designed vectors. The degree of closeness of the agents's vectors can be provided as a shared information to the overall agents while they are designing their vector. In fact, sharing such information dynamically and in real time can give the agents an idea on how their vectors are located in the group, and how to change their vector accordingly. This information can be represented as in **Figure 5**, and is available to each agent. On the x axis, we have the agents's indexes from 1 to 6 represented by 6 bars, and on the y axis we represent their corresponding values. When-

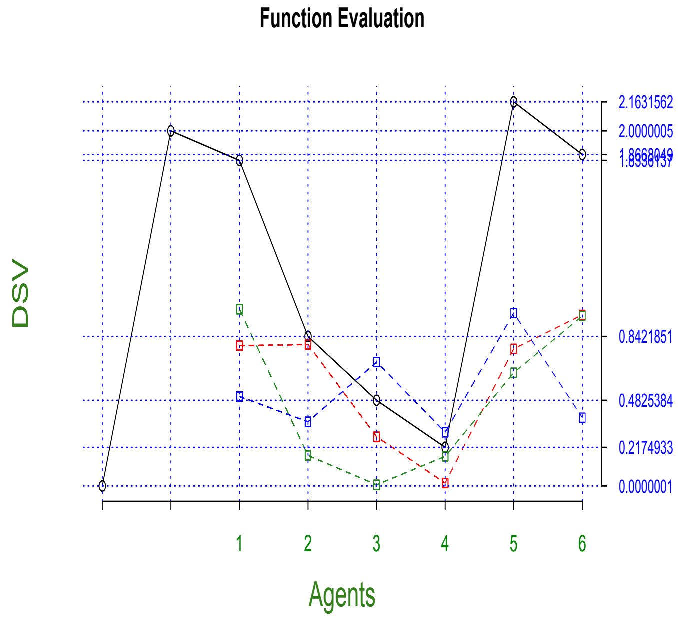


Figure 4: DSVs comparison

ever an agent changes his vector, the representation in **Figure 5** will change accordingly. Such method of collaborative design will

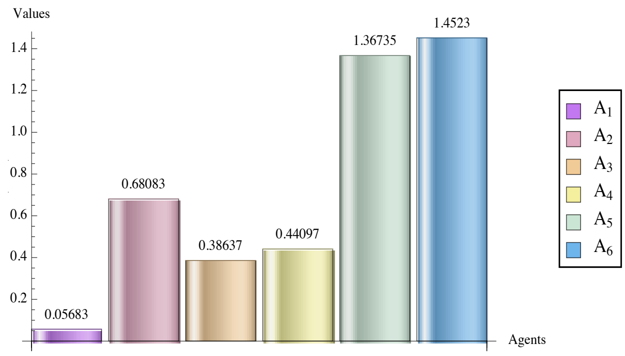


Figure 5: Proximity of the designed vectors

give the agents the possibility to orient their design based on the overall group's preferences, ensuring social welfare. It is possible to extend the simple vector represented by x_1 and x_2 to a more complex vector. Another example of vectorial design is represented in **Figure 6** where 7 agents are designing 7 vectors. At different times, each agent A_i will provide a vector $V_{A_i} = (X_{i1}, X_{i2})$, where X_{ij} are real values. During the design process, each agent A_j can visualize the similarities between his design and the other agents $A_{k \neq j}$ as in **Figure 7**. Therefore A_j can update his vector according to the evolution of the other agents' designs.

The represented values in **Figure 7** correspond to the designed vectors represented in **Figure 6**. We can can see that the vectors V_{A_6} and V_{A_5} are graphically close in **Figure 6**, therefore their corresponding values in **Figure 7** will be also close (1.30555 and 1.4523). The same comparisons can be done to the other vectors,

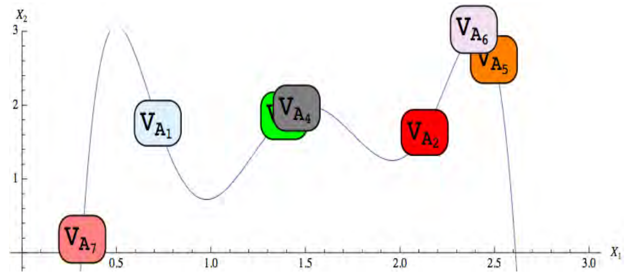


Figure 6: Vectors representation

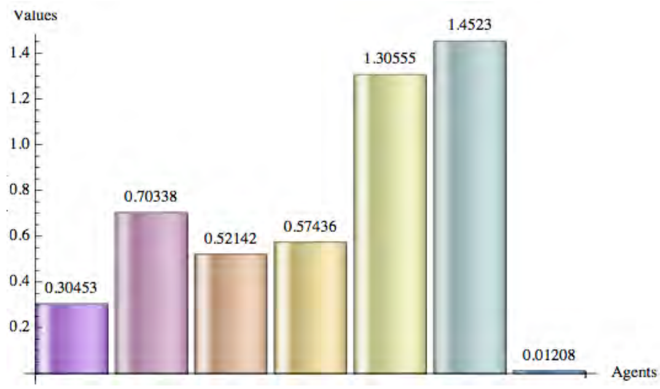


Figure 7: Decisional Values representation

allowing the agents to see the likelihood and the convergences of the global design.

7. CONCLUSION

The contributions of this paper are two-fold. On the one hand, we proposed a theoretical model to reason about multi-attribute contracts representation taking into consideration the attributes' interdependencies. On the other hand, we provided the notion of decisional structure value as a main criterion for agents' decisional settings comparison. The defined structure-value captures the main similarities between the agents' decisional settings. We have shown that it is possible to represent such decisional setting as a Constraints-Utilities-Belief space. Furthermore, we provided an example of usage of such value in the case of group formation based on the degree of similarity between the agent's decisional spaces.

As a future work, we would like to consider the performances of the method used to generate the decisional structure value. Moreover, we would like to elaborate a complete negotiation process, by defining a concrete protocol based on the formed groups. For example, we can develop the case where the agents being part of the same group can open and share their utility functions.

8. REFERENCES

[1] F. J. Anscombe and R. J. Aumann. A definition of subjective probability. *The Annals of Mathematical Statistics*, 34(1):199–205, 1963.

[2] R. I. Brafman and Y. Engel. Directional decomposition of multiattribute utility functions. In *ADT '09: Proceedings of the 1st International Conference on Algorithmic Decision Theory*, pages 192–202, Berlin, Heidelberg, 2009. Springer-Verlag.

[3] U. Chajewska and D. Koller. Utilities as random variables: Density estimation and structure discovery. In *In Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 63 – 71, 2000.

[4] T. Ito, H. Hattori, and M. Klein. Multi-issue negotiation protocol for agents : Exploring nonlinear utility spaces. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pages 1347–1352, 2007.

[5] R. L. Keeney and H. Raiffa. *Decisions with multiple objectives*. Cambridge University Press, 1993.

[6] S. Kraus and O. Schechter. Strategic-negotiation for sharing a resource between two agents. *Computational Intelligence*, 19:9–41, 2003.

[7] G. Lai and K. Sycara. A generic framework for automated multi-attribute negotiation. *Group Decision and Negotiation*, 18:169–187, 2009.

[8] P. L. Mura and Y. Shoham. Expected utility networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 1999.

[9] V. Robu, D. J. A. Somefun, and J. L. Poutre. Modeling complex multi-issue negotiations using utility graphs. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005)*, pages 280–287, 2005.

[10] H. Sagan. Space-filling curves. New York, 1994. Springer-Verlag.

[11] Y. Shoham. A symmetric view of utilities and probabilities. In *Proceedings of the Fifteenth international joint conference on Artificial intelligence (IJCAI'97)*, pages 1324–1329, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[12] H. Tamura and Y. Nakamura. Decompositions of multiattribute utility functions based on convex dependence. *Operations Research*, 31(3):488–506, 1983.

[13] J. von Neumann and O. Morgenstern. In *Theory of Games and Economic Behavior, 2nd ed. Princeton University Press, Princeton, NJ.*, 1947.

An Adaptive Bilateral Negotiation Model Based on Bayesian Learning

Chao Yu
School of Computer Science and Software Engineering
University of Wollongong
Wollongong, NSW, Australia
cy496@uowmail.edu.au

Fenghui Ren
School of Computer Science and Software Engineering
University of Wollongong
Wollongong, NSW, Australia
fren@uow.edu.au

Minjie Zhang
School of Computer Science and Software Engineering
University of Wollongong
Wollongong, NSW, Australia
minjie@uow.edu.au

ABSTRACT

Endowing the negotiation agent with a learning ability such that a more beneficial agreement might be obtained is increasingly gaining emphasis in agent negotiation. In this paper, we present a novel bilateral negotiation model based on Bayesian learning to enable self-interested agents to adapt negotiation strategies dynamically during the negotiation process. Specifically, we assume that two agents negotiate over a single issue based on time-dependent tactic. The negotiation agent has a belief about the probability distribution of its opponent's negotiation parameters (i.e., the deadline and reservation offer). By observing the historical offers of the opponent and comparing them with the fitted offers derived from a regression analysis, the agent can revise its belief using the Bayesian updating rule and can correspondingly adapt its concession strategy to benefit itself. By being evaluated empirically, this model shows its effectiveness for the agent to learn the possible range of its opponent's private information and alter its concession strategy adaptively.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multi-agent systems

General Terms

Learning, Algorithm

Keywords

Agent negotiation, Bayesian learning, Concession strategy

1. INTRODUCTION

In recent years, researchers in multiagent systems have paid their increasing attentions to the integration of learning techniques into agent negotiation [2] [1] [8] [5] [11]. In this type of learning circumstances, agents need adapt themselves to the changes of opponents and/or the environment through learning in order to achieve a satisfactory result. However, due to the essence of competition, privacy and uncertainty in real life negotiation [3], negotiators are always unwilling to reveal their private information (e.g., parameters such as the deadline, reserve price, or strategy profiles) to their opponents in case of being forced to a worse outcome, thus making learning in negotiation a challenging problem.

In current literature, a number of approaches have been developed by employing agents learning methods into negotiation process. Zeng and Sycara proposed an approach based on Bayesian learning to learn the opponent's reserve price [12]. Their approach assumed that agents have priori knowledge about the opponent's bidding strategy. This assumption may not always be true in real-world negotiation. Hindriks and Tykhonov also proposed an approach to discover opponent's information [5] by using Bayesian learning based on the assumptions that 1) agents know about the opponent's weights ranking on negotiation issues and 2) all agents' preferences can be modelled by three proposed functions, which may impact the use of this approach in a wide range when these assumptions conflict with the real world situations. Ren and Zhang introduced a regression analysis approach to predict the opponent's concession strategy by using the historical offers only [9]. However, their approach did not give further advice on how to adapt agent self's concession strategy based on the learning results. Brzostowski and Kowalczyk also presented a way to estimate partners' behaviors in different types of agents, based only on the historical offers in the current negotiation [3]. However, the accuracy of classification on partners' types may impact the accuracy of prediction results. The current challenging issues in agent learning during negotiation include (1) how to design a learning method without priori knowledge of the opponent's private information, (2) how to develop an effective learning strategy only based on the historical offers of current negotiation, and (3) how to produce a constructive guidance from learning to adapt agent's negotiation behaviors so as to achieve a better negotiation outcome.

This research attempts to solve the above three changing issues. In this paper, we propose a novel model by combining Bayesian learning and a regression analysis approach to dynamically learn the opponent's negotiation deadline and reservation offer. Firstly, a negotiation agent defines some regions and evenly initialize the probability of each region. The probability here indicates how likely that the opponent's deadline and reservation offer are located in the corresponding region. By using the predefined regions, the agent can have some estimations on the opponent's negotiation behaviors. Secondly, by using the regression analysis, the differences between the opponent's real negotiation behavior and the agent's estimation results are calculated. The more similar between the opponent's real behavior and an estimated behavior, the more likely that the opponent's real deadline and reservation offer will be located in the corresponding region. Thirdly, based on the similarities between the opponent's real behavior and the estimated behaviors, the probabilities assigned to each region will be updated dynamically through Bayesian learning. Lastly, the agent will propose a countermeasure for each estimated behavior of the opponent, and all countermeasures will be combined based on the likelihood of each estimated behavior. The combined result will be employed by the agent to perform a reasonable reaction. During the negotiation, each region's probability will be dynamically updated and gradually close to the real situation. Thus, the agent will also gradually adapt its negotiation strategy to reach a better negotiation outcome. Our model only use historical offers in the current negotiation, without requesting prior knowledge about the environment and the opponent.

The remainder of this paper is organized as follows. In Section 2, we recap the general negotiation model, especially the basic principles of the time dependent tactic. The proposed learning model is introduced in detail in Section 3, and in Section 4 empirical evaluation and analysis are presented. The discussion and related work are given in Section 5. Finally the paper is concluded and future work is outlined in Section 6.

2. A GENERAL NEGOTIATION MODEL

Before laying out our learning model, we give a brief description of a time dependent, bilateral single-issue negotiation model, which is widely used in many applications. Let i ($i \in \{b, s\}$) represent a negotiator, i.e., b for a buyer agent and s for a seller agent. Both agents have an initial price IP_i and reserve price RP_i for the negotiating issue. The interval $[IP_i, RP_i]$ indicates the range of all the possible agreements, and can be normalized in-between $[0, 1]$ using a utility function. In this paper, we choose the widely accepted linear utility function [4] shown in Equation 1:

$$u_i(p_i) = \frac{p_i - RP_i}{IP_i - RP_i} \quad i \in \{b, s\} \tag{1}$$

where p_i is the value of an offer in the range of $[IP_i, RP_i]$.

In time dependent tactic, agent i concedes its utility $u_i(t)$ under the time constraint. At the beginning of negotiation, agent i has its highest utility of 1 for the *initial price*. As the negotiation proceeds on, the utility $u_i(t)$ decreases according to a family of polynomial functions [4] given by Equation 2.

$$u_i(t) = 1 - (\frac{t}{T_i})^{\beta} \quad i \in \{b, s\} \tag{2}$$

where T_i is the deadline of agent i and β is the concession parameter. $\beta > 1$, $0 < \beta < 1$ and $\beta = 1$ represent three concession strategies called *Conceder, Boulware, Linear*, respectively, signifying different concession rates in a negotiation process.

When β is settled, the utility $u_i(t)$ can be computed during the negotiation. As a result, the agent can give a counter offer at time t according to the following offer generating equation [4].

$$Offer_i(t) = RP_i + u_i(t)(IP_i - RP_i) \quad i \in \{b, s\} \tag{3}$$

Combining Equation 2 and 3, the offer generating Equation 3 is rewritten as Equation 4.

$$Offer_i(t) = IP_i + (RP_i - IP_i)(\frac{t}{T_i})^{\beta} \tag{4}$$

In a non-learning negotiation setting, once an agent sets the value of β, the agent will keep this value unchange through the negotiation process, without any adaptation to the dynamic environment or the revelation of opponent's private information. However, if the agent can learn some useful information from the opponent during the negotiation, it will be able to adapt its original concession strategies and gain more benefits to produce good outcomes for negotiation. In the following section, we will present an adaptive negotiation model using regression analysis and Bayesian learning to enable agents to alter their concession strategies, thereby a better outcome will be obtained.

3. AN ADAPTIVE NEGOTIATION MODEL

In this section, an adaptive negotiation model is proposed. This model includes two parts which are a learning mechanism and an adaptive concession strategy. Each part will be introduced in detail by Subsections 3.2 and 3.3, respectively. In this paper hereinafter, the discussion is taken from the perspective of the buyer agent unless otherwise specified. However, such a discussion will not lose the generality of our model, i.e. a seller agent can also use our model to learn its opponent's behaviors.

3.1 Model Description

As we can see from Equations 4, the parameters of deadline and reserve price are two main factors dominating the negotiation process and outcomes. If agents can obtain the information about these two parameters from the opponent, a better strategy can be employed to increase agents' benefits and/or the negotiation efficiency.

Definition 1. Let x-axis represent negotiation time and y-axis represent the negotiation price. A *detecting region DetReg* is a rectangle in this two-dimensional area to present an estimation of the opponent's deadline and reserve price. This area is defined by a 4-tuple $DetReg = (T^l, T^h, P^l, P^h)$, where T^l, T^h are the estimated lower and upper boundary of the opponent's deadline, and P^l, P^h are the estimated lower and upper boundary of the opponent's reserve price.

As shown in Figure 1, the shadowy area indicates the detecting region for a buyer agent during a learning process based on Definition 1. T_b is buyer's deadline and t_b is the current time in negotiation. IP_b and RP_b represent the buyer's initial price and reserve price, respectively, and IP_s is the seller's initial price. Points appeared in the detection region of the figure will be explained in Definition 3. The lines shown in the figure will be explained in Subsection 3.2 during introducing the learning mechanism.

A buyer agent can initialize the value of each component of *DetReg* according to its estimation about seller's private

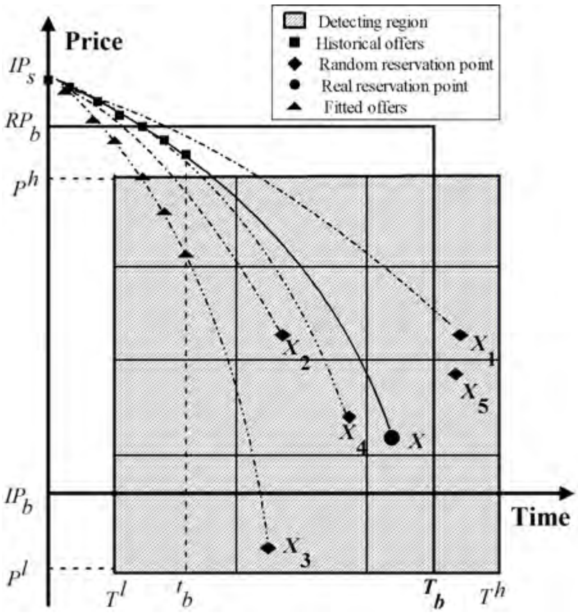


Figure 1: An example of demonstrating our learning process

information. The more precise the estimation is, the smaller the detecting region will be and the buyer can strive for a better result because more errors can be avoided when the buyer agent adapts its concession strategy based on this estimation.

After confirming the detecting region by the buyer agent, this region will be further divided into smaller areas according to $N = (N^t, N^p)$ in which N^t denotes that the detecting region is evenly divided into N^t columns on the x-axis (i.e. time values), and N^p stands for the row number on the y-axis (i.e. price values) in the detecting region. In this way, the detecting region can be divided into a number of smaller blocks, called *detecting cells*. The total number of detecting cells in a detecting region is represented by N_{all} and can be calculated by the formula $N_{all}=N^t \times N^p$. Fig. 1 exemplifies a scenario with $N = (3, 4)$ and there are totally 12 detecting cells in the whole region.

Definition 2. A *detecting cell* C_i ($i \in 1, 2, ..., N_{all}$) is a divided block in the detecting region, which can be denoted by a 4-tuple $C_i = (t_i^l, t_i^h, p_i^l, p_i^h)$ where t_i^l, t_i^h are the lower and higher boundaries of time in the cell and p_i^l, p_i^h are the lower and higher boundaries of price in the cell, respectively.

Definition 3. A *random reservation point* $X_i(t_i^x, p_i^x)$ is a randomly selected point in each cell C_i, where $t_i^l < t_i^x < t_i^h$ and $p_i^l < p_i^x < p_i^h$.

In Figure 1, points X_1, X_2, X_3, X_4 are several random reservation points in the detecting region and point X is the real reservation point of the opponent. The detecting cell is a region where seller's real reservation point X might be located. That means the real reservation point X might be out of the detecting region in real case. The buyer agent has some belief about the probability distribution of all the detecting cells. The probability of each cell signifies the likelihood that the opponent's real reservation point X might be located in this cell. This belief can be revised more precisely through learning from opponent's historical offers (see Subsection 3.2). Based on this learning result, the agent can adjust its concession strategy adaptively (see Subsection 3.3) to gain more profit over its opponent.

3.2 The Learning Mechanism

The purpose of this leaning mechanism is to let the agent revise its belief about the probability distribution of the cells in the detecting region. Because the agent has no knowledge about the opponent, it is hard to determine the precise location of the real reservation point. However the agent can observe its opponent's historical offers to renew the belief about the approximate range of the reservation point. This mechanism consists of two parts, a regress analysis and a Bayesian learning. In regression analysis, (1) an agent chooses a random reservation point in every detecting cell first, based on the belief that this point is the reservation point of the opponent; (2) the agent conducts the regression analysis for all random reservation points corresponding to all detecting cells, respectively; (3) the agent compares the fitted offers on each regression line with opponent's historical offers by the non-linear correlation. By this way, resemblance between the selected random reservation point and the opponent's real reservation point can be calculated. The bigger the non-linear correlation between two lines is, the more alike they will be. That also means that the randomly chosen reservation point has a bigger possibility to be the real reservation point. Then by using Bayesian learning, the agent's belief on the probability distribution will be dynamically updated at every step of the negotiation. The regression analysis and our Bayesian learning method are introduced in the following two subsections, respectively.

3.2.1 Regression Analysis

Before the leaning process, the buyer should initialize $DetReg$, N as well as the probability distribution in each detecting cell, which presents the likelihood that the seller's reservation point is in this cell. When the learning begins, the buyer can do the following steps sequentially.

Step 1: At round t_b, the buyer selects a random reservation point $X_i(t_i^x, p_i^x)$ in each cell C_i of the detecting region;

Step 2: Using each point $X_i(t_i^x, p_i^x)$ chosen in Step 1, the buyer calculates the regression line l_i based on the seller's historical offers $O_{t_b} = \{p_0, p_1, ..., p_{t_b}\}$ until round t_b. Based on Equation 4, the following power regression function is generated to calculate the regression curve.

$$Offer_i(t) = p_0 + (p_i^x - p_0)(\frac{t}{t_i^x})^b \quad (5)$$

where p_0 is the *initial price* of seller. The regression coefficients b is the concession parameter β in the utility function in Equation 4. Then we can calculate coefficient b based on seller's historical offers O_{t_b} by Equation 6 as proposed in [9].

$$b = \frac{\sum_{i=1}^{t_b} t_i^* p_i^*}{\sum_{i=1}^{t_b} t_i^{*2}} \quad (6)$$

where $p_i^* = \ln \frac{p_0 - p_i}{p_0 - p_i^x}$, $t^* = \ln \frac{t}{t_i^x}$. In Figure 1, the solid line is the curve of the seller's historical offers while the dashed line is the regression curve based on each random reservation point.

Step 3: Based on the calculated regression line l_i given by Equation 5 and 6, the buyer can calculate the fitted offers $\hat{O}_{t_b} = \{\hat{p}_0, \hat{p}_1, ..., \hat{p}_{t_b}\}$ at each round.

Step 4: The buyer calculates the non-linear correlation between seller's historical offers O_{t_b} and the fitted offers $\hat{O_{t_b}}$. The coefficient of nonlinear correlation γ can be calculated by Equation 7.

$$\gamma = \frac{\sum_{i=1}^{t_b}(p_i - \bar{p})(\hat{p}_i - \bar{\hat{p}})}{\sqrt{\sum_{i=1}^{t_b}(p_i - \bar{p})^2 \sum_{i=1}^{n}(\hat{p}_i - \bar{\hat{p}})^2}} \tag{7}$$

where $\bar{\hat{p}}$ is the average value of all the fitted offers till time t_b and $\bar{p}$ represents the average value of all the historical offers of the seller. The non-linear correlation γ, where $(0 \le \gamma \le 1)$, is a parameter reflecting the non-linear similarity between the fitted offers and the historical offers, which can be used as a criterion to evaluate the resemblance between the random reservation point X_i and seller's real reservation point X. This is an important parameter to be used in Bayesian learning for the belief updating as described in the following section.

3.2.2 *Bayesian Learning*

In general, Bayesian learning can be used when an agent has a set of hypotheses about its opponent's information. The belief about the probability distribution of these hypotheses can be revised through a posterior probability by observing the outcome of its opponent. In our model, we define the hypothesis space as $H_i, (i \in 1, 2, 3, ..., N_{all})$, where N_{all} is the total cell number in the detecting region. Each hypothesis H_i stands for the assumption that seller's reservation point X is in cell C_i. The prior probability distribution, denoted by $P(H_i), (i \in 1, 2, 3...N_{all})$, signifies the agent's belief about the hypothesis, that is, how likely the hypothesis fits the real situation. At first, the agent can initialize the probability distribution of the hypotheses based on some public information if available, otherwise a uniform distribution $P(H_i) = 1/N_{all}$ is assigned.

During each round of negotiation t_b, the probability of each hypothesis can be altered by the Bayesian updating rule given in Equation 8.

$$P(H_i|O) = \frac{P(H_i)P(O|H_i)}{\sum_{k=1}^{N_{all}} P(O|H_k)P(H_k)} \tag{8}$$

where the conditional probability $P(O|H_i)$ represents the likelihood that outcome O might happen based on hypothesis H_i. In our learning model, the agent has no information about its opponent, thus the observed outcome O is opponent's historical offers $O_{t_b} = \{p_0, p_1, ..., p_{t_b}\}$. The conditional probability $P(O|H_i)$ thereby means how likely seller's historical offer O_{t_b} can happen based on the hypothesis H_i that seller's real reservation point X is in cell C_i. The posterior probability $P(H_i|O)$ is a renewed belief based on the observed outcome O and at next round, the agent will update the prior probability $P(H_i)$ using the posterior probability $P(H_i|O)$, thus a more precise estimation is achieved.

To let the Bayesian learning rule work, the most critical problem is how to obtain the conditional probability $P(O|H_i)$. Most approaches using Bayesian learning method usually require a priori knowledge as the conditional probability, such as [12]. However, our learning model does not require any priori knowledge about the opponent and works based only on the historical offers received until t_b from the opponent. By comparing the fitted points $\hat{O}_{t_b}$ on the regression line based on each random reservation point X_i with the historical offers O_{t_b}, the conditional probability $P(O|H_i)$ is obtained. The more consistent the fitted offers are with opponent's historical offers, the higher the conditional probability $P(O|H_i)$ it will be. As showed at Step 3 in Subsection 3.2.1, the difference between the regression curve and opponent's bidding sequence can be indicated by the non-linear correlation coefficient γ. Thus, we can use the value of γ as the conditional probability.

The learning approach will increase the probability of a hypothesis when the random reservation point selected in the detecting cell is most consistent with the real reservation point of the opponent. However, in some cases, it is possible that the learning may have errors. As seen in Figure 1, compared with point X_5, point X_4 has a higher non-linear correlation with the real reservation point X, but point X_4 and X are not in the same detecting cell. As a result, the hypothesis that the real reservation point X belongs to the cell, where point X_4 is located, has a higher probability. Nevertheless, we claim that this situation does not affect the learning effectiveness based on the following two considerations. Firstly, although in certain circumstances, using the non-linear correlation to calculate the difference between the regression line and the real bidding sequence does not necessarily reveal the real situation, the error will be eased through Bayesian learning from a probabilistic point of view. Secondly, even the error exists, the learning approach still works because we only need to find an approximate range of the reservation point, not the precise value of opponent's reservation point. In some cases, the real reservation point X might not be located in the whole detecting region, but those cells which are closer to point X will still have a higher probability compared with other cells.

Another issue that should be taken into account is the learning rate and efficiency. At the early stage of leaning, the hypotheses space can be quite large depending on the value of $DetReg$ and N (recall Subsection 3.1). It is time consuming to keep all the hypotheses in the searching space. Some hypotheses can be precluded from the hypotheses space when the current time and opponent's bidding value have surpassed the detecting cell boundary. For example, for a cell $C_i = [t_i^l, t_i^h, p_i^l, p_i^h]$, if current negotiation time $t_b > t_i^h$, the hypothesis based on this cell is meaningless because the negotiation process has already proved it false.

3.3 The Adaptive Concession Strategy

Through regression analysis and Bayesian learning stated above, a more precise estimation of the opponent's reservation point is derived, represented by the renewed belief of the probability distribution of the hypothesis H_i. Now, the agent needs to take an action to give a counter offer based on this new belief, i.e. which concession strategy to take and how strong it should be in terms of a value of the concession parameter β.

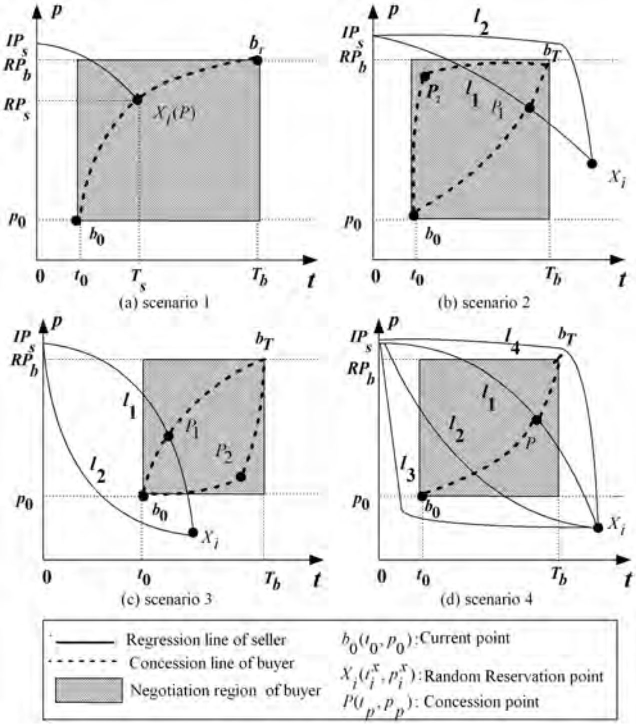


Figure 2: Four Scenarios of Concession Strategy

3.3.1 The Optimal Concession strategy

There are four scenarios according to different location of the random reservation point. As we believe that the agent is rational, it always strives for a highest utility of its own regardless of its opponent fully. Therefore, in each scenario, the buyer needs to adopt different concession strategies to maximize its expected utility as depicted in Figure 2. In Figure 2, point $b_0(t_0, p_0)$ is buyer's current offer at time t_0, point $b_T(T_b, RP_b)$ is the buyer's reservation offer at deadline T_b, and point $X_i(t_i^x, p_i^x)$ is the random reservation point of seller. Then the buyer needs to find another point $P(t_p, p_p)$, which is called a concession point, in its negotiation region to set the concession strategy and the value of β.

- **Scenario 1:** $(t_i^x < T_b)$ and $(p_i^x > p_0)$.

  In this scenario, the random reservation point X_i is in the buyer's negotiation region. Because the buyer agent is rational, it will always try to gain the maximal utility itself. If the buyer knows that the seller will quit the negotiation at point X_i (i.e., the deadline of the seller t_i^x is shorter than its deadline T_b), the optimal concession strategy for the buyer is to set his bidding price to p_i^x at time t_i^x. Otherwise, if the buyer gives more concession, it cannot achieve the maximal utility after finishing negotiation. On the contrary, a less concession may result in a failure of the negotiation. As illustrated in Figure 2(a), the random reservation point X_i is set to be the concession point P in this case and the dashed line crossing point X_i is the concession line of the buyer.

- **Scenario 2:** $(t_i^x >= T_b)$ and $(p_i^x >= p_0)$.

  In this scenario, random reservation point X_i is out of the buyer's negotiation region. There are two cases in this scenario according to the different regression lines of the seller. As can be seen in Figure 2(b), in the first case, regression line l_1 traverses the buyer's negotiation region while l_2 does not. In the same way of analyzing in *Scenario 1*, buyer's optimal concession line for l_1 is to pass through the intersection point of the line l_1 and the right boundary of the buyer's negotiation region. Considering that the buyer should give out its reserve price at deadline T_b, for simplicity, we let the buyer's concession line cross the concession point P_1 on the regression line one step ahead of the deadline T_b (i.e., $T_b - 1$) such that a concrete value of the concession parameter β can be computed. As for the second case, the regression curve l_2 has no intersection with the buyer's negotiation region, which means even the buyer concedes, the negotiation based on this random reservation point is doomed to fail. Nevertheless, the buyer will spare no efforts to reverse this unfavorable situation. So, it will give the reserve price at next round $(t_0 + 1)$. To compute a value of β, we choose a variable $\phi_{max}(0 < \phi_{max} < 1)$ which is quite close to 1. The concession point P_2 in this case is set to be $P_2(b_0 + 1, \phi_{max} \cdot RP_b)$ so as to make the price at next round close to the reserve price of RP_b and finally to give out the reserve price RP_b at the deadline.

- **Scenario 3:** $(t_i^x < T_b)$ and $(p_i^x < p_0)$.

  There are also two cases in this scenario, which can be signified by l_1 and l_2 shown in Figure 2(c). As for case 1, the optimal strategy of the buyer is to cross the intersection of l_1 and bottom line of the buyer's negotiation region. To compute a value of β, we set the concession point P_1 be the point one step earlier than the intersection point on the regression line l_1. As for case 2, the line l_2 does not go through the buyer's negotiation region. In this case, the optimal strategy for the buyer is to keep its price unchanged until $T_b - 1$ and then gives its reserve price at the deadline. To compute the value of β, we can set the price at concession point P_2 very close to current price p_0. Similarly, a variable $\phi_{min}(0 < \phi_{min} < 1)$, which is quite close to 0, can be chosen to set the price at next round to $(1 + \phi_{min}) \cdot p_0$ such that this price will keep almost the same as the current price p_0.

- **Scenario 4:** $(t_i^x >= T_b)$ and $(p_i^x >= p_0)$.

  This scenario, a combination of the former Scenarios 2 and 3, is the most complicated case of all. Each line of l_1, l_2, l_3 and l_4 can be analyzed in the same way stated before. In Figure 2(d), we depict the concession line based on l_1 as an example.

3.3.2 The Combined Mechanism

We have given out all possible situations of the random reservation points and the corresponding optimal concession strategies that the buyer can adopt to increase its utility as well as to avoid the failure of negotiation to its best. Because the buyer still uses the family of polynomial functions to concede, the counteroffer from point $b_0(t_0, p_0)$ can be generated by Equation 9 based on Equation 4.

$$Offer_b(t) = p_0 + (RP_b - p_0)\left(\frac{t - t_0}{T_b - t_0}\right)^{\beta} \quad (t > t_0) \qquad (9)$$

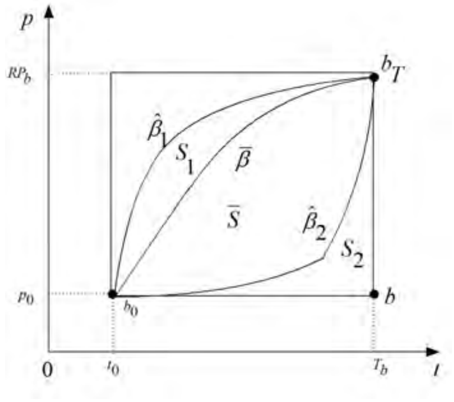


Figure 3: Combination of the parameter β

Using this equation, we can guarantee that at its deadline T_b, the buyer will give the reserve price RP_b. At current time t_0, the buyer's offer is p_0 and the buyer concedes in the form of polynomial function. Then given the concession point $P(t_p, p_p)$ in its negotiation region, a new value of parameter $\hat{\beta}$ can be calculated as follows.

$$\hat{\beta} = \log_{\frac{t_p - t_0}{T_b - t_0}} \frac{p_0 - p_p}{p_0 - RP_b} \quad (t_0 < t_p < T_b) \tag{10}$$

We have calculated all the concession values $\hat{\beta}$ for each valid random reservation point in the detecting region, with a probability distribution $P(H_i) = \{p(H_1), p(H_2), ...p(H_n)\}$ over these values deprived from the regression analysis and Bayesian learning. Now comes to the problem of how to combine all the estimated value of $\hat{\beta}$ to an overall value. Let $\hat{\beta}_i$ $(i \in \{1, 2, ...n\})$ be the estimated concession value calculated from the concession point based on the random reservation point in cell C_i. $P(H_i)$ is the probability of the $\hat{\beta}_i$, presenting the weighting proportion of the corresponding $\hat{\beta}_i$ in all the concession values. The value of $\hat{\beta}_i$ signifies the concession degree of the agent, which can be represented by the area between the concession line and the time axis, which is called *concession area*. As can be seen from Figure 3, the concession area of $\hat{\beta}_1$ is S_1, which can be denoted by $S_{b_0 \hat{\beta}_1 b_T b}$. Let S_i be the concession area of $\hat{\beta}_i$ and let the concession area of the overall concession parameter $\bar{\beta}$ be $\bar{S}$. Based on Equation 9, we can have the following equation.

$$\bar{S} = \int_{t_0}^{T_b} [p_0 + (RP_b - P_0)(\frac{t - t_0}{T_b - t_0})^{\bar{\beta}}] d_t \tag{11}$$

$$\sum_{i=1}^{n} P(H_i) S_i = \sum_{i=1}^{n} P(H_i) \int_{t_0}^{T_b} [p_0 + (RP_b - P_0)(\frac{t - t_0}{T_b - t_0})^{\hat{\beta}}] d_t \tag{12}$$

because,

$$\bar{S} = \sum_{i=1}^{n} P(H_i) S_i \tag{13}$$

we can get the overall concession parameter $\bar{\beta}$ as follows:

$$\bar{\beta} = \frac{1}{\sum_{i=1}^{n} \frac{P(H_i)}{1 + \hat{\beta}_i}} - 1 \tag{14}$$

Then the buyer can set its concession parameter as $\bar{\beta}$ to give counter offer based on Equation 9 at every step of the negotiation. Each $\hat{\beta}_i$ is changing at each step according to the randomly selected reservation point and the corresponding $P(H_i)$ is revised by Bayesian learning throughout the negotiation process. So the concession parameter $\bar{\beta}$ adopted by the buyer at each step is totally different, making the negotiation an adaptive process in the point view of the buyer.

4. EXPERIMENT

In this section, the empirical experimental results are displayed to demonstrated the good performance of our model.

4.1 Experimental Setting

In the experiment, a buyer and a seller negotiate over the price ranged in-between \$0 $\sim$ \$100. In order to simplify the comparison process, we set the buyer agent's initial price to \$0 and the seller agent's initial price to \$100. The buyer's reserve price is randomly selected in-between \$50 $\sim$ \$100 and seller's reserve price is randomly selected in-between \$0 $\sim$ \$50. Such a setting will ensure the agreement zone between the two agents always exists. Our agents' deadlines are randomly selected in-between $[20, 40]$, and the concession strategies are randomly selected in-between $[0.5, 2]$. The negotiation parameter initialization is showed in Table 1.

Table 1: Parameter initialization

Agent	IP_i	RP_i	T_i	β_i
Buyer (i=b)	\$0	[\$50,\$100]	[20,40]	[0.5,2]
Seller (i=s)	\$100	[\$0,\$50]	[20,40]	[0.5,2]

To provide a benchmark we compare our negotiation model with the NDF model. In the NDF model, both agents randomly initialize their negotiation parameters according to Table 1, and keep these parameters unchanged during the negotiation process. On the contrary, in our model, the buyer agent will learn how to adjust its concession strategy adaptively to reach a better negotiation outcome. To use the learning mechanism, the buyer initializes the detecting region as $DetReg = (0, 1.5T_b, 0, RP_b)$, $\phi_{min} = 0.01$, $\phi_{max} = 0.99$. We outline four cases according to the different numbers of detecting cells (see Table 2).

Table 2: Four scenarios of different detecting cell numbers

Case	N^t	N^p	N_{all}
1	4	4	16
2	8	8	48
3	16	16	256
4	20	20	400

4.2 Results And Analysis

As our model depends on the regression analysis which may yield errors as stated before, we do not expect the learning result to be completely precise. Further more, many variables affect the learning process such as the number of detecting cells. The objective of this experiment is therefore carried out to analyze the overall performance of this learning approach considering this uncertainty and error.

We run 100 episodes for each case to show the generality and robustness of our model. The results of this experiment are presented in Figure 4. The x-axis indicates the four cases and the y-axis indicates the average utility of the buyer in

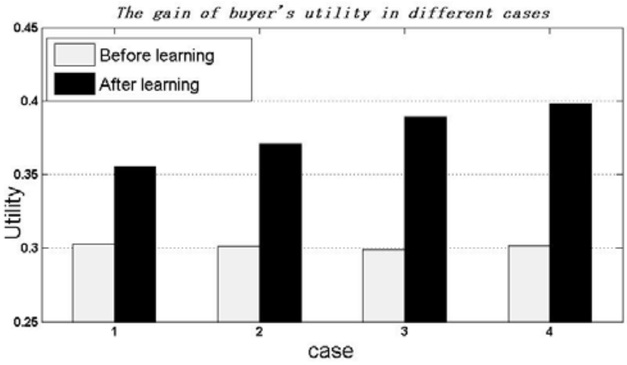


Figure 4: The average utility in different cases

each case. We use solid bars to represent the buyer's average utility gained by using our model while the empty bars to represent the buyer's average utility gained by NDF model. We can see from Figure 4, the solid bars are higher than the empty bars in all cases, and gradually increase as the number of the detecting cells increase. Such experimental results indicate that: (1)using our learning mechanism and the adaptive concession strategy will result in a higher utility than the static concession strategy; and (2) as the total number of detecting cells increases, the agent has a more precise estimation of the opponent's reservation point, thus can result in a higher utility.

In order to illustrate the dynamic adaptation of the concession parameter β, we give out the whole negotiation process to show how the buyer agent changes its concession strategy adaptively. We select three scenarios with the negotiation parameters as follows:

- Scenario 1 ($1 < \beta < 2$): RP_b = \$69.58, RP_s = \$11.38,$T_b = 32$, $T_s = 36$, $\beta_b = \beta_s = 2.0$
- Scenario 2 ($\beta = 1$): RP_b = \$81.04, RP_s = \$17.82,$T_b = 35$, $T_s = 36$, $\beta_b = \beta_s = 1.0$
- Scenario 3 ($0 < \beta < 1$):RP_b = \$50.34, RP_s = \$11.38,$T_b = 30$, $T_s = 22$, $\beta_b = \beta_s = 0.5$

Figures in 5(a) give the negotiation process between both agents before & after learning. Figures in 5(b) show the adaption of the the buyer's concession parameter β. In Scenario 1, the seller adopts the *Boulware* concession strategy. Before learning, the negotiation ends at \$49.53 and both agents' concession strategies keep unchanged through the negotiation process. After learning, the buyer agent adjusts its concession strategy adaptively in terms of parameter β and the agreement price is reduced to \$38.46, which is a better result than that of before learning for the buyer agent. In Scenario 2, the seller uses the *Linear* concession strategy. Before learning, the negotiation ends at \$48.57 and after learning, the buyer can have a better agreement at \$22.34. In Scenario 3, the seller uses the *Conceder* concession strategy. Before learning, the final agreement is \$33.38 and after learning this value decreases to \$20.58. According to these experimental results from the three scenarios, we can conclude that, through learning of the opponent's historical offers, the agent employing our negotiation model can effectively adapt its concession strategy so as to increase its negotiation outcome. Our negotiation model is robust when the opponent employs different concession strategies.

In this section, we illustrate the experimental results of our negotiation model and compare the results with the NDF's. The experimental results indicate that our negotiation model can dynamically adapt a negotiation agent's concession strategy and significantly increase a negotiation agent's utility through the learning of the opponent's historical offers.

5. RELATED WORK

Although incorporating learning in agent negotiation is a relatively new research topic, many approaches, models and mechanisms have been developed in recent years to solve different issues in this topic [6] [10] [5] [11]. In this section, we discuss several related works and compare them with our model proposed in this paper.

Zeng and Sycara proposed a Bayesian approach to learn the reserve price of an opponent under negotiation setting [12]. In their approach, a sequential decision making model called *Bazaar* was proposed to model beliefs of the opponent's reservation point. Our model differs from their approach in two ways. (1) *Bazaar* can only learn the reserve price of the opponent while our model can learn both opponent's price and deadline, and (2) *Bazzar* requests priori knowledge about the potential distribution of of the opponent's reserve price while our model has no this request.

Bzostowski and Kowalczyk [3] presented an approach for modeling behaviors of negotiators and predictive decision-making. Both their approach and our work used the similar method in term of adaptive concession strategy based only on the historical offers in the current negotiation. However, their approach focuses more on the analysis of the differences between adjacent offers from the opponents, and will become ineffective when these differences are not significant. Our approach employs the regression analysis and will not be affected by the variance of adjacent offers.

Narayanan and Jennings proposed a novel adaptive negotiation model considering the dynamism in E-commerce settings [7]. Their model manages negotiation process as a Markov Decision Process(MDP) and uses a value iteration algorithm to acquire optimal policies to adopt different concession strategies. However, their method can only determine the adaptive action to choose a concession strategy and cannot produce a precise concession value while our model can provide constructive guidance to the agent to dynamically adaptive its behaviors including both strategies and concession values.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed an adaptive bilateral negotiation model based on Bayesian learning. This model includes a learning mechanism and an adaptive concession strategy. Through Bayesian learning, an agent's belief about the opponent's reserve price can be revised dynamically during negotiation by comparing the fitted offers based on regression analysis. The proposed model can enable an agent to adapt its concession strategies according to the updated probability distribution in a predicting region. The experimental results demonstrate the good performance of our model by comparison with NDF model.

The future works are to test our model in more complex scenarios and extend it to a multi-issue negotiation environment by considering more factor which can affect negotiation

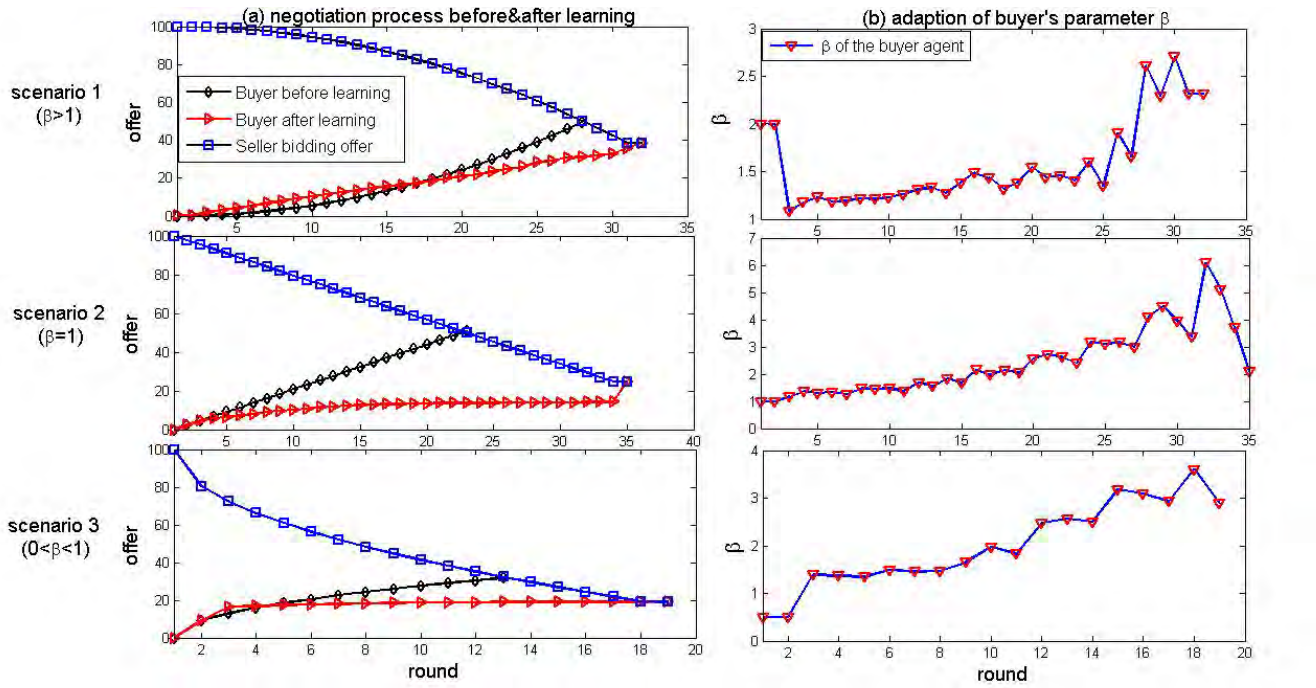


Figure 5: The adaptive concession process

process so as to produce win-win outcomes for both negotiation parties.

7. REFERENCES

[1] R. Aydogan and P. Yolum. Learning consumer preferences using semantic similarity. In *Proceedings of Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1293–1300. Citeseer, 2007.

[2] R. Aydogan and P. Yolum. Ontology-based learning for negotiation. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, pages 177–184. IEEE Computer Society, 2009.

[3] J. Bzostowski and R. Kowalczyk. Predicting partner's behavior in agent negotiation. In *Proceedings of 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 355–361, 2006.

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

[5] K. Hindriks and D. Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *In Proceedings of the 7th International Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2008)*, pages 331–338, 2008.

[6] C. Hou. Modelling agents behaviour in automated negotiation. Technical Report KMI-TR-144, Knowledge Media Institute, The Open University, Milton Keynes,UK, May 2004.

[7] V. Narayanan and N. Jennings. An adaptive bilateral negotiation model for e-commerce settings. In *Seventh IEEE International Conference on E-Commerce Technology*, pages 34–41. IEEE, 2005.

[8] V. Narayanan and N. Jennings. Learning to negotiate optimally in non-stationary environments. *Cooperative Information Agents*, pages 288–300, 2006.

[9] F. Ren and M. Zhang. Predicting partners' behaviors in negotiation by using regression analysis. In *Proceedings of the 2nd international conference on Knowledge science, engineering and management*, pages 165–176. Springer-Verlag, 2007.

[10] F. Ren, M. Zhang, and K. Sim. Adaptive conceding strategies for automated trading agents in dynamic, open markets. *Decision Support Systems*, 46(3):704–716, 2009.

[11] V. Robu and H. La Poutré. Learning the structure of utility graphs used in multi-issue negotiation through collaborative filtering. *Multi-Agent Systems for Society*, pages 192–206, 2009.

[12] D. Zeng and K. Sycara. Benefits of learning in negotiation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 36–42. Citeseer, 1997.

Acceptance Conditions in Automated Negotiation

Tim Baarslag
Man-Machine Interaction Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
T.Baarslag@tudelft.nl

Koen Hindriks
Man-Machine Interaction Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
K.V.Hindriks@tudelft.nl

Catholijn Jonker
Man-Machine Interaction Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
C.M.Jonker@tudelft.nl

ABSTRACT

In every negotiation with a deadline, one of the negotiating parties has to accept an offer to avoid a break off. A break off is usually an undesirable outcome for both parties, therefore it is important that a negotiator employs a proficient mechanism to decide under which conditions to accept. When designing such conditions one is faced with the acceptance dilemma: accepting the current offer may be suboptimal, as better offers may still be presented. On the other hand, accepting too late may prevent an agreement from being reached, resulting in a break off with no gain for either party.

Motivated by the challenges of bilateral negotiations between automated agents and by the results and insights of the automated negotiating agents competition (ANAC), we classify and compare state-of-the-art generic acceptance conditions. We focus on *decoupled* acceptance conditions, i.e. conditions that do not depend on the bidding strategy that is used. We performed extensive experiments to compare the performance of acceptance conditions in combination with a broad range of bidding strategies and negotiation domains. Furthermore we propose new acceptance conditions and we demonstrate that they outperform the other conditions that we study. In particular, it is shown that they outperform the standard acceptance condition of comparing the current offer with the offer the agent is ready to send out. We also provide insight in to why some conditions work better than others and investigate correlations between the properties of the negotiation environment and the efficacy of acceptance conditions.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*intelligent agents, multi-agent systems*

General Terms

Algorithms, Bargaining, Experimentation, Negotiation

Keywords

Automated bilateral negotiation, acceptance criteria, acceptance conditions

1. INTRODUCTION

Negotiation is an important process to reach trade agreements, and to form alliances or resolve conflicts. The field of negotiation originates from various disciplines including artificial intelligence, economics, social science, and game theory (e.g., [2, 16, 20]). The strategic–negotiation model has a wide range of applications, such as resource and task allocation mechanisms, conflict resolution mechanisms, and decentralized information services [16].

A number of successful negotiation strategies have already been established both in literature and in implementations [6, 7, 12, 13, 19]. And more recently, in 2010 seven new negotiation strategies were created to participate in the first automated negotiating agents competition (ANAC 2010) [3] in conjunction with the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-10). During post tournament analysis of the results, it became apparent that different agent implementations use various conditions to decide when to accept an offer. In every negotiation with a deadline, one of the negotiating parties has to accept an offer to avoid a break off. Therefore, it is important for every negotiator to employ a mechanism to decide under which conditions to accept. However, designing a proper acceptance condition is a difficult task: accepting too late may result in the break off of a negotiation, while accepting too early may result in suboptimal agreements.

The importance of choosing an appropriate acceptance condition is confirmed by the results of ANAC 2010 (see Table 1). Agents with simple acceptance criteria were ranked at the bottom, while the more sophisticated time- and utility-based criteria obtained a higher score. For instance, the low ranking of *Agent Smith* was due to a mistake in the implementation of the acceptance condition [27].

Despite its importance, the theory and practice of acceptance conditions has not yet received much attention. The goal of this paper is to classify current approaches and to compare acceptance conditions in an experimental setting. Thus in this paper we will concentrate on the final part of the negotiation process: the acceptation of an offer. We focus on decoupled acceptance conditions: i.e., generic acceptance conditions that can be used in conjunction with an arbitrary bidding strategy.

Our contribution is fourfold:

1. We give an overview and provide a categorization of current decoupled acceptance conditions.
2. We introduce a formal negotiation model that supports the use of arbitrary acceptance conditions.
3. We compare a selection of current generic acceptance conditions and evaluate them in an experimental setting.

Rank	Agent	Acceptance condition
1	*Agent K*	Time and utility based
2	*Yushu*	Time and utility based
3	*Nozomi*	Time and utility based
4	*IAMhaggler*	Utility based only
5	*FSEGA*	Utility based only
6	*IAMcrazyHaggler*	Utility based only
7	*Agent Smith*	Time and utility based

Table 1: An overview of the rank and acceptance conditions of every agent in ANAC 2010.

4. We propose new acceptance conditions and test them against established acceptance conditions, using varying types of bidding techniques.

The remainder of this paper is organized as follows. Section 2 defines the model of negotiation that we employ and provides an overview of current acceptance conditions. In Section 3, we also consider combinations of acceptance conditions. Section 4 discusses our experimental setup and results, which demonstrate that some combinations outperform traditional acceptance conditions. Finally, Section 5 outlines our conclusions and our plans for further research on acceptance strategies.

2. ACCEPTANCE CONDITIONS IN NEGOTIATION

This paper focuses on acceptance conditions (also called acceptance criteria) that are decoupled: i.e. generic acceptance conditions that are not tied to a specific agent implementation and hence can be used in conjunction with an arbitrary bidding strategy. We first describe a general negotiation model which fits current decoupled acceptance conditions. We have surveyed existing negotiation agents to examine the acceptance criteria that they employ. We then categorize them according to the input that they use in their decision making process.

2.1 Negotiation Model

We consider *bilateral* negotiations, i.e. a negotiation between two parties or agents A and B. The agents negotiate over *issues* that are part of a negotiation *domain*, and every issue has an associated range of alternatives or *values*. A negotiation outcome consists of a mapping of every issue to a value, and the set Ω of all possible outcomes is called the *outcome space*. The outcome space is common knowledge to the negotiating parties and stays fixed during a single negotiation session.

We further assume that both parties have certain preferences prescribed by a *preference profile* over Ω. These preferences can be modeled by means of a utility function U, which maps a possible outcome $\omega \in \Omega$ to a real-valued number in the range $[0,1]$. In contrast to the outcome space, the preference profile of the agents is private information.

Finally, the interaction between negotiating parties is regulated by a *negotiation protocol* that defines the rules of how and when proposals can be exchanged. We use the alternating-offers protocol [23] for bilateral negotiation, in which the negotiating parties exchange offers in turns.

As in [26], we assume a common global time, represented here by $\mathcal{T} = [0,1]$. We supplement the alternating-offers protocol with a deadline $t = 1$, at which moment both agent receive utility 0. This is the same setup as [8], with the exception that issues are not necessarily real-valued and both agents have the same deadline equal to $t = 1$. We represent by $x^t_{A\to B}$ the negotiation outcome proposed by agent A to agent B at time t. A *negotiation thread* (cf. [6, 26]) between two agents A and B at time $t \in \mathcal{T}$ is defined as a finite sequence

$$H^t_{A\leftrightarrow B} := \left(x^{t_1}_{p_1\to p_2}, x^{t_2}_{p_2\to p_3}, x^{t_3}_{p_3\to p_4}, \ldots, x^{t_n}_{p_n\to p_{n+1}}\right),$$

where

1. $t_k \le t_l$ for $k \le l$, the offers are ordered over time $\mathcal{T}$,
2. $p_k = p_{k+2} \in \{A, B\}$ for all k, the offers are alternating between the agents,
3. All t_i represent instances of time $\mathcal{T}$, with $t_n \le t$,
4. $x^{t_k}_{p_k\to p_{k+1}} \in \Omega$ for $k \in \{1,\ldots,n\}$, the agents exchange complete offers.

Additionally, the last element of $H^t_{A\leftrightarrow B}$ may be equal to one of the particles $\{Accept, End\}$. We will say a negotiation thread is *active* if this is not the case.

When agent A receives an offer $x^t_{B\to A}$ from agent B sent at time t, it has to decide at a later time $t' > t$ whether to accept the offer, or to send a counter-offer $x^{t'}_{A\to B}$. Given a negotiation thread $H^t_{A\leftrightarrow B}$ between agents A and B, we can formally express the action performed by A with an *action function* X_A:

$$X_A(t', x^t_{B\to A}) = \begin{cases} End & \text{if } t' \ge 1 \\ Accept & \text{if } \mathbf{AC}_A(t', x^{t'}_{A\to B}, H^t_{A\leftrightarrow B}) \\ x^{t'}_{A\to B} & \text{otherwise} \end{cases}$$

Note that we extend the setting of [8, 26] by introducing the *acceptance condition* $\mathbf{AC}_A$ of an agent A. This model enables us to study arbitrary decoupled acceptance conditions. $\mathbf{AC}_A$ that takes as input

$$\mathcal{I} = (t', x^{t'}_{A\to B}, H^t_{A\leftrightarrow B}),$$

the tuple containing the current time t', the offer $x^{t'}_{A\to B}$ that the agent considers as a bid (in line with the bidding strategy the agent uses), and the ongoing negotiation thread $H^t_{B\leftrightarrow A}$.

The resulting action given by the function $X_A(t', x^t_{B\to A})$ is used to extend the current negotiation thread between the two agents. If the agent does not accept the current offer, and the deadline has not been reached, it will prepare a counter-offer $x^{t'}_{A\to B}$ by using a bidding strategy or *tactic* to generate new values for the negotiable issues. Tactics can take many forms, e.g. time-dependent, resource dependent, imitative, and so on [26]. In our setup we will consider the tactics as given and try to optimize the accompanying acceptance conditions.

2.2 Acceptance Criteria

Let an active negotiation thread

$$H^t_{A\leftrightarrow B} = \left(x^{t_1}_{p_1\to p_2}, x^{t_2}_{p_2\to p_3}, \ldots, x^{t_{n-1}}_{A\to B}, x^{t_n}_{B\to A}\right),$$

be given at time $t' > t = t_n$, so that it is agent A's turn to perform an action.

As outlined in our negotiation model, the action function X_A of an agent A uses an acceptance condition $\mathbf{AC}_A(\mathcal{I})$ to decide whether to accept. In practice, most agents do not use the full negotiation thread to decide whether it is time to accept. For instance many agent implementations, such as [9, 8, 26], use the following implementation of $\mathbf{AC}_A(\mathcal{I})$:

$$\mathbf{AC}_A(t', x^{t'}_{A\to B}, H^t_{A\leftrightarrow B}) \iff U_A(x^t_{B\to A}) \geq U_A(x^{t'}_{A\to B}).$$

That is, A will accept when the utility U_A for the opponent's last offer at time t is greater than the value of the offer agent A is ready to send out at time t'. The acceptance condition above depends on the agent's upcoming offer $x^{t'}_{A\to B}$. For $\alpha, \beta \in \mathbb{R}$ this may be generalized as follows:

$$\mathbf{AC}^{\mathcal{I}}_{\text{next}}(\alpha, \beta) \overset{\text{def}}{\iff} \alpha \cdot U_A(x^t_{B\to A}) + \beta \geq U_A(x^{t'}_{A\to B}).$$

We can view α as the scale factor by which we multiply the opponent's bid, while β specifies the minimal 'utility gap' [13] that is sufficient to accept.

Analogously, we have acceptance conditions that rely on the agent's *previous* offer $x^{t_{n-1}}_{A\to B}$:

$$\mathbf{AC}^{\mathcal{I}}_{\text{prev}}(\alpha, \beta) \overset{\text{def}}{\iff} \alpha \cdot U_A(x^t_{B\to A}) + \beta \geq U_A(x^{t_{n-1}}_{A\to B}).$$

Note that this acceptance condition does not take into account the time that is left in the negotiation, nor any offers made previous to time t. Other acceptance conditions may rely on other measures, such as the remaining negotiation time or the utility of our previous offer. For example, there is a very simple acceptance criterion that only compares the opponent's offer with a constant α:

$$\mathbf{AC}^{\mathcal{I}}_{\text{const}}(\alpha) \overset{\text{def}}{\iff} U_A(x^t_{B\to A}) \geq \alpha.$$

Last but not least, instead of considering utility agents may employ a time-based condition to accept after a certain amount of time $T \in \mathcal{T}$ has passed:

$$\mathbf{AC}^{\mathcal{I}}_{\text{time}}(T) \overset{\text{def}}{\iff} t' \geq T.$$

We will omit the superscript $\mathcal{I}$ when it is clear from the context. We will use these general acceptance conditions to classify existing acceptance mechanisms in the next section.

2.3 Existing Acceptance Conditions

We give a short overview of decoupled acceptance conditions used in literature and current agent implementations. We are primarily interested in acceptance conditions that are not specifically designed for a single agent. We do not claim the list below is complete; however it serves as a good starting point to categorize current decoupled acceptance conditions. We surveyed the entire pool of agents of ANAC 2010, including *Agent K* and *Nozomi* [25], *Yushu* [1], *IAM(crazy)Haggler* [5], *FSEGA* [24] and *Agent Smith* [27]. We also examined well-known agents from literature, such as the Trade-off agent [7], the Bayesian learning agent [11], *ABMP* [13], equilibrium strategies of [9], and time dependent negotiation strategies as defined in [22], i.e. the Boulware and Conceder tactics.

Listed in Table 2 is a selection of generic acceptance conditions found.

Table 2: A selection of existing decoupled acceptance conditions.

AC	α	β	Agent
$\mathbf{AC}_{\text{prev}}(\alpha, \beta)$	1.03	0	*FSEGA*, *Bayesian Agent*
	1	0	*Agent Smith*
	1.02	0	*IAM(crazy)Haggler*
	1	0.02	*ABMP*
$\mathbf{AC}_{\text{next}}(\alpha, \beta)$	1	0	*FSEGA*, Boulware, Conceder, Trade-off, Equilibrium strategies
	1.02	0	*IAM(crazy)Haggler*
	1.03	0	*Bayesian Agent*
$\mathbf{AC}_{\text{const}}(\alpha)$	1	-	*FSEGA*
	0.9	-	*Agent Smith*
	0.88	-	*IAM(crazy)Haggler*
	T		
$\mathbf{AC}_{\text{time}}(T)$	0.92	-	*Agent Smith*

Some agents also use logical combinations of different acceptance conditions at the same time. This explains why some agents are listed multiple times. For example, both *IAMHaggler* and *IAMcrazyHaggler* [4] accept precisely when

$$\mathbf{AC}_{\text{const}}(0.88) \vee \mathbf{AC}_{\text{next}}(1.02, 0) \vee \mathbf{AC}_{\text{prev}}(1.02, 0).$$

We will not focus on the many possible combinations of all acceptance conditions that may thus be obtained; we will study the basic acceptance conditions in isolation with varying parameters. However in addition to this we study a small selection of combinations in Section 3. We leave further combinations for future research.

As can be seen from Table 2, in our sample the most commonly used acceptance condition is $\mathbf{AC}_{\text{next}} = \mathbf{AC}_{\text{next}}(1, 0)$, which is the familiar condition of accepting when the opponent's last offer is better than the planned offer of the agent. The function $\beta \mapsto \mathbf{AC}_{\text{prev}}(1, \beta)$ can be viewed as an acceptance condition that accepts when the utility gap [13] between the parties is smaller than β. We denote this condition by $\mathbf{AC}_{\text{gap}}(\beta)$.

3. COMBINED ACCEPTANCE CONDITIONS

We define three acceptance conditions that are designed to perform well in conjunction with an arbitrary bidding strategy. This will incorporate all ideas behind the traditional acceptance conditions we have described so far. We will show in Section 4 that they work better than the majority of simple generic conditions listed in Table 2.

From a negotiation point of view, it makes sense to alter the behavior of the acceptance condition when time is running short. Many ANAC agents such as Yushu, Nozomi and FSEGA [1, 24, 25] split the negotiation into different intervals of time and apply different sub-strategies to each interval.

The basic idea behind combined acceptance conditions $\mathbf{AC}_{\text{combi}}$ is as follows. In case the bidding strategy plans to propose a deal that is worse than the opponent's offer, we have reached a consensus with our opponent and we accept the offer. But if there still exists a gap between our offer and time is short, the acceptance condition should wait for an offer that is not expected to improve in the remaining time. Thus $\mathbf{AC}_{\text{combi}}$ is designed to be a proper extension of $\mathbf{AC}_{\text{next}}$, with adaptive behavior based on recent bidding behavior near the deadline.

To define $\mathbf{AC}_{\text{combi}}$, suppose an active negotiation thread

$$H^t_{A\leftrightarrow B} = \left(x^{t_1}_{p_1\to p_2}, x^{t_2}_{p_2\to p_3}, \ldots, x^{t_{n-1}}_{A\to B}, x^{t_n}_{B\to A}\right),$$

is given at time $t' > t = t_n > \frac{1}{2}$ near the deadline, when it is agent A's turn. Note that there is $r = 1 - t'$ time remaining in the negotiation, which we will call the *remaining time window*. A good sample of what might be expected in the remaining time window consists of the bids that were exchanged during the previous time window $W = [t' - r, t'] \subseteq \mathcal{T}$ of the same size.

Let

$$H^W_{B\to A} = \left\{x^s_{B\to A} \in H^t_{A\leftrightarrow B} \mid s \in W\right\}$$

denote all bids offered by B to A in time window W. We can now formulate the average and maximum utility that was offered during the previous time window in the negotiation thread $H = H^W_{B\to A}$:

$$\text{MAX}^W = \max_{x\in H} U_A(x).$$

and

$$\text{AVG}^W = \frac{1}{|H|}\sum_{x\in H} U_A(x).$$

We let $\mathbf{AC}_{\text{combi}}(T,\alpha)$ accept at time t' exactly when the following holds: $\mathbf{AC}_{\text{next}}$ indicates that we have to accept, *or* we have almost reached the deadline ($t' \geq T$) and the current offer suffices (i.e. better than α) given the remaining time:

$$\mathbf{AC}_{\text{combi}}(T,\alpha)$$
$$\overset{\text{def}}{\iff}$$
$$\mathbf{AC}_{\text{next}} \vee \mathbf{AC}_{\text{time}}(T) \wedge \left(U_A(x^t_{B\to A}) \geq \alpha\right).$$

Note that we have defined $\mathbf{AC}_{\text{combi}}(T,\alpha)$ in such a way that it splits the negotiation time into two phases: $[0,T)$ and $[T,1]$, with different behavior in both cases.

We will consider three different combined acceptance conditions:

1. $\mathbf{AC}_{\text{combi}}(T, \text{MAX}^W)$, the current offer is good enough when it is better than all offers seen in the previous time window W,

2. $\mathbf{AC}_{\text{combi}}(T, \text{AVG}^W)$, the offer is better than the average utility of offers during the previous time window W,

3. $\mathbf{AC}_{\text{combi}}(T, \text{MAX}^{\mathcal{T}})$, the offer should be better than any bid seen before.

4. EXPERIMENTS

In order to experimentally test the efficacy of an acceptance condition, we considered a negotiation setup with the following characteristics. We equipped a set of agents (as defined later) with an acceptance condition, and measured its result against other agents in the following way. Suppose agent A is equipped with acceptance condition $\mathbf{AC}_A$ and negotiates with agent B. The two parties may reach a certain outcome $\omega \in \Omega$, for which A receives the associated utility $U_A(\omega)$. The score for A is averaged over all trials on various domains (see Section 4.1.2), alternating between the two preference profiles defined on that domain. E.g., on the negotiation scenario between England and Zimbabwe, A will play both as England and as Zimbabwe against all others.

For our experimental setup we employ Genius (General Environment for Negotiation with Intelligent multi-purpose Usage Simulation) [17]. This environment, which is also used in ANAC, helps to facilitate the design and evaluation of automated negotiators' strategies. It can be used to simulate tournaments between negotiating agents in various negotiation scenarios, such as the setup described in this section. It supports the alternating offer protocol with a real-time deadline as outlined in our negotiation model. The default negotiation time in Genius and in the setup of ANAC is 3 minutes per negotiation session; therefore we use the same value in our experiments.

4.1 Detailed Experimental Setup

4.1.1 Agents

We use the negotiation tactics that were submitted to The Automated Negotiating Agents Competition (ANAC 2010) [3]. ANAC is a negotiation competition aiming to facilitate and coordinate the research into proficient negotiation strategies for bilateral multi-issue negotiation, similar to what the Trading Agent Competition (TAC) has achieved for the trading agent problem [28].

The seven agents that participated in ANAC 2010 have been implemented by various international research groups of negotiation experts. We use these strategies in our experiments as they are representative of the current state-of-the-art in automated negotiation. Firstly, we removed the built-in acceptance mechanism from this representative group of agents; this leaves us with its pure bidding tactics. As outlined in our negotiation model, this procedure allows us to test arbitrary acceptance conditions in tandem with any ANAC tactic.

We aimed to tune our acceptance conditions to the top performing ANAC 2010 agents. Therefore we have selected the top 3 of ANAC agents that were submitted by different research groups, namely *Agent K*, *Yushu* and *IAMhaggler* (we omitted *Nozomi* as the designing group also implemented *Agent K*, cf. Table 1). For the set of opponents, we selected all agents from ANAC 2010, for the acceptance conditions should be tested against a wide array of strategies. The opponents also had their built-in acceptance conditions removed (and hence were not able to accept), so that differences in results would depend entirely on the acceptance condition under consideration. To test the efficacy of an acceptance condition, we equipped the top 3 tactics with this condition and compared the average utility obtained by the three agents when negotiating against their opponents.

4.1.2 Domains

The specifics of a negotiation domain can be of great influence on the negotiation outcome [10]. Acceptance conditions have to be assessed on negotiation domains of different size and complexity. Negotiation results also depend on the *opposition* of the negotiating parties' preferences. The notion of weak and strong opposition can be formally defined [14]. Strong opposition is typical of competitive domains, when a gain for one party can be achieved only at a loss for the other party. Conversely, weak opposition means that both parties achieve either losses or gains simultaneously.

With this in mind, we aimed for two domains (with two preference profiles each) with a good spread of negotiation characteristics. We picked two domains from the three that

	Itex–Cyp	Zim–Eng
Size	180	576
Opposition	Strong	Medium

Table 3: The four preference profiles used in experiments.

were used in ANAC 2010 (cf. [3]). Some agents participating in ANAC 2010 did not scale well and could not deal with a large bid space. We omitted the Travel domain as the agents had too many difficulties with it to make it a reliable testing domain.

Our first scenario is taken from [15], which describes a buyer–seller business negotiation. It involves representatives of two companies: Itex Manufacturing, a producer of bicycle components and Cypress Cycles, a builder of bicycles. There are four issues that both sides have to discuss: the price of the components, delivery times, payment arrangements and terms for the return of possibly defective parts. The opposition between the parties is strong in this domain, as the manufacturer and consumer have naturally opposing requirements. Altogether, there are 180 potential offers that contain all combinations of values for the four issues.

The second domain taken from [17, 18] involves a case where England and Zimbabwe negotiate an agreement on tobacco control. The leaders of both countries must reach an agreement on five issues. England and Zimbabwe have contradictory preferences for the first two issues, but the other issues have options that are jointly preferred by both sides. The domain has a total of 576 possible agreements.

To compensate for any utility differences in the preference profiles, the agents play both sides of every scenario.

4.1.3 Acceptance Conditions

For each acceptance condition we tested all $3\times 7 = 21$ pairings of agents, playing with each of the 4 different preference profiles. We ran every experiment twice, so that altogether each acceptance condition was tested 168 times. We selected the following acceptance conditions for experimental testing. The different values of parameters will be discussed in the section below.

- $\mathbf{AC}_{\text{next}}(\alpha, 0)$ and $\mathbf{AC}_{\text{prev}}(\alpha, 0)$ for $\alpha \in \{1, 1.02\}$,
- $\mathbf{AC}_{\text{gap}}(\alpha)$ for $\alpha \in \{0.02, 0.05, 0.1, 0.2\}$,
- $\mathbf{AC}_{\text{const}}(\alpha)$ for $\alpha \in \{0.8, 0.9\}$,
- $\mathbf{AC}_{\text{time}}(T)$, $\mathbf{AC}_{\text{combi}}(T, \text{MAX}^W)$, $\mathbf{AC}_{\text{combi}}(T, \text{AVG}^W)$ and $\mathbf{AC}_{\text{combi}}(T, \text{MAX}^T)$, where W is the previous time window with respect to the current time t', and $T = 0.99$ (this particular value of T is discussed below).

Additionally, we ran the experiments with agents having their built-in acceptance mechanism in place. That is, we also tested the original agents' *coupled* acceptance mechanism. As we cannot for example, equip *Agent K* with the coupled acceptance condition of *Yushu*, we tested the built-in mechanism by having each agent employ its own mechanism.

4.2 Hypotheses and Experimental Results

The experiments considered here are designed to discuss the main properties and drawbacks of the acceptance conditions listed above. We formulate several hypotheses with respect to the acceptance conditions we have discussed.

Our hypothesis about $\mathbf{AC}_{\text{const}}(\alpha)$ is the following:

Hypothesis 1. For α close to one, $\mathbf{AC}_{\text{const}}(\alpha)$ performs worse than all other conditions.

To evaluate this hypothesis and others below, we have carried out a large number of experiments. The results are summarized in Table 4. The table shows the average utility obtained by the agents when equipped with several acceptance conditions. The "average utility of agreements" column represents the average utility obtained by the agent given the fact that they have reached an agreement. When they do not reach an agreement (due to the deadline), they get zero utility. Thus the following holds:

(The acceptance dilemma)

$$\begin{aligned}\text{Total average utility} \;=\;& \text{Agreement percentage}\\ &\times\\ &\text{Average utility of agreements.}\end{aligned}$$

This formula captures the essence of the acceptance dilemma: accepting bad to mediocre offers yields more agreements of relatively low utility. While accepting only the best offers produces less agreements, but of higher utility.

Now consider $\mathbf{AC}_{\text{const}}(0.9)$ and $\mathbf{AC}_{\text{const}}(0.8)$. When it reaches an agreement, it receives a very high utility (at least 0.9 or 0.8 respectively), but this happens so infrequently (resp. 26% and 38% of all negotiations), that it is ranked at the bottom when we consider total average utility.

We can conclude that our hypothesis is confirmed: in isolation, $\mathbf{AC}_{\text{const}}(\alpha)$ is not very advantageous to use. The main reason is that the choice of the constant α is highly domain-dependent. A very cooperative domain may have multiple win–win outcomes with utilities above α. $\mathbf{AC}_{\text{const}}(\alpha)$ would then accept an offer which is *relatively* bad, i.e. it could have done much better. On the other hand, in highly competitive domains, it may simply 'ask for too much' and may rarely obtain an agreement. Its value lies mostly in using it in combination with other acceptance conditions such as $\mathbf{AC}_{\text{next}}$. It can then benefit the agent by accepting an unexpectedly good offer or a mistake by the opponent.

As we discussed earlier in Section 2.3, the acceptance conditions $\mathbf{AC}_{\text{prev}}(\alpha, 0)$ and $\mathbf{AC}_{\text{next}}(\alpha, 0)$ are standard in literature for $\alpha \in \{1, 1.02\}$. Many agents tend to use these acceptance conditions, as they are well-known and easy to implement. We have formed the following hypothesis:

Hypothesis 2. $\mathbf{AC}_{\text{next}}(\alpha, 0)$ will outperform $\mathbf{AC}_{\text{prev}}(\alpha, 0)$ for $\alpha \in \{1, 1.02\}$. However, both conditions will perform worse than conditions that take the remaining time into account.

To test this hypothesis, we consult Table 4 where we have considered the two values for α. The first observation is that $\mathbf{AC}_{\text{prev}}(\alpha, 0)$ and $\mathbf{AC}_{\text{next}}(\alpha, 0)$ already perform much better than $\mathbf{AC}_{\text{const}}$. The higher value for α yields a better result and $\mathbf{AC}_{\text{next}}(\alpha, 0)$ does indeed outperform $\mathbf{AC}_{\text{prev}}(\alpha, 0)$. It makes sense that comparing the opponent's offer to our up-

Table 4: Utility scores of agents equipped with an acceptance condition

Acceptance Condition	*Agent K*	*IAMhaggler*	*Yushu*	Agreement %	Average utility of agreements	Total avg
$\mathbf{AC}_{\mathrm{combi}}(\mathrm{MAX}^W)$	0.691	0.639	0.695	99%	0.679	0.675
$\mathbf{AC}_{\mathrm{combi}}(\mathrm{AVG}^W)$	0.684	0.634	0.691	99%	0.678	0.670
$\mathbf{AC}_{\mathrm{gap}}(0.1)$	0.636	0.562	0.693	83%	0.761	0.630
Built-in mechanism	0.641	0.547	0.692	82%	0.768	0.627
$\mathbf{AC}_{\mathrm{combi}}(\mathrm{MAX}^{\mathcal{T}})$	0.691	0.577	0.596	89%	0.696	0.621
$\mathbf{AC}_{\mathrm{time}}(0.99)$	0.612	0.580	0.663	99%	0.622	0.618
$\mathbf{AC}_{\mathrm{gap}}(0.2)$	0.626	0.579	0.650	86%	0.721	0.618
$\mathbf{AC}_{\mathrm{gap}}(0.05)$	0.629	0.550	0.672	78%	0.791	0.617
$\mathbf{AC}_{\mathrm{next}}(1.02, 0)$	0.616	0.517	0.696	77%	0.788	0.610
$\mathbf{AC}_{\mathrm{gap}}(0.02)$	0.618	0.491	0.638	73%	0.802	0.582
$\mathbf{AC}_{\mathrm{prev}}(1.02, 0)$	0.618	0.491	0.629	72%	0.805	0.579
$\mathbf{AC}_{\mathrm{next}}(1, 0)$	0.586	0.517	0.597	72%	0.787	0.567
$\mathbf{AC}_{\mathrm{prev}}(1, 0)$	0.588	0.491	0.589	69%	0.805	0.556
$\mathbf{AC}_{\mathrm{const}}(0.8)$	0.286	0.374	0.313	38%	0.851	0.324
$\mathbf{AC}_{\mathrm{const}}(0.9)$	0.215	0.272	0.231	26%	0.935	0.239

coming offer is more beneficial than comparing it to our previous offer, as $\mathbf{AC}_{\mathrm{next}}$ is always 'one step ahead' of $\mathbf{AC}_{\mathrm{prev}}$. However, all time-dependent acceptance conditions outperform both of them, even for $\alpha = 1.02$. This also settles the second part of the hypothesis. The reason for this bad performance is that many bidding strategies focus on the 'negotiation dance' [21]. That is, modeling the opponent, trying to make equal concessions and so on. When a strategy does not explicitly take time considerations into account when making an offer, this poses a problem for the two standard acceptance conditions: they rely completely on the bidding strategy to concede to the opponent before the deadline occurs. When the agent or the opponent does not concede enough near the deadline, the standard conditions lead to poor performance.

Our third hypothesis with respect to the time-dependent condition is as follows:

Hypothesis 3. $\mathbf{AC}_{\mathrm{time}}(T)$ always reaches an agreement, but of relatively low utility.

To evaluate this hypothesis we needed to provide a concrete value for the experimental variable T. We have set $T = 0.99$ for every acceptance condition depending on T. As we have found during preliminary experiments, this value is sufficiently close to the deadline, while it still allows enough time to reach a win-win outcome. From observing the acceptance probability of $\mathbf{AC}_{\mathrm{time}}(0.99)$ in the experimental results, we see that in 1 out of 168 negotiations ($\approx 1\%$) this criterion did not reach an agreement due to agent crashes and protocol errors, in which case both agents received utility zero. But except for these particular events, $\mathbf{AC}_{\mathrm{time}}(T)$ will always reach an agreement, therefore we consider this part of the hypothesis confirmed.

$\mathbf{AC}_{\mathrm{time}}(T)$, with T close to 1 is a sensible criterion to avoid a break off at all cost. It is rational to prefer any outcome over a break off of zero utility. However, the resulting deal can be anything. As we can see from the table, this is the reverse situation of $\mathbf{AC}_{\mathrm{gap}}$: $\mathbf{AC}_{\mathrm{time}}(T)$ yields the lowest agreement score (0.622) of all conditions. This can be explained by the acceptance dilemma: by accepting any offer near the deadline, it reaches more agreements but of relatively low utility. Still the overall score is almost the same (0.618) and thus reasonable. It is interesting to note that $\mathbf{AC}_{\mathrm{time}}(T)$ outperforms both $\mathbf{AC}_{\mathrm{prev}}$ and $\mathbf{AC}_{\mathrm{next}}$ in average overall score.

This insight led us to believe that more consideration has to be given to the remaining time when deciding to accept an offer. The combined acceptance conditions evaluated in the next chapter expand upon this idea to get better deals near the deadline.

4.2.1 Evaluating $\mathbf{AC}_{\mathrm{combi}}(T, \alpha)$

When evaluating $\mathbf{AC}_{\mathrm{combi}}(T, \alpha)$, we expect the following characteristics. $\mathbf{AC}_{\mathrm{combi}}(T, \alpha)$ is an extension of $\mathbf{AC}_{\mathrm{next}}$ in the sense that it will accept under broader circumstances. It alleviates some of the mentioned drawbacks of $\mathbf{AC}_{\mathrm{next}}$ by also accepting when the utility gap between the parties is positive. Also note, that in addition to the parameters that current acceptance conditions use, such as my previous bid $x^{t_{n-1}}_{A\to B}$, my next bid $x^{t'}_{A\to B}$, the remaining time, and the opponent's bid $x^{t}_{B\to A}$, this condition employs the entire bidding history $H^{t}_{A\leftrightarrow B}$ to compute the acceptability of an offer. Therefore we expect better results than with $\mathbf{AC}_{\mathrm{next}}$, with more agreements, and when it agrees, we expect a better deal than by using $\mathbf{AC}_{\mathrm{time}}(T)$.

We capture this last statement in our final hypothesis:

Hypothesis 4. The combination $\mathbf{AC}_{\mathrm{combi}}(T, \alpha)$ outperform other acceptance conditions, such as $\mathbf{AC}_{\mathrm{time}}(T)$ and $\mathbf{AC}_{\mathrm{next}}$ primarily by getting deals of higher utility.

As is evident from the experimental results, $\mathbf{AC}_{\text{combi}}(\text{MAX}^W)$ and $\mathbf{AC}_{\text{combi}}(\text{AVG}^W)$ dominate the other acceptance conditions. They even perform 7% better than the built-in mechanisms of the agent, and 18% better than $\mathbf{AC}_{\text{next}}$. Similar to $\mathbf{AC}_{\text{time}}$, both conditions still get a deal almost every time, but with a higher utility. However, the average utility of an agreement is not the highest: the $\mathbf{AC}_{\text{gap}}$ conditions and the built-in mechanisms get better agreements. But again, we can observe that their agreement rate is also lower, resulting in a higher overall score for the combined criteria.

Aiming for the highest utility that has been offered so far (i.e. $\mathbf{AC}_{\text{combi}}(\text{MAX}^{\mathcal{T}})$) is a less successful criterion, mostly due to a big decrease in agreements. The higher utility that is obtained with this condition does not compensate for the loss of utility that is caused by a break off.

4.3 Related Work

All existing negotiation agent implementations deal with the problem of when to accept. In many cases, the agent accepts a proposal when the value of the offered contract is higher than the offer it is ready to send out at that moment in time. Examples include the time dependent negotiation strategies defined in [22] (e.g. the Boulware and Conceder tactics). The same principle is used in the equilibrium strategies of [9] and for the Trade-off agent [7]. In the setting of [7] however, the deadline can be different for both agents. In this paper, we consider strategies that do not always reach an agreement, and hence we have concentrated on acceptance conditions that yield better results in such cases.

Of all ANAC 2010 participants, we shortly discuss *Agent K* [25] as it employs the most sophisticated method to decide when to accept. Its acceptance mechanism is based on the mean and variance of all received offers. It then tries to determine the best offer it might receive in the future and sets its proposal target accordingly. In contrast to our approach, this mechanism is not fully decoupled from the bidding strategy as it directly influences its bid target. Furthermore, it does not restrict its scope to the remaining or previous time window. Finally, we note that *Agent K* performs better in our experimental setup (cf. Table 4) when equipped with our combined acceptance conditions than with its built-in mechanism.

Although we do not focus on negotiation tactics and convergence results, our negotiation model builds upon the model of [26]. However, in this model, the action function of an agent only takes into account the offer it is ready to send out at that moment in time. Moreover, the focus of the paper is not on comparing acceptance conditions as only one specific instance is studied. We take a more general approach in which the agent utilizes a generic acceptance mechanism, in which the current time and the entire bidding history is considered.

5. CONCLUSION AND FUTURE WORK

In this paper, we aimed to classify current approaches to generic acceptance conditions and to compare a selection of acceptance conditions in an experimental setting. We presented the challenges and proposed new solutions for accepting offers in current state-of-the-art automated negotiations. The focus of this paper is on decoupled acceptance conditions, i.e. general conditions that do not depend on a particular bidding strategy.

Designing an effective acceptance condition is challenging because of the acceptance dilemma: better offers may arrive in the future, but waiting for too long can result in a break off of the negotiation, which is undesirable for both parties.

We have seen that the standard acceptance criterion $\mathbf{AC}_{\text{next}}$ is often used by negotiating agents. From our results, it is apparent that $\mathbf{AC}_{\text{next}}$ does not always yield optimal agreements. We established that it performs worse than more sophisticated acceptance conditions.

In addition to classifying and comparing existing acceptance conditions, we have devised three new acceptance conditions by combining existing ones. This included two acceptance conditions that estimate whether a better offer might occur in the future based on recent bidding behavior. These conditions obtained the highest utility in our experiments and hence performed better than the other conditions we have investigated.

A suggestion for future research would be to explore the many possible combinations of acceptance conditions that may be obtained using conjunction and disjunction (and possibly negation). Some agents already use a logical combination of different acceptance conditions at the same time. For example, the *IAM(crazy)Haggler* agents accept when

$$\mathbf{AC}_{\text{const}}(0.88) \vee \mathbf{AC}_{\text{next}}(1.02, 0) \vee \mathbf{AC}_{\text{prev}}(1.02, 0).$$

A suitable combination of acceptance conditions could provide a considerable improvement over current acceptance conditions.

Secondly, we plan to test acceptance conditions with more agents and on larger domains, using the resources that will be available after the upcoming ANAC 2011 event.

Finally, we did not consider negotiation domains with discount factors, which devaluate utility with the passing of time. Adding discount factors will require new acceptance conditions that give more consideration to the negotiation timeline. We plan to examine such extensions in future work.

Acknowledgements

This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs. It is part of the Pocket Negotiator project with grant number VICI-project 08075.

6. REFERENCES

[1] Bo An and Victor Lesser. Yushu: a heuristic-based agent for automated negotiating competition. In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence (to appear).* Springer-Verlag, 2010.

[2] R.J. Aumann and S. Hart, editors. *Handbook of Game Theory with Economic Applications*, volume 1. Elsevier, March 1992.

[3] Tim Baarslag, Koen Hindriks, Catholijn M. Jonker, Sarit Kraus, and Raz Lin. The first automated negotiating agents competition (ANAC 2010). In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *New Trends in*

Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence (to appear). Springer-Verlag, 2010.

[4] Enrico H. Gerding Colin R. Williams, Valentin Robu and Nicholas R. Jennings. Iamhaggler: A negotiation agent for complex environments. *This volume*.

[5] Enrico H. Gerding Colin R. Williams, Valentin Robu and Nicholas R. Jennings. Iamhaggler: A negotiation agent for complex environments. In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence (to appear)*. Springer-Verlag, 2010.

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

[7] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make negotiation trade-offs. *Journal of Artificial Intelligence*, 142(2):205–237, 2003.

[8] S. S. Fatima, M. Wooldridge, and N. R. Jennings. Optimal negotiation strategies for agents with incomplete information. *Intelligent Agents Viii: Agent Theories, Architectures, and Languages*, 2333:377–392, 2002.

[9] Shaheen S. Fatima, Michael Wooldridge, and Nicholas R. Jennings. Multi-issue negotiation under time constraints. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 143–150, New York, NY, USA, 2002. ACM.

[10] Koen Hindriks and Dmytro Tykhonov. Towards a quality assessment method for learning preference profiles in negotiation, 2008.

[11] Koen V. Hindriks and Dmytro Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning, 2008.

[12] Takayuki Ito, Hiromitsu Hattori, and Mark Klein. Multi-issue negotiation protocol for agents : Exploring nonlinear utility spaces, 2007.

[13] Catholijn M. Jonker, Valentin Robu, and Jan Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Journal of Autonomous Agents and Multi-Agent Systems*, 15(2):221–252, 2007.

[14] G.E. Kersten and S.J. Noronha. Rational agents, contract curves, and inefficient compromises report. Working papers, International Institute for Applied Systems Analysis, 1997.

[15] Gregory E. Kersten and Grant Zhang. Mining inspire data for the determinants of successful internet negotiations. *InterNeg Research Papers INR 04/01 Central European Journal of Operational Research*, 2003.

[16] Sarit Kraus. *Strategic Negotiation in Multiagent Environments*. Mit Press, October 2001.

[17] R. Lin, S. Kraus, D. Tykhonov, K. Hindriks, and C. M. Jonker. Supporting the design of general automated negotiators. In *Proceedings of the Second International Workshop on Agent-based Complex Automated Negotiations (ACAN'09)*, 2009.

[18] Raz Lin, Sarit Kraus, Jonathan Wilkenfeld, and James Barry. Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence*, 172(6-7):823 – 851, 2008.

[19] Raz Lin, Yinon Oshrat, and Sarit Kraus. Investigating the benefits of automated negotiations in enhancing people's negotiation skills. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 345–352, 2009.

[20] Martin J. Osborne and Ariel Rubinstein. *Bargaining and Markets (Economic Theory, Econometrics, and Mathematical Economics)*. Academic Press, April 1990.

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

[22] R. Ros and C. Sierra. A negotiation meta strategy combining trade-off and concession moves. *Autonomous Agents and Multi-Agent Systems*, pages 163–181, 2006.

[23] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982.

[24] Liviu Dan Serban, Gheorghe Cosmin Silaghi, and Cristian Marius Litan. Agent fsega - time constrained reasoning model for bilateral multi-issue negotiations. In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence (to appear)*. Springer-Verlag, 2010.

[25] Katsuhide Fujita Shogo Kawaguchi and Takayuki Ito. Compromising strategy based on estimated maximum utility for automated negotiating agents. In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence (to appear)*. Springer-Verlag, 2010.

[26] C. Sierra, P. Faratin, and N.R. Jennings. A service-oriented negotiation model between autonomous agents. In M. Boman and W. van de Velde, editors, *Proceedings of the 8th European Workshop on Modelling Autonomous Agents in Multi-Agent World, MAAMAWŠ97*, volume 1237 of *Lecture Notes in Artificial Intelligence*, pages 17–35. Springer-Verlag, 1997.

[27] Niels van Galen Last. Opponent model estimation in bilateral multi-issue negotiation. In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence (to appear)*. Springer-Verlag, 2010.

[28] Michael P. Wellman, Peter R. Wurman, Kevin O'Malley, Roshan Bangera, Shou de Lin, Daniel Reeves, and William E. Walsh. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5(2):43–51, 2001.

Heuristic-based Approaches for CP-Nets in Negotiation *

Reyhan Aydoğan[1], Tim Baarslag[2], Koen V. Hindriks[2], Catholijn M. Jonker[2], Pınar Yolum[1]

[1]Bogazici University, Bebek, Istanbul, Turkey
{reyhan.aydogan, pinar.yolum}@boun.edu.tr
[2]Delft University of Technology, Delft, The Netherlands
{T.Baarslag, k.v.hindriks, c.m.jonker}@tudelft.nl

ABSTRACT

CP-Nets have proven to be an effective representation for capturing preferences. However, their use in multiagent negotiation is not straightforward. The main reason for this is that CP-Nets capture partial ordering of preferences, whereas negotiating agents are required to compare any two outcomes based on the request and offers. This makes it necessary for agents to generate total orders from their CP-Nets. We have previously proposed a heuristic to generate total orders from a given CP-Net. This paper proposes another heuristic based on Borda count, applies it in negotiation, and compares its performance with the previous heuristic.

1. INTRODUCTION

Modeling users' preferences is an inevitable part of automated negotiation tools. While representing the user's preferences, there are several issues to be taken into account. One, outcome space grows exponentially with the number of attributes and their possible values. It may be infeasible to ask a user to rank all outcomes when the outcome space is very large. Two, the user may have difficulty in assessing her preferences in a quantitative way [4]. Representing someone's preferences with numerical values is an arduous task for a human. Three, it is difficult to find a mathematical model for representing preferences in which there are preferential dependencies between attributes. Therefore, it is more effective and intuitive to use a qualitative preference model.

Although it is desired for users to express their preferences qualitatively, most of the current negotiation strategies [5, 7] work with quantitative preferences. Hence, to use qualitative preferences in negotiation, it is necessary to estimate quantitative preferences from qualitative preferences. Accordingly, this paper is about estimation of quantitative preferences from qualitative preferences. That is, we propose heuristics to allow agents to have a qualitative preference model, while their negotiation strategy employs quantitative information. In order to do so, we start from a qualitative preference representation, namely CP-Nets. CP-Nets allow representation of conditional preferences and tolerate partial ordering. We extend the GENIUS negotiation framework [6] to allow elicitation of acyclic CP-Net preferences. Then, we apply our heuristics to generate utility-based information from the given CP-Net.

We compare the performance of agents when they apply heuristics on their users' qualitative preferences and negotiate with estimated utilities versus when they have their users' real total preference orderings and negotiate with real utilities. To accomplish this, users were asked to create their preference profiles both quantitatively (UCP-Nets) and qualitatively (CP-Nets), using the GENIUS interface for an apartment renting domain. The given UCP-Nets serve as ground truth. The agents apply heuristics on the given CP-Net and then negotiate with the resulting estimated utilities. Each negotiation outcome is evaluated based on the given UCP-Net, which is not only consistent with the CP-Net but also provides a total ordering of outcomes.

The rest of this paper is organized as follows: Section 2 gives an introduction on CP-Nets and UCP-Nets. Section 3 explains the heuristics that we propose to be used with CP-Nets. Section 4 explains our experimental setup, metrics, and results. Finally, Section 5 discusses our work.

2. BACKGROUND: CP-NETS & UCP-NETS

Conditional preference networks (CP-nets) is a graphical model for representing qualitative preferences in a compact way [4]. In CP-nets, each node represents an attribute and each edge denotes preferential dependency between nodes. If there is an edge from X to Y, X is called "parent node" and Y is called "child node". The preference on child nodes depends on their parent nodes' values. To express conditional preferences, each node is associated with a conditional preference table (CPT), which represents a total order on possible values of that node with respect to its parents' values.

Consider apartment renting domain in Example 1 and a CP-NET expressing that its user's preference on parking area depends on neighborhood. CPT for *Parking Area* shows that the user prefers an apartment having a parking area when the neighborhood is either *Kadikoy* or *Kartal*. However, she prefers an apartment not having a parking area when it is at *Etiler*. In CP-nets, each preference statement is interpreted under "everything else being equal" interpretation. The statement, "*Etiler* is preferred over *Kartal*", means that if all other attributes such as price and parking area are the same, an apartment at *Etiler* is preferred over an apartment at *Kartal*.

EXAMPLE 1. *For simplicity, we have only three attributes in our apartment renting domain: Price, Neighborhood and Parking Area. There are three neighborhoods: Etiler, Kadikoy and Kartal whereas the valid values for the price are categorized as High, Medium and Low. A parking area may exist or not. Thus, the domain for parking area has two values: Yes and No.*

We need to check whether there exists an *improving flip* sequence from one outcome to another (and vice versa) to answer whether an outcome would be preferred over another. An improving flip is changing the value of a single attribute with a more desired value by using CPT for the attribute. If there are not any improving flip sequences between two outcomes, we cannot compare these two outcomes. Thus, the inability of comparing some outcomes is the challenge of using CP-Nets in negotiation.

*This research is supported by Boğaziçi University Research Fund under grant BAP5694 and Turkish Research and Technology Council.

Boutilier *et al.* propose UCP-nets [3] by CP-Nets with generalized additive models. UCP-nets are able to represent preferences quantitatively rather than representing simply preference ordering.

Similar to CP-nets, we firstly specify preferential dependency among attributes. Instead of specifying a total preference ordering over the values of each attribute according to their parents' values (conditions), we assign a real value (utility) for all values of each attribute. Utility function *u*(X_1, X_2,...X_n) is represented in Equation 1 where X_i is the i^{th} attribute of outcome, U_i denotes parents of X_i and $f_i(X_i, U_i)$ represents a factor. Assume that our UCP-Net involves three factors f_1(Neighborhood), f_2(Price) and f_3(Parking Area, Neighborhood). The utility of an outcome is estimated as the sum of these factors.

$$u(X_1, X_2, ...X_n) = \sum_i f_i(X_i, U_i) \quad (1)$$

3. PROPOSED HEURISTICS

Most of the negotiation strategies [5, 7] work with quantitative preferences such as *utility functions*. However, it is desired for users to express their preferences qualitatively. Thus, we propose heuristics to use acyclic CP-Nets in negotiation while agents still negotiate with their strategies using quantitative information, *utility* (a real value between zero and one). To do this, we generate predicted utilities from a given CP-Net by applying our heuristics.

In our framework, a preference graph is induced from a given CP-Net while eliciting a user's preferences as a CP-Net. In this preference graph, each node denotes a possible outcome and each edge represents an improving flip. The direction of edges are ordered from less desired to more desired services. Therefore, the worst outcome will be placed at the top of preference graph (root node) whereas the leaf node holds the best outcome. For intermediate nodes, we only compare the nodes having a path from others. The nodes having no path to each other cannot be compared.

Figure 1 shows a preference graph induced from a CP-Net. The node (*Yes*, *Etiler*, *Low*) represents a low-priced apartment at *Etiler* having a parking area. There is an edge from (*No*, *Kartal*, *High*) to (*No*, *Kartal*, *Medium*). This means that an apartment with a medium price at *Kartal* not having a parking area is preferred over an apartment with a high price at *Kartal* not having a parking area.

An agent having a CP-Net applies one of the following heuristics and uses the estimated utilities produced by a chosen heuristic.

3.1 Depth Heuristic (DH)

We have previously proposed an approach based on capturing the depth of an outcome in preference graph [1] but in that study *depth* is used by the proposed negotiation strategy — it is not independent from the negotiation strategy. However, in this study we use the concept of *depth* to produce estimated utilities of outcomes regardless of negotiation strategy. That is, the agent using this heuristic is able to apply any negotiation strategy.

The depth of an outcome node in a preference graph indicates how far it is from the worst choice. It is intuitive to say that the better (more preferred) a service is, the further it is from the worst outcome. The depth of an outcome node is estimated as the length of the longest path from the root node keeping the worst choice.

According to this approach, the higher the depth of an outcome, the more likely it is to be preferred by the user. Further, if two outcomes are at the same depth, it is assumed that these outcomes are equally preferred by the user. We apply Equation 2 to estimate the utility values between zero and one. In short, the depth of a given outcome is divided by the depth of the preference graph (the highest depth) to obtain estimated utility of that outcome. For example, if we have a preference graph with a depth of 6 in Figure 1, an outcome whose depth is equal to 3 will have utility of $0.5(= 3/6)$.

$$U(x) = \frac{Depth(x, PG)}{Depth(PG)} \quad (2)$$

3.2 Borda Scoring Heuristic

CP-Nets order outcomes partially and there are a plenty of linear orderings consistent with the partial ordering of outcomes induced from a CP-Net. One of these linear orderings may reflect the user's real preference orderings. Thus, this heuristic is based on finding all possible linear extensions of a given partial preference ordering and selecting one of the most suitable linear extensions.

One possible way of a linear ordering is to apply a voting procedure. To do this, we estimate all linear extensions of a given partial preference ordering induced from a preference graph and apply a voting procedure called "Borda Rule" [2] to obtain one of the most suitable linear orderings.

According to Borda Rule, we score outcomes according to their position in the ordering. Assume that we have *m* alternatives ordered as $< o_1, o_2 ... o_m >$ where o_{i+1} is preferred over o_i. When we score the outcomes, each outcome will get a point of its position minus one (o_i will get $i-1$). The sum of points namely *Borda count* represents the aggregation of existing alternative orderings. To illustrate this, consider we have three orderings such as $< x, y, z >$, $< z, x, y >$, $< x, z, y >$ where x, y and z are possible outcomes. Borda count of x would be equal to one $(= 0 + 1 + 0)$. In this approach, Borda count of each outcome over all possible linear extensions will reflect how much that outcome is preferred. Thus, we will estimate utilities based on the calculated Borda counts.

On the other hand, the number of all possible linear extensions of a given partial ordering may be so huge that this technique may become impractical because of high complexity. In order to reduce the complexity, we partition the preference graph and apply Borda Rule to all possible linear extensions of each subpartition.

How do we partition the preference graph? We know that the root node holds the worst outcome while the leaf node holds the best outcome. Thus, we need to find an ordering for the outcomes within the intermediate nodes. We partition this part in such a way that each subpartition can involve at most n, predefined number of outcomes. For this purpose, n can be taken as 10 or 15 according to the size of the preference graph. We choose 10 in this study.

After applying Borda rule to each partition, we normalize Borda counts in a way that Borda count of each outcome will be between zero and one. To do this, we divide Borda count of each outcome in that partition by the maximum Borda count.

Another issue pertains to using these normalized Borda counts in order to estimate final utilities. We distribute the utilities by considering the number of outcomes at each partition. To achieve this, we apply the formula in Equation 3 where $U(x, p_i)$ denotes the utility of outcome x in the i^{th} partition, $U_{max}(i-1)$ denotes the utility of outcome whose utility is maximum in the previous partition $(i-1)$, N denotes the number of possible outcomes, S_{p_i} denotes the number of outcomes in i^{th} partition and $BRCount(x, p_i)$ denotes normalized Borda count of the outcome x. $U_{max}(p_0)$, the utility of worst outcome (root node in the preference graph), is equal to $1/N$.

$$U(x, p_i) = U_{max}(p_{i-1}) + \frac{S_{p_i}}{N} * BRCount(x, p_i) \quad (3)$$

4. EXPERIMENTS

To evaluate the proposed heuristics, we extend GENIUS [6], which is a platform for bilateral negotiation. Our extension enables an

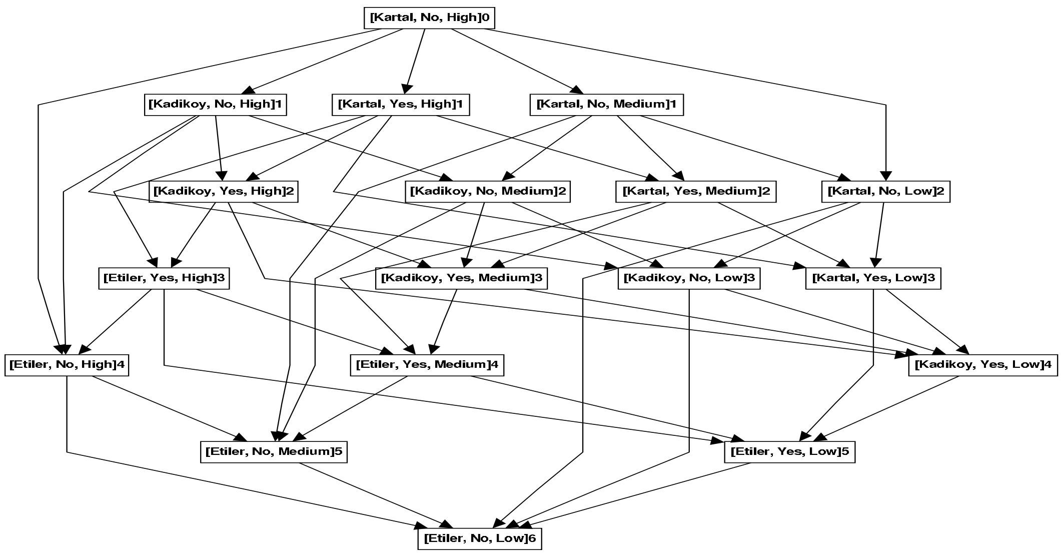


Figure 1: Induced preference graph from a given CP-Net.

agent to elicit user's preferences as CP-Nets and to use utilities estimated by chosen heuristic while negotiating. The platform also stores the user's total ordering of outcomes as UCP-Nets and evaluates each negotiation outcome for that agent based on the given UCP-Net. The given UCP-Net is consistent with the given CP-net. In our experiments, the UCP-Net serves as ground truth. After an agent negotiates using its CP-Net, we evaluate its performance as if we knew the correct total ordering (UCP-Net).

We investigate three test cases to compare the performance of the heuristics. In each test case, two agents *Agent A* and *Agent B* negotiate with each other. We fix both agents' negotiation strategies so that *Agent A* negotiates with the same *Agent B* (having same preference profile and strategy). In the first case, *Agent A* has a CP-net and applies Depth Heuristic (DH) to derive the estimated utilities. During the negotiation, the agent will act on according to these estimated utilities. In the second case, *Agent A* has the same CP-Net with the first case but it applies Borda Scoring Heuristic (BSH) to estimate utilities which will be used in negotiation. In the last case, *Agent A* has its user's real total preference orderings in the form of UCP-Net (consistent with the CP-net and able to compare all outcomes). Thus, it uses the real utilities. Consequently, we are able to observe what the agent gets at the end of negotiation when it applies heuristics on partial preference information (CP-Net) versus when it has total preference information (UCP-Net).

In our experiments, each agent uses a concession based strategy in which the agent starts with the outcome having the highest utility and concedes over time. It also remembers the best counter offer that is made by the opponent agent. If the utility of the current counter offer is higher than or equal to the utility of agent's previous offer, then the agent will accept the offer. The agent will take the best counter offer of its opponent into account while generating its offer. If the utility of the current offer is lower than that of the best counter offer, the agent will take the opponent's best counter offer.

Since the opponent agent (*Agent B*)'s preference profile has a significant impact on negotiation outcome, we generate 50 different preference profiles for *Agent B*. That is, the same *Agent A* will negotiate with 50 different *Agent B*s. *Agent B*'s preferences are represented with a linear additive utility function in this experiment. Another factor having an influence on negotiation outcome in this setting is UCP-Net of the user. Different UCP-Nets mean different ordering of outcomes, so represent different users. Thus, we generate four different UCP-Nets for *Agent A* consistent with the given CP-net—four different users having the same CP-net. As a result, both agents will negotiate 200 times (4 different users of *Agent A* * 50 different *Agent B*).

Furthermore, we investigate the performance of the heuristics from a different point of view by taking the structure of CP-Nets into account. We generate three different CP-Nets. *CPNet*-1 involves one dependency such as preference of *parking area* depends on *neighborhood* whereas *CPNet*-2 involves two dependencies such as both preferences of *parking area* and *price* depend on *neighborhood*. There are not any dependencies between attributes in *CPNet*-3. For each CP-Net, we generate four different UCP-Nets consistent with them and perform the experiments mentioned above.

4.1 Sum of Utilities for Agent A

Our first evaluation criterion is the sum of negotiation outcomes' utilities with respect to *Agent A* over 50 different negotiations with *Agent B*. Table 1 shows these total utilities for three different CP-Nets and four different UCP-Nets consistent with each CP-Net. As expected *Agent A* using UCP-Net gets the highest score when it has a consistent UCP-Net with *CPNet*-1 and *CPNet*-3 since it negotiates with user's real preference orderings. Overall, the performance of the agent using BSH is quite close to that of the agent using UCP-Net (172 vs. 179 and 171 vs. 172). For the case of *CPNet*-2, the score of BSH is approximately the same as the score of UCP-net. Since *CPNet*-2 involves two dependencies (the user specifies her preferences in a more detailed way), the agent may get more information than the case of other CP-Nets (one dependency and no dependency). This leads to better results. The score of heuristics are the highest when they have *CPNet*-2.

Moreover, *Agent A*'s score while applying Borda Scoring Heuristic (BSH) is higher than the case in which it uses Depth Heuristic (DH) for all CP-Nets (based on overall sum over 200 negotiations). According to this criterion, BSH may be preferred over DH.

Table 1: **Sum of Outcome Utilities over 50 Negotiations for *Agent A***

AGENT A	DH	BSH	UCP-Net
CPNET-1 with UCPNet-1A	39.03	39.00	41.88
CPNET-1 with UCPNet-2A	38.27	40.73	43.66
CPNET-1 with UCPNet-3A	45.73	45.69	45.80
CPNET-1 with UCPNet-4A	46.88	46.94	47.29
Overall Sum (200 negotiations):	**169.91**	**172.36**	**178.63**
CPNET-2 with UCPNet-1B	39.93	41.66	41.70
CPNET-2 with UCPNet-2B	42.94	43.56	43.21
CPNET-2 with UCPNet-3B	46.15	46.76	46.75
CPNET-2 with UCPNet-4B	42.18	43.56	43.53
Overall Sum (200 negotiations):	**171.20**	**175.55**	**175.20**
CPNET-3 with UCPNet-1C	40.17	41.61	40.83
CPNET-3 with UCPNet-2C	41.58	42.50	45.64
CPNET-3 with UCPNet-3C	42.83	43.97	43.37
CPNET-3 with UCPNet-4C	42.36	43.36	42.64
Overall Sum (200 negotiations):	**166.94**	**171.44**	**172.48**

4.2 Number of Times as Well as UCP-Net

Our second evaluation criterion is the number of times that the agent that applies a heuristic on a given CP-Net negotiates at least as well as the agent having a UCP-Net. If the utility of outcome for the agent using a heuristic is higher than or equal to the utility of outcome for the agent having UCP-Net, that agent receives one point. Since 50 different *Agent B*s negotiate with the same *Agent A*, we evaluate this criterion over 50 negotiations.

According to Table 2, when *Agent A* uses *CPNet*-1 and applies DH, it negotiates at least as well as the agent having total preference ordering (UCP-Net) in 78 per cent of negotiations whereas BSH is successful at least as UCP-Net in 76 per cent of negotiations. Although the performance of BSH with respect to sum of utilities is better than that of DH, it negotiates as successfully as UCP-Net more than BSH for *CPNet*-1 (78 per cent versus 76 per cent). This stems from the fact that when BSH completes a negotiation better than DH, the difference between utilities of the outcomes is much higher than the case when DH negotiates better than BSH.

Table 2: **Number of Times Heuristics Performs As Well As UCP-Nets**

AGENT A	DH	BSH
CPNET-1 with UCPNet-1A	40	35
CPNET-1 with UCPNet-2A	26	35
CPNET-1 with UCPNet-3A	46	38
CPNET-1 with UCPNet-4A	44	44
Overall Sum (200 negotiations):	**156**	**152**
CPNET-2 with UCPNet-1B	43	49
CPNET-2 with UCPNet-2B	48	48
CPNET-2 with UCPNet-3B	44	50
CPNET-2 with UCPNet-4B	44	50
Overall Sum (200 negotiations):	**179**	**197**
CPNET-3 with UCPNet-1C	44	50
CPNET-3 with UCPNet-2C	27	31
CPNET-3 with UCPNet-3C	45	49
CPNET-3 with UCPNet-4C	47	47
Overall Sum (200 negotiations):	**163**	**177**

For *CPNet*-2 and *CPNet*-3, the agent using BSH negotiates successfully as the agent having UCP-Net more than the agent using DH. When agents have *CPNet*-2, it is seen that BSH beats DH. Note that in 89.5 per cent of negotiations DH negotiates at least as well as UCP-Net whereas 98.5 per cent of negotiations BSH performs at least as good as the UCP-Net.

5. DISCUSSION

Our experimental results show that it would be better to apply Borda Scoring heuristic (BS) in small domains since its performance is higher than that of Depth heuristic (DH). However, we may prefer to use DH in large domains since its complexity is lower than BSH.

Li *et al.* study the problem of collective decision making with CP-Nets [8]. Their aim is to find a Pareto-optimal outcome when agents' preferences represented by CP-Nets. They firstly generate candidate outcomes to increase the computational efficiency instead of using the entire outcome space. Then each agent ranks these candidate outcomes according to their own CP-Nets. For ranking an outcome, they use *the longest path between the optimal outcome and that outcome* in the induced preference graph. Thus, the minimum rank is desired for the agents. They choose the final outcome for the agents by minimizing the maximum rank of the agents. In contrast, we use *the longest path between the worst outcome and that outcome* to estimate the utilities with our depth heuristic. Moreover, while they propose a procedure for collective decision making, we focus on estimating utility values of each outcome that will be used during the negotiation for an agent.

Rosi *et al.* extend CP-Nets to capture multiple agents' preferences and present *m*CP-Nets [9]. They propose several voting semantics to aggregate agents' qualitative preferences and to determine whether an outcome is preferred over another for those agents. They propose to rank an outcome in term of the length of the shortest sequence of worsening flips between that outcome and one of the optimal outcomes while we use the longest sequence of improving flips between the worst outcome and that outcome in our depth heuristic to get the estimated utilities.

6. REFERENCES

[1] R. Aydoğan and P. Yolum. Heuristics for cp-nets: Negotiating effectively with partial preferences. In *HUCOM Workshop in Group Decision and Negotiation*, The Netherlands, 2010.

[2] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

[3] C. Boutilier, F. Bacchus, and R. I. Brafman. Ucp-networks: A directed graphical representation of conditional utilities. In *Proceedings of UAI '01*, pages 56–64, USA, 2001.

[4] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res. (JAIR)*, 21:135–191, 2004.

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

[6] K. Hindriks, C. M. Jonker, S. Kraus, R. Lin, and D. Tykhonov. Genius: negotiation environment for heterogeneous agents. In *Proceedings of AAMAS '09 - Vol. 2*, pages 1397–1398, 2009.

[7] C. M. Jonker, V. Robu, and J. Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *JAAMAS*, 15(2):221–252, 2007.

[8] M. Li, Q. B. Vo, and R. Kowalczyk. An efficient procedure for collective decision-making with cp-nets. In *Proceeding of ECAI 2010*, pages 375–380, Lisbon, Portugal, 2010.

[9] F. Rossi, K. B. Venable, and T. Walsh. mcp nets: Representing and reasoning with preferences of multiple agents. In *Proceeding of AAAI'04*, pages 729–734, 2004.

Experiments on Buyer's Trend in New E-Commerce Evaluation Model

Koki Murakata
Yamagata University
4-3-16 Jonan
Yonezawa, Yamagata
murakata2009@e-activity.org

Tokuro Matsuo
Yamagata University
4-3-16 Jonan
Yonezawa, Yamagata
matsuo@yz.yamagata-u.ac.jp

ABSTRACT
Agent-based e-commerce is a popular and promising field to make rational and robust trading. In e-commerce, buyer can know trader effectively by evaluation system. However, the evaluation information includes incomplete and asymmetric information. We propose a novel evaluation model based on multiattribute. The new part of this model is that seller makes up multiattribute evaluations and number of selected evaluation attributes affect evaluation value. We conduct some experience to survey relation between number of attribute and evaluation value in the situation in which questionnaire results about e-commerce are reflected. Proposed our model shows to reduce incomplete and asymmetric information.

Categories and Subject Desceiptors
I.2 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; I.6 [**Simulation and Modeling**]: Model Validation and Analysis

General Terms
Theory and Experiment

Keywords
Evaluation System, E-Commerce, Incentive Design, Incomplete Information, Asymmetric Information

1. INTRODUCTION
In recent years, a lot of naive users experience a trade on the Internet, such as e-auction, e-group buy, and e-shoppings. The market size of them has been developing and increasing, that is, e-commerce market has increased 10 % or more annually in Japan. As the market size develops, the number of crimes and frauds on the Internet has been increasing year by year. Generally, electronic commerce site provides seller rating function in order to make seller's information disclose [1][2]. Even if the e-commerce site uses simple evaluation system, the evaluation system is effective to avoid dishonest behavior [3][4]. Existing evaluation system has a strong limitation that users input rating on unified evaluation attributes. Criteria to be evaluated also are not defined in existing evaluation systems. Even though a seller provides incomplete/incorrect information on an e-commerce website, buyers can never to declare on the seller rating system. In some cases, there are a lot asymmetric information between buyers and sellers.

To solve the above problems in e-commerce and evaluation system, we propose a new evaluation method in which sellers disclose a lot of faithful item's information. In our method, sellers can freely choose evaluation attributes that is important for sellers to deal. Evaluation for seller is determined the synthetic evaluation based on number of the attributes. If the seller provides many evaluation items, our model give extra points for the seller. In latter of this paper, we provide the result of experiment reflected buyer's preferences to clarify the feature of our model and give some discussion regarding seller's strategy.

The rest of this paper shows as follows. In Section 2, we explain about incomplete information in e-commerce. Section 3 introduces existing evaluation systems and some related work. Then, in Section 4, we propose a novel evaluation model based on the number of displayed information. After that, Section 5 gives experiments using real data regarding buyer's preferences and discusses seller's strategy in each experiment case. Finally, we summarize our study and show our future work in Section 6.

2. PRELIMINARY DISCUSSIONS
2.1 Incomplete Information
In the Internet-based auction, buyers view items information and sellers information based on only displayed information on the web browser. Buyers cannot perfectly know the actual information by the Internet until they receive purchased items. These situations put out incomplete information, such as every existing electronic commerce web site. On another hand, in an e-marketplace, differences of quantity and quality of information between sellers and buyers are huge issue for them. These situations put out the problem on asymmetric information. Web-based marketplace has more asymmetric information than actual marketplaces. In the actual marketplaces, buyers can view items from multiple aspects, sometimes touch and pick up them. Thus, they make sure the material, quality, size, and several other information. On the other hands, when users try to buy items on the electronic marketplace, they cannot touch and pick up items. Further, they just look at some pictures taken by sellers. Some sellers are good faith and honesty, but others may hide a scuff on the item and do not provide adverse information. It makes unfair trades. It is very important for buyers to be filled the gap of information between them and sellers. When there are above unfair issues on the trades, buyers sometimes fail their decision making to select items. This means that buyers' utilities are decreased by unfair information provision.

2.2 Existing Evaluation Systems
Yahoo! [5], Rakuten [6] and amazon.co.jp [7] are popular e-commerce sites in Japan. In their system, users can input their evaluation including total/synthetic evaluation and evaluation by free description. total/synthetic evaluation is not detailed the information and is abstract information. In additional, overall rating is no clear criteria and has the trend to be evaluated according to buyer's subjectivity. Although existing evaluation

systems have these features, buyers can never perfect information about sellers and items with incomplete and asymmetric information. A lot of causes of criminal acts are set up by these problems on information.

3. EVALUATION MODEL

In this section, we propose a new objective evaluation model based on quality and quantity of disclosure of information. First, we put and attach the concept of criteria to evaluate. In existing evaluation systems, users sometimes confuse because of non-criteria for evaluation. For example, popular e-commerce sites provide only synthetic evaluation. Some other sites provide multiple attributes to evaluate like "Speedy deliver", "Politeness to customers", and several others. However, How do sellers gain good evaluation about "Speedy deliver" ? How do sellers get positive score about "Attitude to buyers" ? Even though a seller does same attitude toward helping and taking care of the customer, each evaluation from buyer would be different. Thus, to make more useful information, our proposed evaluation system sets concrete criteria. Further, we set an incentive model for sellers to grow and improve on trading skill.

3.1 Model

Evaluation index for sellers in the evaluated by buyers defined $I=\{1,2,...,i,...,n\}$. Impression value $A=\{\alpha_1,\alpha_2,...,\alpha_i,...,\alpha_n\}$ is defined an impression when a buyer looks at the item's information on the e-commerce site. Influence value C_A is also defined from external effects at a browsing the e-commerce site. Impression value $B=\{\beta_1,\beta_2,...,\beta_i,...,\beta_n\}$ is defined an impression after a buyer receives the item. Influence value C_B is also defined from an external effects when a buyer receives the delivered item. We define that buyer's synthetic impression value C_A with impression function F_A when a buyer browses the item information at the e-commerce site. The value is indicated as $G_A=F_A(A, C_A)$. We define that buyer's synthetic impression value with impression function F_B when he/she receives the delivered item. The value is indicated as $G_B=F_B(B, C_B)$. We simply assume that the functions F_A and F_B are additional value functions.

3.2 Evaluation from Trading Partners

Even though an expression value and item information value are same, sensitivity and feeling of the explanation and introduction of items are different for each buyer. When evaluations are given using a stage assessment model, each buyer evaluates based on his/her multiple scale. To avoid such dispersion, we set a criterion for each evaluation attribute. For example, when the delivered item is evaluated on the sameness between actual item and the picture shown at the e-commerce site, we give a certain criterion like shown in Table 1. The adjusted value of important criteria is higher and the value of unimportant criteria is lower. The values can be changed by the e-commerce site manager.

Table 1. Example of criteria and adjusted values

Criteria	Rate
Delivered item is same with the picture on the web	1.5
Actual item's size is same with the description on the web	1.3
:	

Thus, incomplete information are reduced by these evaluations based on comparison between actual things and criteria. If a lot of buyers evaluate the attribute in which the original item is different from the picture on the web in the past, the seller is known as a person who does not deal in the acceptable item. Our proposed model provides more concrete information comparing with existing e-commerce sites.

3.3 Evaluation from the System

3.3.1 Information Disclosure

Our proposed model is based on number of disclosure of information. Multiple attributes to evaluate are prepared and a seller selects attributes based on his/her strengths. If he/she is good at packing, he/she can choose the "Package" as the evaluated attribute. On the other hand, if he/she does not want to disclose his weakness, he/she can omit the attribute to be evaluated. To design a desirable mechanism in evaluation, we set a control value based on number of information disclosure. When a seller changes five attributes from four attributes to be evaluated, the system gives an incentive points to the seller. Namely, if the seller discloses more attributes, the incentive points are given in proportion. Thus, he/she sets up a lot of attributes to get many incentive points. And also, incomplete information reduce from the shopping site. However, if he/she does so, he/she needs to be careful in each activity on a trade. if a seller provides an item's information by pictures and explanation, a risk on trade is decreased [10][11].

3.3.2 Cumulative Extra Point

Here, we define an experience value based on the cumulative number of trades for each seller. In existing evaluation systems, the score/rating of evaluation is calculated simple cumulative trading experience. For example, when a seller has 30 positive rating without any negative rating and he/she gets a positive rating in a subsequent trade, his/her score becomes 31 rating. However, we propose an appreciate model for outstanding sellers. The outline of the model is that the system gives an extra point for a seller who continues a lot of trading without negative rating from buyers. On the other hands, once he/she gets a negative point, the cumulative number goes back to the start. For example, when a seller has cumulative 100 positive rating without any negative rating and he/she gets a positive rating in a subsequent trade, the system give some extra score automatically. Thus, the marketplace positions outstanding sellers apart from the rest.

4. EXPERIMENTS

We conducted experiments to measure our proposed model. When the system changed the evaluation depending on the number of evaluation attributes, we searched the market conditions where buyer can have dealings with confidence. In the market conditions, we configured (1) Buyer takes precedence elements in dealing, (2) Buyer has a impression to concern the evaluation for seller when buyer looks at multiple attributes, (3) About the number of each buyer type defined by (1) and (2). In the definition of (1), the experiments assumed three buyer types including price-oriented (PO), evaluation-oriented (EO), and neutral buyers (N) in the marketplace. Price-oriented buyers prefer low price item rather than rating of evaluation to decide a seller to trade. Evaluation-oriented buyers have a trend to choose sellers with rating of evaluation rather than item's price. Neutral buyers have both above features. In the definition of (2), we set that buyer has a good impression on the specific number of evaluation attributes and gives seller higher one level rating. In the definition of (3), we investigated how many there are buyer types defined because we didn't know it exist in actual marketplace.

4.1 Survey

We surveyed fifty-seven peoples in order to set parameters in our experiments. We asked two questions. **Question (1)** Which do you prefer buying the item in e-commerce, products low prices, height of seller's rating or both low price and high evaluation? **Question (2)** When there are various sellers disclosing some evaluation attributes between 1 to 10, how many evaluation attributes are you desirable?

Table 2 shows the questionnaire result regarding buyer's preference in online market. From questioner's answer in the question (1), most of them are interested in a feature both low price items and high evaluation sellers. Table 3 shows the questionnaire result regarding buyer's impression about number of evaluation attributes. From questioner's answer in the question (2), many questioners wish the evaluation attributes is around five. We involved the distribution of each buyer type gotten by the above result into the experiments.

Table 2. Questionnaire Result (1)

Priority	Item Price	Evaluation	Both	Other
Totals	13	4	39	1

Table 3. Questionnaire Result (2)

Attributes	1	2	3	4	5	6	7	8	9	10
Totals	3	0	5	5	20	10	3	5	0	6

4.2 Setting

In the marketplace, rating of evaluation is rated through 1 to 5 of integers. Item's price is assumed between \$400 and \$600 chosen by a normal distribution on distribution value 50. The average of price of sold items is \$500. We assume three types of preferences in which buyers have. First, if buyer has the preference about price of item, the threshold of decision-making D_p is shown as equation (1). If P_s is larger than the equation, buyer trades with a seller who deals in at the lowest price out of candidates.

$$D_p : p - \frac{e}{10}$$

Second, if buyer has the preference about seller's evaluation, the threshold of decision-making D_e is shown as equation (2). If E_s is larger than the equation, buyer trades with a seller who deals in at the highest rating out of candidates.

$$D_e : \frac{500 - p}{10} + e$$

Third, if buyer is neutral for price and seller's evaluation, the threshold of decision-making D_n is shown as equation (3). If E_s /3-P_s /500 is larger than the equation, buyer trades with a seller who deals in at the highest value (than threshold value) out of candidates.

$$D_n : \frac{e}{3} - \frac{p}{500}$$

p indicates item's price and e indicates rating of evaluation. P_s indicates item's price shown by seller. E_s indicates seller's rating of evaluation. In the setting of experiments, four types of trends of evaluation are assumed with number of evaluated attributes. The number of evaluated attributes is between 1 and 10.

We assume four types of about evaluation given for seller. The following is detail of each evaluation type. **(A)** Average of evaluation value monotonically increases when the number of evaluated attributes increase. **(B)** When the number of evaluated attribution increases, the average of evaluation value exponentially increases. **(C)** When the number of evaluated attribution increases, the average of evaluation value increases with marginal decreasing. **(D)** When the number of evaluated attribution is around 5, it tends for buyers to give high rating like normal distribution. Figure 1 shows types of evaluation used in experiments. Horizontal axis shows the number of evaluation attribute, vertical axis shows the distribution of average evaluation. These evaluation types include both the rating given by buyer and the extra point from number of evaluation attributes.

Table 4 is a setting of 4 cases of experiments. Buyer's preferences are shown as PO, EO, and N. PO indicates the buyer's preference in which he/she has a price-oriented preference. EO indicates the preference in which he/she has a evaluation-oriented preference. N indicates a neutral buyer who has a preference both price and evaluation. In cases 1 and 2, we assume there is same number of types of buyers in the market. In cases 3 and 4, the rate of buyer's preferences are respectively used from our survey result shown in Table 2. EP indicates the condition where the number of attributes effect buyer's input to evaluate. When EP=0, number of evaluation attributes are not effected in an evaluation by buyers. When EP=1, some buyers give high rate when the number of attribute to be evaluated is same as their preferences shown in Table 3. For example, when a buyer prefer that the number of attributes is 6, he/she give a high rate if the trader provides 6 attributes to be evaluated. In the experiment, we assume that buyer gives 1 additional rate in such case.

Table 4. Experiment Setting

Experiments	Number of Buyer's Type
Experiments (A), (B), (C), (D)	Case1: PO=100, EO=100, N=100, EP=0 Case2: PO=100, EO=100, N=100, EP=1 Case3: PO=69, EO=21, N=210, EP=0 Case4: PO=69, EO=21, N=210, EP=1

Result of experiments shows the average of rate of successful trade in 1000 trials. We assume that there are three hundred potential buyers and one hundred potential sellers to trade.

4.3 Result of Experiment

Experiment results show in Figure 2 to 5. Each experiment condition is respectively employed the evaluation types (A), (B), (C), and (D) shown in 5.2. Horizontal axis indicates the number of evaluation attribute, vertical axis indicates the rate of successful trade in graphs.

4.3.1 Experiment (A)

In this experiment, the type (A) in 1 is used as buyers trend. Figure 2 shows the result of experiment on a setting of Experiment (A) in Table 4. When the each buyer type exists respectively the same rate, the transaction success rate is flat in the number of each evaluation attribute. On the other hands, when we employ a condition of cases 3 and 4, successful trade rate is high when the number of evaluation attribute is between 5 and 10. This means that seller's best strategy is to declare 5 or more than 6 attributes to be evaluated.

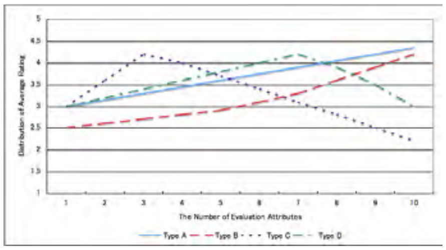


Figure 1. Evaluation

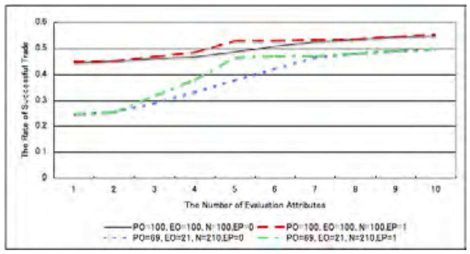


Figure 2. Experiment (A)

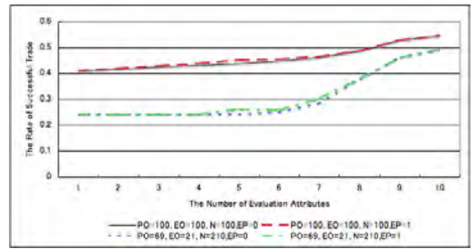


Figure 3. Experiment (B)

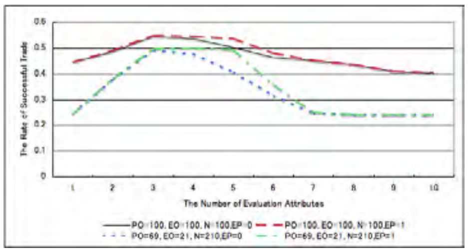


Figure 4. Experiment (C)

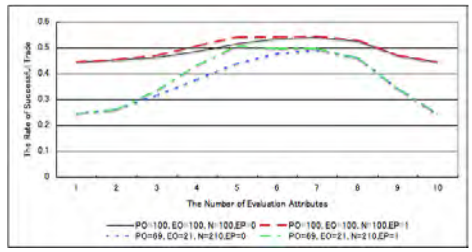


Figure 5. Experiment (D)

4.3.2 Experiment (B)

In this experiment, the type (B) in 1 is used as buyers trend. Figure 3 shows the result of experiment on a setting of Experiment (B) in Table 4. When the average of evaluation given by buyers and the system is three or less (see the type (B) curve in Figure 1), effects of the impression value is low.In cases 3 and 4, the rate of successful trade is extremely low. Namely, the best strategy for sellers is to provide a lot of attributes to be evaluated.

4.3.3 Experiment (C)

In this experiment, the type (C) in 1 is used as buyers trend. Figure 4 shows the result of experiment on a setting of Experiment (C) in Table 4. The successful trade is high on the number of evaluation attributes between 3 and 5. On the other hands, in cases 3 and 4, the number of successful trade is quite low on the number of evaluation attributes between 7 and 10. This means that seller's best strategy is to prepare 4 attributes or around 4 to be evaluated.

4.3.4 Experiment (D)

In this experiment, the type (D) in 1 is used as buyers trend. Figure 5 shows the result of experiment on a setting of Experiment (D) in Table 4. In the Cases 1 and 2, successful trades is the highest between 5 and 8. Considering actual tradings, seller's best strategy is to prepare attributes between 6 and 7 to be evaluated.

5. CONCLUSION

In this paper, we designed an evaluation model in which sellers become to have an incentive to disclose a lot of information of item and themselves. By using our proposed method, users evaluate more precisely sellers because our proposed method provides concrete criteria to be evaluated. Our model is based on multiple attribute evaluation including evaluation from buyer and system. System gives extra point based on the number of evaluation attributes set by sellers. Even though a seller is good at packaging, the system discounts the rating as a penalty when he/she chooses only one attribute "packaging" as detailed rating. Because our model makes an effect to promote information disclosure, to reduce incomplete, and to decrease asymmetric information. Our future work includes to analyze and to model a situation where buyer's preferences to evaluate dynamically changes.

6. REFERENCES

[1] Hiroyuki Takahashi, Yukihiro Usui, "The Effect of Reputation Systems on a network auction", Japan Science and Technology Information Aggregator, Electronic pp.63-68 (2003).

[2] Radu Jurca. Truthful reputation mechanisms foronline systems. PhD thesis, EPFL, 2007.

[3] Reid Kerr and Robin Cohen. Smart cheaters do prosper: defeating trust and reputation systems, 8th International Conference on Autonomous Agents and Multiagent Systems, 993-1000, 2009.

[4] Yukihiro Usui, Hiroyuki Takahashi, Noriaki Yoshikai, ”Study on the Effect of Reputation System as Trust Evaluation in Network Community”, IEICE technical report. Social Implications of Technology and Information Ethics 102(505):19-24, 2002.

[5] http://www.yahoo.co.jp/

[6] http://www.rakuten.co.jp/

[7] http://www.amazon.co.jp/

[8] Miho Nojima, Yoko Takeda, "Typology of Risk Reduction Mechanism in Electronic Commerce", ITME Discussion Paper No.21 (1999)

[9] Miho Nojima, Jiro Kokuryo, Junjiro Shintaku, Yoko Takeda, "Two Strategies in Internet Auction", ITME Discussion Paper No.48 (2000)

Facilitating Better Negotiation Solutions using AniMed*

Raz Lin[1], Yehoshua Gev[1] and Sarit Kraus[1,2]
[1] Department of Computer Science
Bar-Ilan University
Ramat-Gan, Israel 52900
{linraz,sarit}@cs.biu.ac.il

[2] Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742 USA

ABSTRACT
Mediation is an important paradigm for dispute resolution. If done properly, it can lead to "win-win" situations and benefit all parties. Thus, the advantage of designing a proficient automated mediator capable of interacting with people during their negotiations is of great importance. Yet, succeeding in this task is difficult due to the diversity of people and their bounded rationality. To be successful, the mediator must take this into account, and propose solutions deemed relevant, or otherwise, lose the focus and trust of the negotiators. In this paper we present *AniMed*. *AniMed* is an automated animated mediator, incorporated with a novel proposal generation strategy, aimed to increase the social benefit of the negotiating parties. To validate the benefits of using *AniMed* in negotiations, experiments were conducted with more than 100 people negotiating with each other. The results demonstrate the significant increase both in the social welfare and the individual utilities of both parties, compared to negotiations in which another state-of-the-art automated mediator or no mediator was involved.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms
Bilateral Negotiations

Keywords
bilateral negotiations, automated mediation, incomplete information

*This research is based upon work supported in part under NSF grant 0705587 and by the U. S. Army Research Laboratory and the U. S. Army Research Office under grant number W911NF-08-1-0144.

The Fourth International Workshop on Agent-based Complex Automated Negotiations (ACAN). Taipei, Taiwan, May 2011

1. INTRODUCTION
Negotiations are procedures for resolving opposing preferences between two or more parties, by means of discussion. The goal is to reach an agreement, i.e. a mutually accepted solution, without resorting to a struggle. Mediation, which is the involvement of a third party in the negotiation process, dates back to Ancient Greece [11] and has evolved to assist the negotiating parties in reaching beneficial solutions and increasing their social welfare. In many occasions, the mediator does not have the authority to impose a solution on the parties or the power to compel them to uphold the agreement reached (unlike arbitration), and the mediator is usually neutral (unbiased) and objective. This emphasizes the importance of a successful mediation, and thus nowadays it is widespread for dispute resolution ([12], Chapter 2).

Automated mediators, intelligent agents that take the role of an active mediator in the negotiation process, can play an important role in the mediation process between people. They offer a discrete, impartial facilitator that might be more trusted than a human one. The computational resources of automated mediators may also be more useful when incomplete information exists and there is uncertainty regarding the preferences of the parties, as the difficulty for a human mediator only increases. Yet, the use of automated mediators is far from widespread, perhaps due to the difficulties in bridging between people, or due to the computerized (perhaps "cold") nature of them.

In this paper we introduce *AniMed* – a domain-independent automated vivid and animated mediator designed to improve the social welfare of people in bilateral negotiations. *AniMed*, an English speaking avatar, interacts with people who negotiate by means of a video-conference. *AniMed*'s novel design allows it to propose solutions that are in the context of the current negotiation state. This strategy differentiates it from other automated mediators found in the literature. Another original feature implemented in *AniMed* is its capability to propose partial solutions, and by doing so it provides the negotiators with the option to incrementally strive for a beneficial solution. Moreover, the strategy incorporated in *AniMed* does not rely on the structure of the utility function of both negotiators, but rather constructs a preference relation between the possible solutions. Thus, *AniMed* has a generic strategy mechanism, allowing it to be matched and mediate proficiently with many possible types of negotiators without any restriction to any specific domain. Lastly, as *AniMed* was built on top of GENIUS, a generic negotiation framework [7], it will be available for the public and can be modified and used in numerous domains and settings.

AniMed was evaluated in experiments with more than 100 people who negotiated face-to-face on a neighbor dispute domain by means of video-conferences. The negotiation involved uncertainty with respect to the utility values of opposing parties. This uncertainty was also shared by the mediator, that had information solely on the preference relation between the issues under negotiation. *AniMed* significantly increased the individual utility score and the social welfare, measured by the sum of utilities, of both negotiators, compared to experiments in which another state-of-the-art mediator or no mediator were involved. The results also indicate that while people are content with the agreements they achieve without any mediator involved, better agreements can be achieved when *AniMed* is present, which only emphasizes the benefits of using it in human-to-human negotiations. The animated design of *AniMed* and the structure of the experiments was also motivated by findings of Nass and Moon [9], with respect to human-human versus human-computer interaction.

The rest of the paper is organized as follows. Section 2 provides an overview of automated mediators. Section 3 describes the negotiation context, followed by Section 4 that presents the design of *AniMed*, including the user-interface design. Section 5 describes the experimental setting and methodology and reviews the results. Finally, Section 6 provides a summary and discusses future work.

2. RELATED WORK

Few automated mediators are mentioned in the literature. Some are discussed in the context of online dispute resolutions, which are mostly alternative services to litigation. For example, eBay's resolution center[1] tries to facilitate the resolution of conflicts between buyers and sellers.

A number of negotiation support systems are also described in the literature. Family Winner [1], for example, is software that assists divorcees to rationally negotiate their disputes. It does this by advising rational options for trade-offs of assets between opposing parties. However, it is focused on a single domain and cannot be generalized. The HERMES system [6] is a collaborative decision support system that acts as an assistant and advisor by facilitating communication and recommending solutions to members of a decision makers group aiming at reaching a decision. It uses an argumentation framework that provides an issue-based discussion forum [5] whereby users can propose and discuss alternative solutions. Like HERMES, *AniMed* uses the issue-based discussion approach, yet *AniMed* is implemented to support face-to-face negotiation, and not as a collaborative decision support system.

PERSUADER [13] is a computer program that acts as an automated labor mediator in hypothetical negotiations relying on Case-Based Reasoning methods (i.e., logic formulation of the problem). PERSUADER is topic-embedded, and requires data from previous negotiation sessions to reason. In addition, it employs manipulation methods as a mean of manipulating the parties. In contrast, *AniMed* enables the parties to reach satisfactory agreements without the need to resort to manipulations and without the need of a historic database. In addition, unlike PERSUADER, *AniMed* was evaluated with people.

e-Alliance [2] is an automated mediator that offers support for multi-issue, multi-participant (different partners can be involved) and multiple-cycle (cycles of proposals and counter proposals over the same set of attributes) negotiations. These characteristics make the facility flexible enough for use in different domains. While e-Alliance was developed for agent-agent interactions, we are interested in the problem of human-human interactions.

Olive *et al.* [10] formalize the functionalities an automatic mediator should be able to activate when operating in a multi agent environment. However, as *AniMed* operates in bilateral human environments, some of the defined functionalities are not implemented. For example, the storage of the dialog protocol and its specifications, as well as resolving disputes over the protocol's rules, are irrelevant since *AniMed* uses a pre-defined protocol.

AutoMed [4] is an automated mediator that most resembles our proposed mediator. *AutoMed* monitors the exchange of offers and actively suggests possible solutions, during the negotiation process. It uses a qualitative model for the negotiator's preferences, without past knowledge. The suggestions it makes are basically Pareto-optimal solutions that maximize the social welfare of the negotiating parties. *AutoMed* was evaluated with human negotiators, who negotiated using a computer system, by exchanging offers selected from drop-down lists. *AutoMed* participates as a third-party that sends suggestions via the system. However, *AutoMed* has its limitations. Mainly it does not suggest incremental (partial) solutions nor does it provide explanations for its suggestions. Moreover, *AutoMed* constrains the negotiators to negotiate through the system, while a more natural approach would be to negotiate face-to-face. These drawbacks are nonexistent in *AniMed*, allowing it to generate more satisfactory agreements that are deemed more relevant by the negotiating parties.

3. NEGOTIATION PROBLEM DESCRIPTION

We consider the problem of a proficient automated mediator as a key to improving the performance of two human negotiators, who strive to reach an agreement on conflicting issues. The mediator is situated in finite horizon bilateral negotiations with incomplete information between two people. The negotiation consists of a finite set of multi-attribute issues and time constraints. The incomplete information is expressed as uncertainty regarding the utility preferences of the opponent, as explained below.

The negotiation can end when (a) the negotiators reach a full agreement, (b) one of the negotiators opts out, thus forcing the termination of the negotiation with an opt-out outcome, or (c) a predefined deadline, denoted dl, is reached, whereby, a status quo outcome, denoted SQ, is implemented.

[1] `http://resolutioncenter.ebay.com/`

Given a set of issues, $I = \{I_1, I_2, \ldots, I_m\}$ and a set of values $dom(I_j) = \{v_1^j, v_2^j, \ldots, v_{b_j}^j, \perp\}$ for each $I_j \in I$ (since we allow partial solutions to be proposed, $\perp \in dom(I_j)$) and let O be a finite set of discrete values for all issues $(I_1 \times I_2 \times \ldots \times I_m)$. A solution is denoted as a vector $\vec{o} \in O$, where its utility is calculated as a sum of its values. While the utility is known to each negotiator, it is unknown to the mediator. A full order, $\prec$, exists over the values of $dom(I_j)$ and this is the only information shared between the negotiators and the mediator, that basically captures the preference values in the sense of which is "more important than".

It is assumed that the negotiators can take actions during the negotiation process until it terminates. If a partial agreement was accepted it is then implemented. While we did not incorporate time costs in our settings, they can be easily generalized to include time costs which are assigned to the negotiators' utilities. In each period each agent can propose any number of possible agreements, and the other agent can accept the offer, reject it or opt out. Each agent can either propose an agreement which consists of all the issues in the negotiation, or a partial agreement.

To make the problem more realistic the negotiation we consider a setting in which the negotiation is done face-to-face using a video conference and a negotiation system. Thus, the parties negotiated freely and discussed the different issues until they arrived at potential solutions to agree upon.

The negotiation problem also involves incomplete information with regards to the preferences of the opponent. While each side knows its own utilities, the utilities of the other side are private information. Formally, we denote the utility of each side $l \in \{A, B\}$ as u_l, and $u_l : \{O \cup \{SQ\} \cup \{OPT\}\} \rightarrow \mathbb{R}$.

4. MEDIATOR DESIGN

The design of *AniMed* is built on top of the GENIUS infrastructure, which is an integrated environment for supporting the design of generic automated negotiators [7]. This environment is rich and supports bilateral multi-issue and multi-attribute negotiations, both with human counterparts and automated agents. An example snapshot of a negotiation interface is given in Figure 1. The focus of the research was to design an automated formulating mediator (as opposed to a manipulative one). That is, the agent tries to propose solutions and help the negotiators reach a mutually agreed outcome. *AniMed* is not topic embedded, allowing it to be used in many scenarios, and it is aimed to be proficient in bilateral negotiations involving people.

AniMed was implemented using two main considerations. First, a proficient strategy was used to enable it to generate offers deemed relevant by the negotiating parties. To achieve this, *AniMed* utilizes recent offers proposed by the negotiators when generating its own offers, thus centering its offer on the current context of the negotiation. The second consideration involves its user interface design. *AniMed* was implemented with a rich animated interface to make it appealing and user friendly for people (see Figure 2).

The motivation behind the strategy design of *AniMed* was to generate offers that would maximize the social welfare of

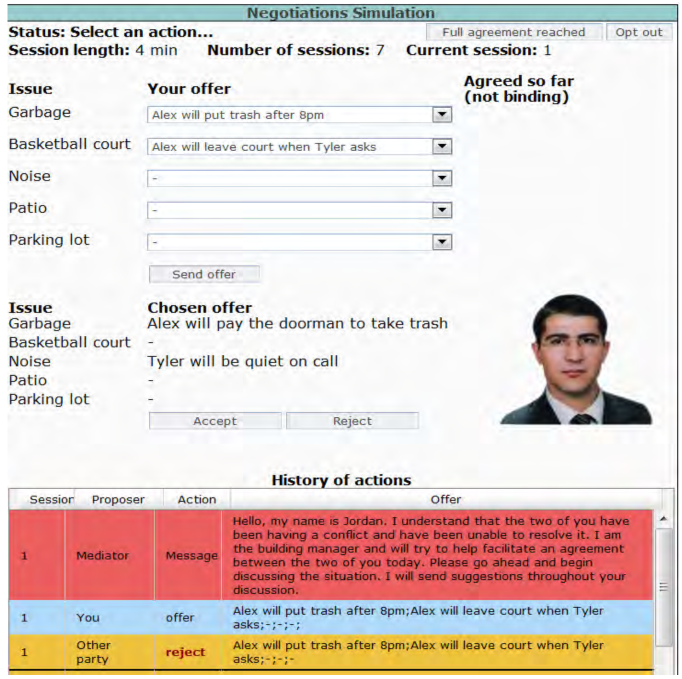


Figure 1: An example of a negotiation snapshot using Genius.

both negotiators. However, this is a difficult task due to conflicting interests between the negotiators. To overcome this, *AniMed* starts proposing only after both negotiators have proposed or accepted an offer in the past. It uses this information to try to find a set of solutions that can still increase both negotiators' utilities. One of its strengths is its ability to provide a solution only on a subset of the issues under negotiation, allowing the negotiators to incrementally improve the final solution. In addition, to prevent the negotiators from labeling its offers as irrelevant, *AniMed* does not propose any offer if it is identical to the last offers made by the negotiators.

The strategy used by *AniMed* consists of five steps, which we describe hereafter using an example to better illustrate it. Assume that in a given negotiation domain there are two agents, A and B, and 7 possible solutions. Also assume agent A and B just proposed solutions indexed 3 and 5, respectively. Table 1 lists the information about the domain and the steps taken by *AniMed* to decide on a solution to propose.

The first step in the algorithm used by *AniMed* is taken before the negotiation starts. While the utilities of the solutions are private information of each negotiator and unknown to the mediator, *AniMed* uses a linear function, $order(\cdot)$, to obtain an ordinal scale of all solutions. Each issue I_j is given a cost, λ_{I_j}, which is its ranking compared to all other issues, based on the preference relation between the issues. Each issue's value is also ranked based on the preference relation between the possible values of the given issue. Then, the mediator multiplies the costs of issues and values to obtain the linear preference relation. Note that this order may be different from the actual order of the values of the ne-

Solution Idx	$order_A(\vec{o})$	$order_B(\vec{o})$	joint order	diff order
1	7	1	8	6
2	6	2	8	4
3	5	3	8	2
4	4	4	8	0
5	3	6	9	3
6	2	5	7	3
7	1	7	8	6

Table 1: A sample domain for choosing a solution to propose. For each negotiator, A and B, the possible solutions are ordered by her own preferences. The last offers made by the negotiators are marked in bold.

gotiators. Then, during the game play, *AniMed* chooses its suggestions based on these orderings and on the last offers made by the negotiating parties, which are marked in bold in Table 1.

The next two steps are motivated by the strategy of *AutoMed*. The second step in *AniMed*'s strategy is to discard all solutions that, for each party, have a lower ranking than her last proposal. Thus, *AniMed* removed solutions #1, #2 (higher ordering for agent A) and #7 (higher ordering for agent B) while it kept the four other offers. Then, in its third step, *AniMed* searches for any non-Pareto optimal offers and removes them as well. In our example, one of the solutions was non-Pareto optimal (#6).

In the fourth step *AniMed* orders all remaining solutions based on the following criteria. First, it orders them based on the solutions' joint ordering (that is, the sum of both orderings, marked as "joint order" in Table 1). If the solutions have the same joint ordering, it compares them to previous solutions proposed by each negotiator. However, to allow *AniMed* to propose solutions which are in context with the solutions previous suggested by the negotiating parties, solutions that are more similar to previously suggested solutions, measured by the number of similar values between the solutions, are ordered higher. If there are still solutions with the same rank, it orders them based on the absolute difference in their ordering (marked as "diff order" in Table 1). Algorithm 1 describes the pseudo-code of the algorithm for generating a full proposed solution.

AniMed now has a full solution that it can propose. However, from preliminary experiments, we observed that the dynamics of the negotiation mainly involves partial agreements. Thus, the fifth step in *AniMed*'s strategy is to generate partial solutions that could still benefit the negotiators. *AniMed* incorporates two mechanisms for generating partial solutions. The first is based on joint-value issues. That is, issues with values that are estimated as generating higher utilities for both parties, based on their orderings, can be suggested by the mediator. The second is based on a trade-off between the issues. This is done by calculating the distances between the ordering of given issues, denoted by "diff order" in Table 2. *AniMed* then continues to calculate the difference between the orderings of each

Algorithm 1 Generating A Full Proposed Solution

for all $\vec{o} \in O$ **do**
  Insert $\vec{o}$ to $\textbf{OrderedList}_A$, $\textbf{OrderedList}_B$
  Using $order_A(\vec{o})$, $order_B(\vec{o})$
end for
if Both sides interacted **then**
  for all $\vec{o} \in O$ **do**
    if $order_A(\vec{o}) < lastOffer(A)$ **then**
      remove $\vec{o}$ from $\textbf{OrderedList}_A$
    end if
    if $order_B(\vec{o}) < lastOffer(B)$ **then**
      remove $\vec{o}$ from $\textbf{OrderedList}_B$
    end if
  end for
  OffersList = Intersect($\textbf{OrderedList}_A$, $\textbf{OrderedList}_B$)
  OffersList = ParetoOffers(**OffersList**)
  Define jointOrder($\vec{o}$) = $order_A(\vec{o})$ + $order_B(\vec{o})$
  Define diffOrder($\vec{o}$) = abs($order_A(\vec{o})$ - $order_B(\vec{o})$)
  Sort **OffersList**
  Using jointOrder($\vec{o}$)
  Then SimilarityToRecentOffers($\vec{o}$)
  Then diffOrder($\vec{o}$)
end if

Issue Idx	$order_A(issue)$	$order_B(issue)$	diff order
1	1	4	3
2	2	2	0
3	3	3	0
4	4	5	1
5	5	1	-4

Table 2: An example of deciding the trade-off between issues. For each negotiator, the order of importance of an issue is determined by the maximum value she can achieve from that issue.

pair of issues, that is, $\forall i, j \in I, (order_A(i) - order_B(i)) - (order_A(j) - order_B(j))$. For example, the result for the pair $\langle Issue_1, Issue_5 \rangle$ is $3 - (-4) = 7$. The higher the sum, the better candidate it is for selection in the partial solution in order to allow trade-offs. This evaluation is motivated by our belief that people tend to perceive issues as "important" or "not important", thus they do not use the full possible ranking of solutions. The aforementioned evaluation tries to capture this observation, that is, a pair of issues that would be deemed "important" and "not important" to one of the negotiators, and the opposite to the other party.

From those possible partial solutions, *AniMed* tries to make proposals based on trade-offs between issues, or joint-value issues, that were agreed upon or discussed by the negotiators.

Another consideration implemented in *AniMed* is a simple argumentation mechanism to try to convince the parties why the proposed solution suggested by *AniMed* should be considered. When AniMed proposes a solution it attaches a text message stating that if the negotiators make the suggested trade-off they can achieve higher scores (the text is slightly different if the suggested solution includes issues that were previously agreed upon by the parties or simply discussed).

An additional approach incorporated in *AniMed* relates to its presence during negotiations. In order to compel people

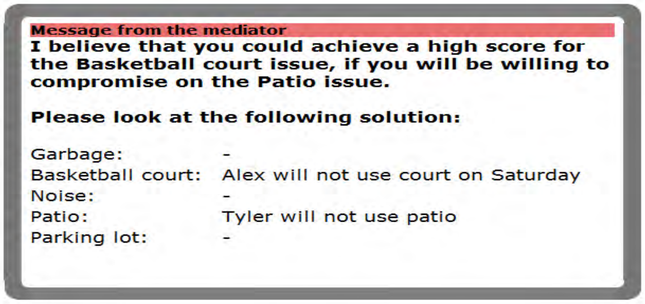


Figure 2: *AniMed* avatar example.

to listen to the mediator's proposals, whenever it proposes a solution, it takes over the entire screen so people cannot conceal or ignore it. Moreover, the mediator was implemented as an English (translation to native language was given in the text) speaking avatar (see Figure 2), using a commercial text-to-speech engine, to convey a more "humanized" appearance and a less distant and computerized one.

5. EXPERIMENTS

We matched people on a given domain using the GENIUS environment, an integrated environment for supporting the design of generic automated negotiators [7]. The two negotiators were given a task to negotiate a beneficial agreement. Four different experiments were conducted using the same domain, in order to compare the proficiency of *AniMed*. One experiment involved matching people without any mediator. In another experiment we matched two people with a simple automated mediator, *AutoMed*, suggested by [4]. Chalamish and Kraus demonstrated that this mediator enables the negotiators to achieve more satisfactory agreements in environments where only messages are exchanged.

Finally, we matched people in a setting which included our proposed mediator, once while providing them a facilitation mechanism that provided them a "negotiation calculator" which enabled them to calculate the utilities of each solution at any given time, and once without that mechanism. They played in only one of the experiments in order not to bias the results. We begin by describing the domain which was used in all the experiments and then continue with the experimental methodology and the results.

5.1 The Negotiation Domain

For the negotiation domain we chose a neighbor dispute domain. In this domain, a negotiation takes place between two tenants, Alex and Tyler, due to a neighbors' dispute. Both negotiators need to negotiate in order to resolve the dispute, or otherwise be forced to undergo a lengthy and costly dispute resolution process. The issues under negotiation are:

1. **Trash**. This issue dictates the solution to the fact that Alex puts its trash every morning on the stairwell, making Tyler angry as it attracts flies and stinks. The possible values are (a) Alex will continue to put trash on the stairwell, (b) Alex will put trash at 5pm, (c) Alex will get a friend to take out the trash, (d) Alex will pay the doorman to take out the trash, or (e) Alex will put trash after 8pm.

2. **Basketball court**. This issue describes how the basketball court will be shared between Alex and Tyler on Saturdays. The possible values are (a) Alex will continue to use the court on Saturdays at any given time, (b) Alex will use court for two hour only, (c) Alex will use court for one hour only, (d) Alex will leave court when Tyler asks him to, or (e) Alex will not use the court on Saturdays.

3. **Noise**. The noise issue tries to resolve the problem of Tyler making noise at nights, disturbing Alex's sleep. The possible values are (a) Tyler will be quiet after 11pm, (b) Tyler will be quiet after 12am, (c) Tyler will be quiet after 1am, (d) Tyler will be quiet upon request by Alex, or (e) Tyler will continue to be loud.

4. **Patio**. This issue describes how the patio will be shared between Alex an Tyler. The possible values are (a) Tyler will not use patio, (b) Tyler will use patio for one hour every other night, (c) Tyler will use patio for one hour every night, (d) Tyler will use patio for two hours every night, or (e) Tyler will continue to use patio whenever he wants to.

5. **Parking lot**. This issue describes the resolution for using the parking lot by Tyler's guests. The possible values are (a) Call the police to give tickets or tow away unauthorized cars, (b) Alex and Tyler will try to recruit other residents to move unauthorized cars, (c) Alex and Tyler will donate money to install "non-parking" signs , (d) complain to the owner about the situation, or (e) Tyler and Alex do nothing.

In this scenario, a total of 3,125 possible solutions existed ($5 \times 5 \times 5 \times 5 \times 5 = 3125$). The scenario was symmetric for both negotiators, in the sense that the negotiators could compromise and make tradeoffs between the issues and the gains and losses were equivalent. On one of the issues both negotiators received the same utility. On two other issues the more one gained, the less the other gained. These two issues had the same scale in the utility scores. For the last two issues, the negotiators could use tradeoffs between the values of both issues to gain the same utilities[2]. The utility values ranged from 0 to 1,000 for both negotiators, where the Pareto-optimal solution generated a utility of 720 for both.

Each turn in the scenario equated to two minutes of the negotiation, and the negotiation was limited to 28 minutes.

[2]Detailed score functions for the domain can be found at `http://u.cs.biu.ac.il/~linraz/Papers/neighbors-utilities.pdf`

If the negotiators did not reach an agreement by the end of the allocated time, the negotiation ended and both tenants would be required to undergo a costly dispute resolution session. This outcome was modeled for both agents as the status quo outcome. Each negotiator could also opt-out of the negotiation if it felt that the prospects of reaching an agreement with the opponent were slim and that it was no longer possible to negotiate. The status quo value equaled the opting out value and which was 280 for both negotiators.

5.2 Experimental Methodology

The human negotiators accessed the negotiation interface via a web address. The negotiation itself was conducted as follows. Using a video conference the two negotiators negotiated face-to-face on the different negotiable issues. Since the focus of the research was on the strategy of the automated mediator, natural language processing (NLP) was beyond the scope of our research, and thus we required the negotiators to submit their proposals also using the negotiation system. This allowed the information to be processed by the automated mediator. Nonetheless, the negotiation itself was not constrained and was employed via a face-to-face video conference. The acceptance or decline of the offer was also done using the user interface. The mediator in turn sent proposed solutions to the parties via the animated avatar and the negotiation system.

We tested our agent against human subjects, all of whom are computer science undergraduates and graduate students. 104 human subjects participated in the experiments (52 pairs). A total of four sub-experiments were conducted, 13 pairs in each sub-experiment. The subjects did not know any details regarding the automated mediator with which they were matched, e.g., whether it was a human or an automated one and what type of strategy it used. The outcome of each negotiation was either they reached a full agreement, they opted out or the deadline was reached.

Prior to the experiments, the subjects were given oral instructions and were shown an instruction video regarding the experiment and the domain. The subjects were instructed to play based on their score functions and to achieve the best possible agreement for them.

5.3 Experiment Results

To verify the proficiency of *AniMed* we compared the final utility results in all experiments, as well as the number of proposals exchanged between the negotiators in each experiment. Lastly, we administrated questionnaires inquiring about the satisfaction of the negotiators from the outcome and their view on the helpfulness of the automated mediator.

Throughout this section, we also evaluate the significance of the results, compared to the results of *AniMed* without the facilitation mechanism. With respect to the utility values, the significant test was performed by applying the *t-test* on the results, which is a statistical hypothesis test in which the test statistic has a *t-distribution* if the null hypothesis is true. This test requires a normal distribution of the measurements ([3], Chapter 3). Thus, in our analysis it is used to compare the utility values of the negotiation parties in the different experiments conducted, which have continuous values. We applied the *Mann-Whitney U-test* on the results

		Alex	*Tyler*	*SW*
AniMed w/o facilitation	Average	723	665	1388
	Stdev	92	69	75
AniMed with facilitation	Average	735	669	1404
	Stdev	52	45	59
	p-value	0.35	0.43	0.28
AutoMed	Average	651	590	1241
	Stdev	80	103	145
	p-value	0.022	0.02	0.002
No Mediator	Average	645	595	1240
	Stdev	130	121	150
	p-value	0.045	0.041	0.002

Table 3: Average utility scores, standard deviations, social welfare (SW) and p-values in the different experiments. p-values are compared to experiments involving *AniMed* without the facilitation mechanism.

of all other parameters [8]. The *Mann-Whitney U-test* is a non-parametric alternative to the paired t-test for the case of two related samples or repeated measurements on a single sample, suitable for data without normal distribution (as in our case).

Table 3 summarizes the results of the individual utilities and the social welfare, measured by the sum of utilities of the negotiating parties (in our domain they are denoted as Alex and Tyler). First, we examined the final utility values of all the negotiations for each role, and the social welfare, measured by the sums of the final utility values. When *AniMed* was involved, the average utility for both negotiators was significantly higher (735 and 669 or 723 and 665 for Alex and Tyler with and without facilitation, respectively) than in any of the cases in which it was not involved (that is, with *AutoMed* – 651 and 590 for Alex and Tyler, respectively – or without any mediator – 645 and 595).

Comparing the sum of utility values of both negotiators when *AniMed* was involved to cases in which it was not involved, also reveals that the sum was significantly higher in cases in which *AniMed* was involved (1241 and 1240 with *AutoMed* or without any mediator, respectively, as compared to 1,388 and 1,404 with *AniMed*). These results were found to be significant (using the 2-sample *t-test*: $p < 0.002$ for both cases). It is interesting to note that though the utility scores were symmetric for both negotiators, on average Alex received higher scores than Tyler. When analyzing the results and videos we can see that there were two issues (noise and garbage) for which non-symmetric agreements were reached. We believe this was due to possible values of the issues and their scores. It seems that the *content* of the value caused subjects to choose them since they seemed reasonable enough, though other values could have generated higher utilities. For example, for the noise issue there were two values – being quiet after 1am, which yielded equal utilities for both Alex and Tyler, or being quiet after 12am, which yielded a higher utility for Alex, yet was preferred by both negotiators. It seems that the country where the negotiations were conducted, being quiet after 12am seemed reasonable enough to be chosen, even though it generated lower utilities for Tyler.

It is also noteworthy that in the two cases in which *AniMed* was involved, once with the facilitation mechanism, and once

	AniMed w/o facilitation	*AniMed* with facilitation	*AutoMed*	*No Mediator*
Average Proposals	7.38	6.46	5.77	6.38

Table 4: Average number of proposals exchanged.

	Outcome Satisfaction	p-value	Mediator's Helpfulness	p-value
AniMed w/o facilitation	3.29		1.08	
AniMed with facilitation	3.31	0.05	2	0.01
AutoMed	3.11	0.18	0.35	0.03
No Mediator	3.19	0.31	N/A	

Table 5: Average satisfaction levels (with 0 being the lowest and 4 the highest) and p-values in the different experiments. p-values are compared to experiments involving *AniMed* without the facilitation mechanism.

without, similar results were revealed without any statistical differences between them, both for the individual utilities of the parties and for the social welfare.

We then analyzed the number of proposals exchanged between the negotiating parties (see Table 4). More proposals were exchanged when *AniMed* was involved than in the other cases, though the differences were not statistically significant. We believe this could be due to the fact that when *AniMed* intervenes in the negotiation process it makes the parties aware of more resolution possibilities, which they later propose themselves.

Finally, we gathered the satisfaction levels of the negotiators from the final outcome they reached and their perception of how helpful the mediator was in reaching this outcome (see Table 5). The satisfaction levels ranged from 0 (lowest) to 4 (highest). The results significantly demonstrate that the negotiators perceived *AniMed* as more helpful than *AutoMed* ($p < 0.03$). Surprisingly, the negotiators were content with the final outcome in every experiment, and though the satisfaction level was slightly higher when *AniMed* was the mediator the difference was not statistically significant. This is in contrast to the fact that the negotiators achieved *significantly higher* utilities, both individually and combined, when *AniMed* was involved, compared to the other experiments. These results support our belief in the need and benefits of using mediators in negotiation settings when people are involved.

6. CONCLUSIONS

This paper presents *AniMed*, a novel automated mediator capable of proficiently interacting with people. The success in proficiently interacting with people has great implications on the outcome of the negotiations and allows the negotiating parties to maximize their revenues.

Experiments with more than 100 people demonstrated the benefits of *AniMed* compared to another automated mediator and to settings without any mediator. The fact that *AniMed* can be employed in any setting with the requirement of knowing only the structure of the preference relation between the issues, reflects on its generality and its prospects of becoming widespread and useful in numerous settings.

Future research will involve validating the results on additional scenarios, including ones with nonlinear utility functions and ones with a larger number of issues. We will also extend *AniMed* to present the negotiators with threats and the ability to enforce solutions and penalties. These features will extend the functionality and the richness of the mediator. Experiments are needed to validate whether these capabilities will still allow the mediator to be successful and whether better agreements can be achieved compared to the current design. Moreover, this kind of manipulative mediator can be used in interesting studies on the impact of different mediation styles on negotiations. In addition, mechanism for obtaining information from the video conference will facilitate the negotiation and will allow the negotiation turn more realistic.

7. REFERENCES

[1] E. Bellucci and J. Zeleznikow. Family winner: A computerised negotiation support system which advises upon australian family law. In *ISDSS2001*, pages 74–85, London, 2001.

[2] M. Bratu, J. M. Andreoli, O. Boissier, and S. Castellani. A software infrastructure for negotiation within inter-organisational alliances. In *AMEC-IV*, pages 161–179, Berlin, Germany, 2002.

[3] S. R. Brown and L. E. Melamed. *Experimental Design and Analysis*. Sage Publications, Inc., CA, USA, 1990.

[4] M. Chalamish and S. Kraus. AutoMed - an automated mediator for multi-issue bilateral negotiations. *JAAMAS*, 2011.

[5] T. F. Gordon, N. Karacapilidis, H. Voss, and A. Zauke. Computer-mediated cooperative spatial planning. In *Decision Support Systems in Urban Planning*, pages 299–309, 1997.

[6] N. Karacapilidis and D. Papadias. Computer supported argumentation and collaborative decision making: The HERMES system. *Information Systems*, 26(4):259–277, 2001.

[7] R. Lin, S. Kraus, D. Tykhonov, K. Hindriks, and C. M. Jonker. Supporting the design of general automated negotiators. In *ACAN*, 2009.

[8] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 1947.

[9] C. Nass and Y. Moon. Machines and mindlessness: Social responses to computers. *Journal of Social Issues*, 56(1):81–103, 2000.

[10] E. Oliva, P. McBurney, A. Omicini, and M. Viroli. Argumentation and artifacts for negotiation support. *International Journal of Artificial Intelligence*, 4(S10):90–117, 2010.

[11] D. Roebuck. *Ancient Greek Arbitration*. Holo Books The Arbitration Press, 2001.

[12] B. Starkey, M. A. Boyer, and J. Wilkenfeld. *Negotiating a Complex World: An Introduction to International Negotiation*. Rowman & Littlefield Publishers, 2 edition, 2005.

[13] K. P. Sycara. Problem Restructuring in Negotiation. *Management Science*, 37(10):1248–1268, October 1991.

Implementation of Collective Collaboration Support System based on Automated Multi-Agent Negotiation

Mikoto Okumura
Nagoya Institute of Technology
Gokiso-cho, Nagoya-shi, Aichi, 466-8555 Japan
okumura@itolab.mta.nitech.ac.jp

Katsuhide Fujita
Nagoya Institute of Technology
Gokiso-cho, Nagoya-shi, Aichi, 466-8555 Japan
fujita@itolab.mta.nitech.ac.jp

Takayuki Ito
Nagoya Institute of Technology
Gokiso-cho, Nagoya-shi, Aichi, 466-8555 Japan
ito.takayuki@nitech.ac.jp

ABSTRACT

Recently, discussions among many people about global warming and global product development have been increasing. Efficient collaborative support based on multi-agent systems is necessary to collect the huge number of opinions and reach optimal agreements among many participants. In this paper, we propose a collaborative park-design support system as an example of collective collaboration support systems based on multi-agent systems. In this system, agents elicit the utility information of users, collect many alternatives, and reach optimal agreements based on automated negotiation protocol. In particular, we focus on the steps for determining the attribute space and estimating the utility spaces of users in real world.

1. INTRODUCTION

Recently, discussions among many people about global warming and global product development are increasing. Efficient collaborative support based on multi-agent systems is necessary to collect huge number of opinions and reach optimal agreements among many participants. Many automated negotiation mechanisms are existed, however, the perfect utility functions of agents are assumed [1, 2, 3, 4, 5, 6, 7, 8]. In real world, it takes a lot of time to elicit the whole utility spaces of users.

In this paper, we propose a collective collaboration support system based on the multi-issue automated negotiation mechanisms [1, 2, 3, 4, 5, 6, 7, 8]. In this system, the agents elicit the utility information of users, collect many alternatives, and reach optimal agreements based on the automated negotiation protocol. Especially, we focus on the steps of deciding the attribute space and estimating the utility spaces of users in real world.

In this paper, we adopt a collaborative park-design support system as an example of a collective collaboration support system. Many users, like citizens and designers, should join the work to design parks. Many opinions and preferences of participants should be respected. Additionally, the designs of parks have some interdependent issue, for example, there are some dependence between the amount of playground equipments and the cost. In such a case, the automated negotiation protocol with issue-interdependency is effective [5, 6, 7, 8]. However, to apply the automated negotiation protocol with issue-interdependency, we need utility functions of users because most of the papers assumed the perfect utility functions of agents. In real world, it is impossible to elicit all the utility information of agents.

Our system estimates the interdependent multi-attributes utility functions of users based on users' evaluation of the designs generated by our system. In this paper, the utility function is composed of some simple fundamental functions. One fundamental function is defined by one user's evaluation of designs. The fundamental functions has a character that the utility grows low as the point is far from the sampling point corresponding to the design. The bumpy utility space is generated by combining the simple mound of the fundamental functions.

The remainder of this paper is organized as follows. First, we describe the outline of the collaborative park-design support systems. Next, we propose a new method of estimating the utility functions of users in real life. Third, we demonstrate some results of our method and conduct an experiment to evaluate the effectiveness of our system. Finally, summarizing our paper.

2. COLLABORATIVE PARK DESIGN SUPPORT SYSTEM

Figure 1 shows the outline of a collaborative system based on multi-agent system. The details of the system are followings.

`[Step1]` Collecting the opinions and preferences
The system decides sampling points and generates the alternative of the park design at the sampling point. After that, this system elicit the uses' preferences based on users' evaluations.

`[Step2]` Estimating utility functions
The system predicts the whole utility spaces based on the sampling points collected in the `[Step1]`. This estimation will be shown in the section3.

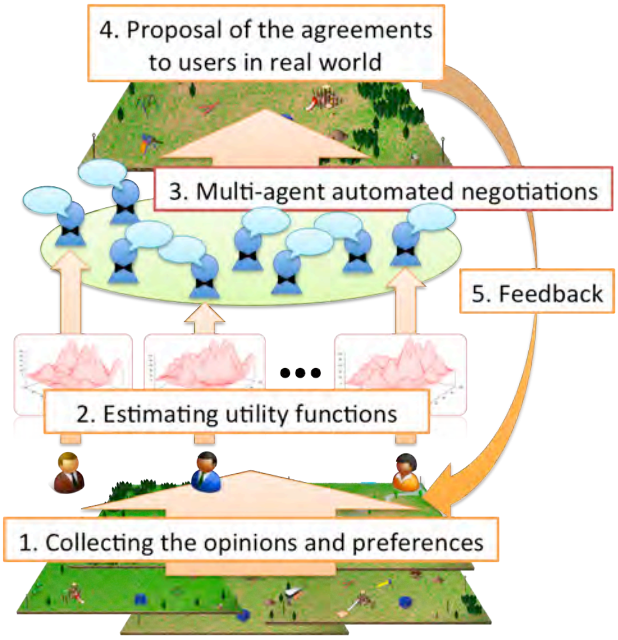


Figure 1: Collaborative public space design processes

`[Step3]` **Multi-agent automated negotiations**
The agent which is behalf of the users finds the optimal agreements by the automated negotiation protocol[1, 2, 3, 4, 5, 6, 7, 8].

`[Step4]` **Proposal of the agreements to users in real world**
The system generates a design from the agreement (a point on the attribute space) in `[Step3]`.

`[Step5]` **Feedback**
The system sends the design result(final alternative) in this round, and the users give a feedback. If the most of users agree to the final alternative, it is an optimal agreement. If the most of users don't agree to the final alternative, these steps are repeated(`[Step1]`∼`[Step5]`). But, our system doesn't have this step to be simple because our study and implementations of system based on multi-agent system are in its early stage.

3. METHOD OF ELICITING THE UTILITY SPACES

The method of eliciting utility functions corresponds to the `[Step1]`,`[Step2]` in the section2. This system generates the park designs automatically, receives the users' evaluations, and estimates the utility spaces.

We employ the parametric design because the parameters in the attribute space correspond one-to-one with park designs. In other words, our system can convert an agreement point in the automated negotiation to the park-design in real world.

In this paper, the utility function is composed of some simple fundamental functions. The fundamental functions has a

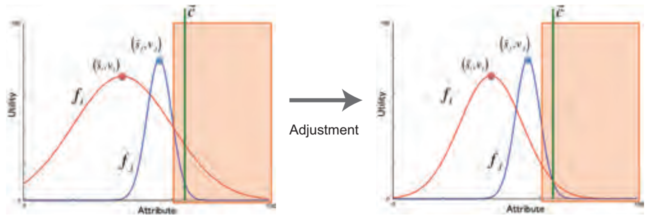


Figure 2: The Fundamental Function is Under the Other Fundamental Function (Case1)

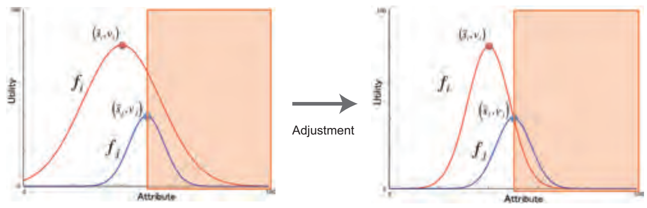


Figure 3: Most of the Fundamental Function are Under the Other Fundamental Function (Case2)

character that the utility grows low as the point is far from the sampling point. The bumpy utility space is generated by combining the simple mound of the fundamental functions.

3.1 Fundamental Function

Definition 1. *Fundamental Function*

$\mathbb{R}^+$ is a set of positive real numbers more than 0, $\mathbb{R}^*_+$ is a set of all positive real numbers. When i is an index, s_i shows a sampling point, d_i is the distribution of f_i and v_i is the evaluation value of $s_i(v_i, d_i \in \mathbb{R}^*_+)$. The fundamental function f_i is defined as a following expression.

$$f_i(\vec{x}) = v_i \cdot \exp(-\frac{(\vec{x} - \vec{s_i})^2}{d_i}) \qquad (1)$$

- The fundamental function is always more than 0 and a multi-dimensional space.

$$f_i : \mathbb{R}^{+n} \rightarrow \mathbb{R}^+$$

- The maximum of the fundamental function is equal to the evaluation value of the user.

$$\max f_i(\vec{x}) = v_i$$

- The maximum point of fundamental function means the sampling point.

$$\arg\max_{\vec{x}} f_i(\vec{x}) = \vec{s_i}$$

- The value of the fundamental function is smaller as it grows far from the sampling point.

$$||\vec{x_1} - \vec{s_i}|| > ||\vec{x_2} - \vec{s_i}|| \rightarrow f_i(\vec{x_1}) < f_i(\vec{x_2})$$

3.2 The combination of the fundamental functions

DEFINITION 2. *Utility Function*

When there are N sampling points; $(\vec{s_1}, .., \vec{s_N})$, the utility function U is defined as follows:

$$U(\vec{x}) = \max_{i=1,..,N} f_i(\vec{x}) \tag{2}$$

However, the definition 2 has two main problems as Figure 2, Figure 3 shows. In the left part of Figure 2, the sampling s_j don't work well because the function f_j is totally smaller than the function f_i. For instance, our system should employ f_j at the square area because the sample point of f_j (s_j) is closer to the square area than that of f_i (s_i). In the left part of Figure 3, the sampling s_j don't work well because most of the function f_j is smaller than the function f_i. For instance, our system should employ the function f_j at the square area in the Figure 2 because the point in the square area is closer to the sampling point of function f_j (s_j) than that of function f_i (s_i). Following two techniques resolve these two problems by modifying d_i which is the distribution of the fundamental function f_i.

METHOD 1. *A fundamental function is under other fundamental function (Case 1)*

This method adjusts the f_i as Figure 2 showing by modifying d_i. For instance, we assume that two different sampling points $\vec{s_i}, \vec{s_j}((i \neq j, \max f_i(\vec{x}) \geq \max f_j(\vec{x})))$ exist. If $f_i(\vec{s_j}) > f_j(\vec{s_j})$, then this method modifies d_i to satisfy $f_i(\vec{s_j}) = f_j(\vec{s_j})$ using the expression(3).

$$d_i = \frac{(\vec{s_j} - \vec{s_i})^2}{\ln \frac{v_i}{v_j}} \tag{3}$$

METHOD 2. *Most of a fundamental function are under other fundamental function (Case2)*

This method adjusts the f_i as Figure 3 showing by modifying d_i. For instance, we assume that two different sampling points $\vec{s_i}, \vec{s_j}(i \neq j, d_i > d_j)$ exist. If $f_i(\vec{c}) > f_j(\vec{c})$, then this method modifies d_i to satisfy $f_i(\vec{c}) = f_j(\vec{c})$ by the expression(4).

$$d_i = \frac{(\vec{c} - \vec{s_i})^2}{\frac{k^2}{2} - \ln \frac{v_j}{v_i}} \tag{4}$$

$\vec{c} = \vec{s_j} + k\sqrt{\frac{d_j}{2}}\vec{u}$, $\vec{u} = \frac{1}{||\vec{s_j} - \vec{s_j}||}(\vec{s_j} - \vec{s_j})$. $\vec{u}$ is the unit vector whose direction is from $\vec{s_i}$ to $\vec{s_j}$. $\vec{c}$ is a control point of adjusting. $\vec{c}$ depends on the parameter $k (\in \mathbb{R}^*_+)$. As k grows, this method is performed at the point which distance from s_j is large. For example, $\vec{c}$ goes right as k grows in Figure 3.

The simple way of estimating utility functions is to connect all sampling points smoothly as Figure 4 showing. However, the utility is usually estimated as higher value than real one when the distance between some sampling points is large. Our method improves this issue by using a maximum of fundamental functions as Figure 4 showing.

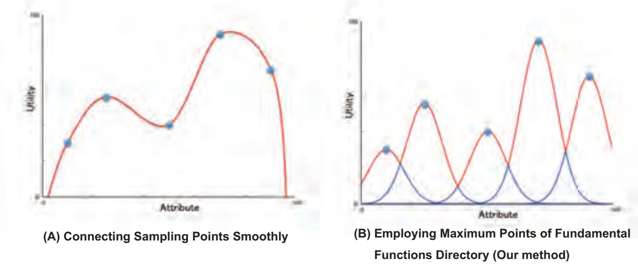


Figure 4: A Estimated method of Connecting Sampling Points Smoothly and Employing Maximum Points of Fundamental Functions Directory

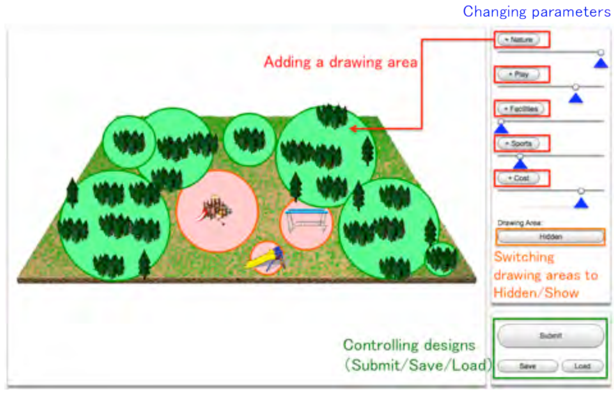


Figure 5: User Interface of Creating a Fundamental Park Design

Our method has a tendency to make agreement at the area of containing more information because the utility with enough information is large. By contract, it is difficult for agents to make agreement when the number of samples is not enough. However, our method modifies this problem because the sampling points are decided based on the users' suggestions.

3.3 Estimating the Utility Space of users in real world

In this paper, our system estimates the user's utility space as follows(`[Step1]`〜`[Step4]`).

`[Step1]` **Creating a Fundamental Design**
The manager of the negotiation sets up a negotiation. The manager creates a fundamental design by the user-interface as Figure 5 showing. The manager decides the arrangements of trees, playground equipments, facilities and so on. The manager can check the park designs generated automatically by our system, and change some parameters for reflecting his ideas.

`[Step2]` **Deciding Sampling Points**
Our system decides some sampling points in the attribute space. In this paper, the sampling points are selected randomly.

`[Step3]` **Evaluation by the users**
Our system generates the park designs at the sampling points. There are some appraising methods for evaluating the sam-

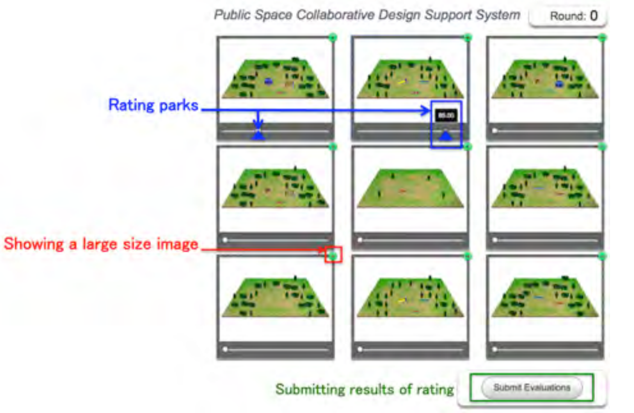


Figure 6: User Interface of Evaluating Park Designs

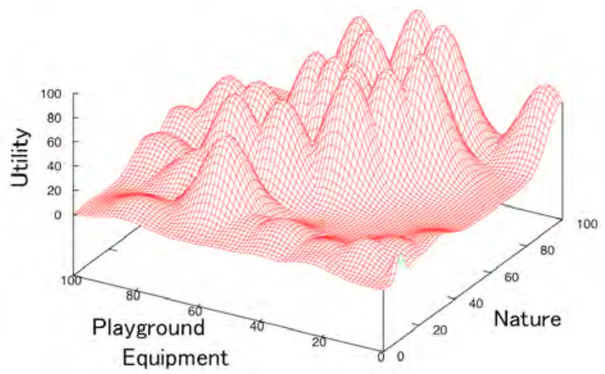


Figure 7: Estimated Utility Function

pling points (e.g. voting, rating). In this paper, we employ the rating method. Users rate each park design and submit the results of rating by the user-interface as Figure 6 showing.

`[Step4]`**Estimating Utility Functions**
First, the system generates the fundamental functions. Next, our system combines all of the fundamental functions by Method1 and Method2). Specifically, these methods adjust $d_i, d_j (0 \leq j < i)$. d_i is initialized by $D_0 (d_i := D_0$, D_0 is the initial value of the distribution of fundamental function).

`[Step2]`$\sim$`[Step4]` is repeated during the period of negotiation decided by the manager in the [Step1].

4. AN EXAMPLE OF ESTIMATING UTILITY FUNCTION

In this section, we demonstrate some results of our systems. The purpose of the demonstration is to evaluate our method and to show the characters of our method.

In this demonstration, we assume that the user has a following idea: "The parks which have many trees and some playground equipments are good. The parks which have too many or few playground equipments are not so good." Figure 7 shows an example of the estimated utility function. In this demonstration, the number of sampling is 30 and the number of attributes is 2. The reason of small number of attributes is that we can't show graphically when the number of attributes is more than 3. As you know, our method can be applied when the number of attributes is more than 3.

The axis "Nature" in Figure 7 shows how rich the nature of the park is. The large value of "Nature" means that the park has rich nature. The axis "Playground Equipment" shows how many the playground equipments are in the park. The large value of "Playground Equipment" means that the park has many playground equipments.

In Figure 7, the utility is the highest when "Playground Equipment" is 50 and "Nature" is the high value. Therefore, the estimated utility function represents the accurate preferences of the user. However, the utility is too high when "Nature" is 60 and "Playground Equipment" is 50. This is because that too many samplings are happened at the point. The efficient sampling for estimation of utility spaces is one of the future work.

	U1	U2	Est1	Est2	Err1	Err2
User1	80	85	94	61	14	24
User2	80	80	-	64	-	16
User3	60	50	19	74	41	24
User4	80	85	-	-	-	-
User5	85	75	80	73	5	2
User6	90	97	62	82	28	15
User7	70	80	57	51	13	29
User8	90	86	67	58	23	28
User9	80	80	90	87	10	7
User10	90	90	86	75	4	15
User11	65	65	69	71	4	6
Average	79.09	79.36	69.30	69.60	15.78	16.60

Table 1: Utility Values for The Optimal Agreement

5. EXPERIMENTS

5.1 Setting of Experiments

We conducted an experiment to evaluate the effectiveness of our system and confidence of our preference elicitation method. In the experiment, we ran 2 negotiations. In each negotiation, a number of participants is 11. "Nature" and "Playground Equipment" are used as attributes. Each attribute is a real number which is bigger than or equal to 0 and less than or equal to 100. The period of first negotiation is 10 minutes and second negotiation is 5 minutes. Because participants understood our system, the period of second negotiation is reduced. To find the optimal agreement, we used simulated annealing (SA) and the best result of 5 SAs is adopted as the optimal agreement because SA is easy to implement and finding optimal agreement is not our main work. After the negotiations, we send out questionnaires to get users' comments.

5.2 Results of Experiments

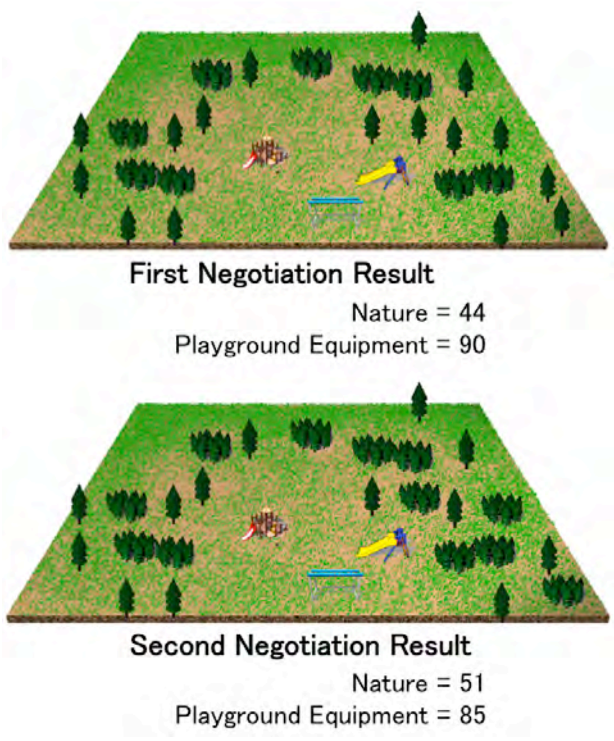


Figure 8: Negotiation Results

Figure 8 shows the results of negotiations. Table 1 shows some information about utility values for the results. U1 and U2 are user's rates (real utility values) for the results. Est1 and Est2 are user's estimated utility values for the results. Err1 and Err2 are margins of error between real utility value (U1, U2) and estimated utility value (Est1, Est2). U1, Est1 and Err1 are values for the first negotiation. U2, Est2 and Err2 are values for the second negotiation.

An average of U1 is 79.09 and U2 is 79.36. Therefore, we find many people agree to the results. An average of Err1 is 15.78 and Err2 is 16.60. These values are not so good but our method of preference elicitation can elicit tendencies of user's preference.

Table 1 shows that most of the users' utility function are accurately elicited like User5 but some users' preference elicitation are not accurately elicited like User3. Figure 9 is a elicited utility function of User5. This case is preference of a user is accurately elicited. Figure 10 is a elicited utility function of User3. This case is preference of a user is not accurately elicited. A shape which has many sharp mounds like Figure 10 occurs when some close points on a attributes space have very different utility values, in other words, some similar park designs are got very different rates. A reason of this problem is no considering changes of preferences. In fact, User3 commented "My criteria of rating are inconsistent in a process of evaluations." on a questionnaire. In real world, human preferences change as time passes. Additionally, almost all methods of preference elicitation[9, 10, 11]

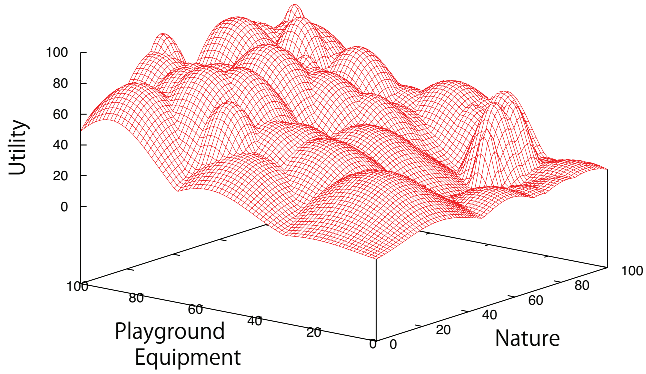


Figure 9: A Good Case of Elicitation (Elicited Utility Function of User5)

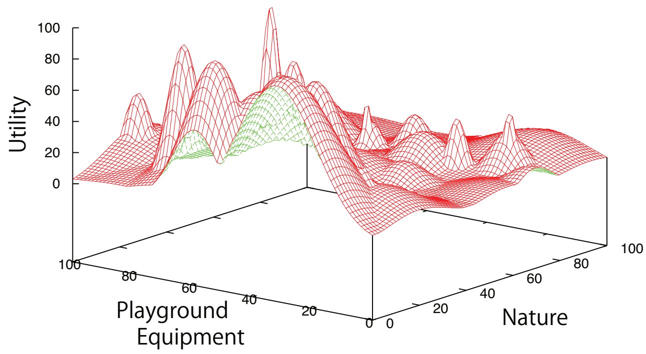


Figure 10: A Bad Case of Elicitation (Elicited Utility Function of User3)

and utility theory has this problem, too. It seems establishing "time" as a attribute resolves this problem but sampling points on a utility space from a user can not be finished. Because a number of samples getting at once is limited and samples acquired on different steps have different time as an attribute. As a result, a number of samples can not be enough to describe a utility space.

6. RELATED WORKS

Most previous works on multi-issue negotiation have addressed only linear utilities[1, 2, 3, 4]. Recently some researchers have been focusing on more complex and nonlinear utilities. For example, Ito, Fujita and Mizutani et. al[5, 6, 7, 8] proposes the automated negotiation protocol with issue-interdependency. However, most of the paper assumed the perfect utility functions of agents. In real world, it is impossible to elicit the all utility information of agents. In this paper, we propose the method of estimating the utility spaces with issue-dependences.

Luo et al.[9] proposes a method of eliciting and quantifying the trade-off between issues by the user-interactions. However, the system don't work well when the utility function is complex with the dependences between more than 3 issues. On the other hand, our system can work well when the utility functions are more complex.

In [12, 13], the system supports a yard-design based on interactive GA. This system can generate the efficient yard-designs based on the preference of users. However, this system isn't assumed the multi-party negotiations. Our system supports to the multi-party collaborative designs and consensuses among many users.

7. CONCLUSION

In this paper, we implemented a collaborative park-design support system based on the multi-agent systems. In particular, we focused on the steps for determing the attribute space and estimating the utility spaces of users in real world. Our experimental results shows our system succeeded to build a consensus many participants agreed to.

Our future works are the method of selecting sampling points for efficient estimation of utility spaces, an implementation of a feedback step and establishing a new model considerd changes of human preferences.

8. REFERENCES

[1] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. In Artificial Intelligence, pages 142:205-237, 2002.

[2] S. Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. Multi-issue negotiation with deadlines. J. Artif. Intell. Res. (JAIR), 27:381-417, 2006.

[3] S. Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. Approximate and online multi-issue negotiation. In AAMAS, page 156, 2007.

[4] S. Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In AAMAS (2), pages 1041-1048, 2009.

[5] Ito, T., Hattori, H., and Klein, M.: Multi-issue Negotiation Protocol for Agents : Exploring Nonlinear Utility Spaces, in Proc. of IJCAI-2007, pp. 1347?1352 (2007).

[6] Katsuhide Fujita, Takayuki Ito, Mark Klein: Approximately fair and secure protocols for multiple interdependent issues negotiation. AAMAS (2) 2009: 1287-1288.

[7] Katsuhide Fujita, Takayuki Ito, Mark Klein: Secure and efficient protocols for multiple interdependent issues negotiation. Journal of Intelligent and Fuzzy Systems 21(3): 175-185 (2010).

[8] Nobuyasu Mizutani, Katsuhide Fujita, Takayuki Ito: Issue Clustering and Distributed Genetic Algorithms for Multi-issue Negotiations. ACIS-ICIS 2010: 593-598.

[9] X Luo, N. R. Jennings, and N. Shadbolt. Acquiring user strategies and preferences for negotiating agents: A default then adjust method. International Journal of Human Computer Studies, 64(4):304-321, 2006.

[10] Kuriyama Koichi and Ishii Yutaka. Estimation of the Environmental Value of Recycled Wood Wastes : A Conjoint Analysis Study. Journal of forest research 5(1), 1-6, 2000-02-16.

[11] Tuomas Sandholm and Craig Boutilier. Preference Elicitation in Combinatorial Auctions. Proceedings of the Third ACM Conference on Electronic Commerce, 2001.

[12] Nanba Masayoshi, Kobe Masashi and Cai Dongsheng. Japanese Garden Design using Interactive GA. In IPSJ SIG Notes of Information Processing Society of Japan (IPSJ): 2001(96), 55-62, 2001-10-12.

[13] Nanba Masayoshi, Kobe Masashi and Cai Dongsheng. Designing Japanese Garden using Interactive GA. In IPSJ SIG Notes of Information Processing Society of Japan (IPSJ): 2003(86), 53-58, 2003-08-18.

APPENDIX

A. THE PRODUCTION OF THE EXPRESSION (3)

The expression (3) modifies d_i to satisfy $f_i(\vec{s_j}) = f_j(\vec{s_j})$.

$$\begin{aligned} f_i(\vec{s_j}) &= f_j(\vec{s_j}) \\ v_i \cdot \exp(-\frac{(\vec{s_j}-\vec{s_i})^2}{d_i}) &= v_j \\ d_i &= \frac{(\vec{s_j}-\vec{s_i})^2}{\ln \frac{v_i}{v_j}} \end{aligned}$$

d_i is produced.

To f_i becomes the gaussian, d_i must be positive.

$$\begin{aligned} d_i = \frac{(\vec{s_j}-\vec{s_i})^2}{\ln \frac{v_i}{v_j}} &> 0 \\ v_i &> v_j \end{aligned}$$

Because of $f_i(s_j) > f_j(s_j)$ which is the condition of applying the method 1,

$$v_i = f_i(s_i) > f_i(s_j) > f_j(s_j) = v_j$$

$d_i > 0$ is evidenced.

B. THE PRODUCTION OF THE EXPRESSION (4)

The expression (4) modifies d_i to satisfy $f_i(\vec{c}) = f_j(\vec{c})$.

$$\begin{aligned} f_i(\vec{c}) &= f_j(\vec{c}) \\ v_i \cdot \exp(-\frac{(\vec{c}-\vec{s_i})^2}{d_i}) &= v_j \cdot \exp(-\frac{(\vec{c}-\vec{s_j})^2}{d_j}) \end{aligned}$$

Because of $\vec{c} = \vec{s_j} + k\sqrt{\frac{d_j}{2}}\vec{u}$,

$$\exp(-\frac{(\vec{c}-\vec{s_i})^2}{d_i}) = \frac{v_j}{v_i} \cdot \exp(-\frac{k^2}{2}\vec{u}^2)$$

$\vec{u}^2 = 1$ because $\vec{u}$ is a unit vector.

$$\begin{aligned} \exp(-\frac{(\vec{c}-\vec{s_i})^2}{d_i}) &= \frac{v_j}{v_i} \cdot \exp(-\frac{k^2}{2}) \\ d_i &= \frac{(\vec{c}-\vec{s_i})^2}{\frac{k^2}{2} - \ln \frac{v_j}{v_i}} \end{aligned}$$

d_i is produced.

Learning and Evaluating Realistic Behavior in the Social Ultimatum Bargaining Game

Yu-Han Chang
Univ. of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
ychang@isi.edu

Tomer Levinboim
Univ. of Southern California
Los Angeles, CA
tomer.levinboim@gmail.com

Rajiv Maheswaran
Univ. of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
rmaheswar@isi.edu

ABSTRACT

We address the challenges of evaluating the fidelity of autonomous agents that are attempting to replicate human behaviors. This is a fundamental issue in the emerging intersection of artificial intelligence and social science motivated by problems such as training in virtual environments, human-agent negotiations, and large-scale social simulation. Our specific interest focuses on emulating human strategic behavior over time, in a repeated negotiation setting. We introduce and investigate the Social Ultimatum Game, an extension of the classical Ultimatum bargaining game, and discuss the efficacy of a set of metrics in comparing various autonomous agents to human behavior collected from experiments.

Categories and Subject Descriptors

I.1.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Algorithms, Economics, Experimentation

Keywords

Human-Agent Negotiation, Metrics, Multi-Agent Systems, Game Theory, Ultimatum Game, Mathematical Models of Human Behavior, Learning, Adaptation

1. INTRODUCTION

Much of the complexity in a multi-agent interaction lies not in the complexity of the domain being considered, but in the complexity of the decision-making processes of the other agents involved. To play optimally, a player must make the right assumptions and arrive at the correct beliefs about the opponents being faced. For example, in the classic Ultimatum bargaining game, a player acting as the proposer would choose to act quite differently depending on the player's beliefs about the kind of opponent being faced. A typical westerner, Peruvian Amazonian tribe-member, economist, or an autistic individual all behave differently [6], and the player would have to adjust his negotiating strategy accordingly. In this paper, we are interested in autonomous agents that can replicate human behavior in this type of negotiation setting, and in metrics that are suitable for evaluating the similarity between the agent behavior and actual human behavior.

In traditional AI, the classical Turing Test relies on human evaluation to judge the verisimilitude of the conversation produced by the autonomous agent. In more restricted problems, such as classification, we are satisfied when a machine consistently produces the correct label (a perfect match), given a test data point. In this paper, we are concerned with domains falling somewhere in the middle, where an agent's human-like behavior will not necessarily produce a perfect match to some predefined standards, but where we would prefer not to rely exclusively on human judgement to determine whether an agent's outputs are "close" to real human behavior.

In particular, we are interested in multi-agent domains where humans make sequential decisions over time, such as in a multi-round negotiation. Building a realistic autonomous agent in this type of domain has practical applications in many other areas, for example training in virtual environments [13], large-scale social simulation [3], and adversarial modeling [1]. In the emotional agents community, the degree of realism is typically evaluated by a human judge [9]. In the machine learning and reinforcement learning community, agent "goodness" is typically evaluated relative to optimal behavior, using a metric like expected reward. However, realistic human behavior is often not optimal, and in many of the domains of interest, the notion of optimality is ill-defined.

Optimality of one agent in a multi-agent domain is dependent on the other agents. If a machine's assumptions about the other agents is incorrect, then its behavior, even if optimal given those assumptions, could be wildly different from normal human behavior. We will see an example of this shortly, in a variant of the classic Ultimatum game. Since the validity of these assumptions is an essential part of what must be evaluated, optimality based on the assumptions is not a good metric for realism. We need a different approach.

Human data in multi-agent domains is getting easier to collect, given the current state of access to the Internet and online interaction. Thus, we can obtain baseline collections of behavior trajectories that describe human play. The challenge is to find a way to compare collections of traces produced by autonomous agents with this existing baseline, in order to determine which agents exhibit the most realistic behavior.

In this paper, we investigate these issues in the context of the Social Ultimatum Game (SUG). SUG is a multi-agent multi-round extension of the Ultimatum Game [6], which has been a frequently studied game over the last three decades as a prominent example of how human behavior deviates from game-theoretic predictions that use the "rational actor" model. Data gathered from people playing SUG was used to create various classes of autonomous agents that modeled the behaviors of the individual human players. We then created traces from games with autonomous agents emulating the games that the humans played. We develop several metrics to compare the collections of traces gathered from games played by humans and games played by the autonomous agents. From this analysis, it becomes clear that human behavior contains unique temporal patterns that are not captured by the simpler metrics. In SUG, this is revealed in the likelihood of reciprocity as a function of the history of reciprocity. The key implication is that it is critical to retain the temporal elements when developing metrics to evaluate the efficacy of autonomous agents for replicating human strategic behavior in dynamic settings.

2. THE SOCIAL ULTIMATUM GAME

To ground our subsequent discussion, we begin by introducing the Social Ultimatum Game. The classical Ultimatum Game, is a two-player game where P_1 proposes a split of an endowment $e \in \mathbb{N}$ to P_2 who would receive $q \in \{0, \delta, 2\delta, \ldots, e-\delta, e\}$ for $\delta \in \mathbb{N}$. If P_2 accepts, P_2 receives q and P_1 receives $e - q$. If P_2 rejects, neither player receives anything. The subgame-perfect Nash or Stackelberg equilibrium has P_1 offering $q = \delta$ (i.e., the minimum possible offer), and P_2 accepting, because a "rational" P_2 should accept any $q > 0$, and P_1 knows this. Yet, humans make offers that exceed δ, make "fair" offers of $e/2$, and reject offers greater than the minimum.

To represent the characteristics that people operate in societies of multiple agents and repeated interactions, we introduce the Social Ultimatum Game. The players, denoted $\{P_1, P_2, \ldots, P_N\}$, play $K \geq 2$ rounds, where $N \geq 3$. The requirement of having at least three players in necessary to give each player a choice of whom to interact with. In each round k, every player P_m chooses a recipient R_m^k and makes them an offer $q_{m,n}^k$ (where $n = R_m^k$). Each recipient P_n then considers the offers they received and makes a decision $d_{m,n}^k \in \{0,1\}$ for each offer $q_{m,n}^k$ to accept (1) or reject (0) it. If the offer is accepted by P_m, P_m receives $e - q_{m,n}^k$ and P_n receives $q_{m,n}^k$, where e is the endowment to be shared. If an offer is rejected by P_n, then both players receive nothing for that particular offer in round k. Thus, P_m's reward in round k is the sum of the offers they accept (if any are made to them) and their portion of the proposal they make, if accepted:

$$r_m^k = (e - q_{m,n}^k) d_{m,n}^k + \sum_{j=1 \ldots N, j \neq m} q_{j,m}^k d_{j,m}^k \tag{1}$$

The total rewards for P_m over the game is the sum of per-round winnings, $r_m = \sum_{k=1}^{K} r_m^k$. A game trajectory for P_m is a time-series of proposed offers, $O_m^k = (R_m^k, q_{m,n}^k, d_{m,n}^k)$ and received offers, $O_{n,m}^k = (R_n^k, q_{n,m}^k, d_{n,m}^k)$. At time k, the trajectory for P_m is its history of offers made and received: $T_m^k = (O_m^k, \{O_{n,m}^k\}_n, O_m^{k-1}, \{O_{n,m}^{k-1}\}_n, \ldots, O_m^1, \{O_{n,m}^1\}_n)$. Assuming no public information about other players' trajectories, T_m^k includes all the observable state information available to P_m at the end of round k.

3. METRICS

Let C_m be the collection of trajectories P_m produces by taking part in a set of Social Ultimatum Games. In other domains, these traces could represent other interactions. Our goal is to evaluate the resemblance of a set of human trace data C to other sets of traces $\widetilde{C}$, namely those of autonomous agents. We need a metric that compares sets of multi-dimensional time series: $d(C, \widetilde{C})$. Standard time-series metrics such as Euclidean or absolute distance, edit distance, and dynamic time warping [11] are not appropriate in this type of domain.

One challenge arises because we are interested in the underlying behavior that creates the trajectories rather than superficial differences in the trajectories themselves. If we can collapse a collection of traces C to a single probability distribution Q, by aggregating over time, then we can define a *time-collapsed* metric,

$$d(C, \widetilde{C}) = KL(Q||\widetilde{Q}) + KL(\widetilde{Q}||Q) \tag{2}$$

where KL denotes the Kullback-Leibler divergence. The sum enforces symmetry and nonnegativity. Time-collapsed metrics for SUG include:

- **Offer Distribution.** Let Q^O be the distribution of offer values $\{q_{m,n}^k\}$ observed over all traces and all players.

- **Target-Recipient Distribution.** Let Q^R denote the likelihood that a player will make an offer to the k^{th} most likely recipient of an offer. This likelihood is non-increasing in k. In a 5-person game, a single player may have an target-recipient distribution that looks like $\{0.7, 0.1, 0.1, 0.1\}$ which indicates that they made offers to their most-targeted partner 7 times more often than their second-highest-targeted partner. We can produce Q^R by averaging over all games to characterize a player and further average over all players to characterize a population.

- **Rejection Probabilities.** For each offer value q, we have a Bernoulli distribution Q^{B_q} that captures the likelihood of rejection by averaging across all players, games and rounds in a collection of traces. We then define a metric:

$$d^B(C, \widetilde{C}) = \sum_{q=0}^{10} KL(Q^{B_q}||\widetilde{Q}^{B_q}) + KL(\widetilde{Q}^{B_q}||Q^{B_q}).$$

We can also define *time-dependent* metrics that acknowledge that actions can depend on observations of previous time periods. One prominent human manifestation of this characteristic is reciprocity. We define two time-dependent metrics based on reciprocity:

- **Immediate Reciprocity** When a player receives an acceptable offer from someone, they may be more inclined to reciprocate and propose an offer in return in the next round. We can quantify this $p(R_m^{k+1} = n | R_n^k = m)$ across all players and games in a collection of traces. This probability defines a Bernoulli distribution Q^Y from which we can define a metric d^Y as before.

- **Reciprocity Chains** Taking the idea of reciprocity over time further, we can calculate the probability that an offer will be reciprocated, given that a chain of reciprocity has already occurred. For example, for chains of length $c = 2$, we $p(R_m^{k+1} = n | R_n^k = m, R_m^{k-1} = n)$; for $c = 3$, we calculate $p(R_m^{k+1} = n | R_n^k = m, R_m^{k-1} = n, R_n^{k-2} = m)$. As before, these probabilities can be used to define a Bernoulli distribution Q^{Y_c} for each length c. Then, for some L, we define

$$d_L^Y(C, \widetilde{C}) = \sum_{c=1}^{L} KL(Q^{Y_c} || \widetilde{Q}^{Y_c}) + KL(\widetilde{Q}^{Y_c} || Q^{Y_c}).$$

We expect that the longer a pair of players reciprocate, the higher the likelihood that they will continue doing so. The probabilities of how likely humans are to reciprocate can be obtained from the experimental data.

4. AUTONOMOUS AGENTS

In this section, we describe various agent models of behavior. We first apply traditional game-theoretic analysis to the Social Ultimatum Game to derive the "optimal" behavior under rational actor assumptions. We then describe two distribution-based agents that do not model other agents but are capable of incorporating human behavior data. Finally, we describe an adaptive agent that incorporates some aspects of human behavior such as fairness and reciprocity.

4.1 Game-Theoretic Agents

Let strategies be characterized by the statistics that they produce in steady-state: the distribution of offers made by each player, where $p_m^g(n, q)$ denotes the likelihood that P_m will *give* an offer of q to P_n, and the distribution of offers accepted by each player, where $p_m^a(n, q)$ denotes the likelihood that P_m will *accept* an offer of q from P_n. Then, the expected reward for P_m per round in steady-state is $r_m =$

$$\sum_{n,q} q p_n^g(m, q) p_m^a(n, q) + \sum_{n,q} (e - q) p_m^g(n, q) p_n^a(m, q) \qquad (3)$$

where $\sum_{n,q} p_m^g(n, q) = 1,\ \forall m$, as the total outgoing offers must total one offer per round, and the acceptance likelihoods are $p_m^a(n, q) \in [0, 1],\ \forall m, n, q$. A player maximizing these rewards will modify their offer likelihoods $\{p_m^g(n, q)\}$ and acceptance likelihoods $\{p_m^a(n, q)\}$, given those of other players. A player can create the desired statistics by playing a stationary mixed strategy with the desired likelihoods. To optimize the offer likelihoods, P_m sets

$$p_m^g(n, q) > 0, \forall n \in \mathcal{N}^g \subset \arg\max_n \max_q (e - q) p_n^a(m, q)$$

such that $\sum_{n,q} p_m^g(n, q) = 1$, and $p_m^g(n, q) = 0$, otherwise. Thus, in equilibrium, P_m will make offers to those agents whose acceptance likelihoods yield the highest expected payoff.

Proposition. In the Social Ultimatum Game, accepting all offers is not a dominant strategy.

We first note that players make offers to the players and of the values that maximize their expected rewards. Thus, for P_n to receive an offer from P_m, it must be the case that $(e - q) p_m^a(n, q)$ is maximized for P_m over n and q, given $P_n's$ choice of $p_m^a(n, q)$. Let us now assume that

$$p_m^a(n, q) = 1\ \forall q \geq \underline{q}, m, n \qquad (4)$$

$$p_m^a(n, q) = 0, \forall q < \underline{q}, m, n \qquad (5)$$

for all m, n and $\underline{q} > \delta$. This says that all players accept offers above some minimum threshold that is greater than the minimum offer and never accept offers below that threshold. Let us further assume the case that all offers are made uniformly among players. Under these conditions, each player gains $\underline{q}$ per round in rewards from incoming offers. If P_m was to switch to the strategy of accepting all offers of value δ, then all players would see an expected value of $(e - \delta)$ of making all offers to P_m which would result in P_m gaining $(N - 1)\delta$ in rewards per round. We note that it is not necessarily the case that $(N - 1)\delta \geq \underline{q}$, thus the "greedy" strategy is not dominant in the Social Ultimatum Game. ■

Consider the case where all players accept only $(e - \delta)$ or above in a game where $e = 10$ and $N = 5$. Switching to the "greedy" strategy would reduce gains from receiving offers from 9 per round to 4 per round. This rationalizes the idea that getting fewer high value offers can be more valuable than a lot of low offers.

Proposition. In the Social Ultimatum Game, Nash equilibrium outcomes only happen when players employ strategies of the form "greedy' strategies, where

$$p_m^g(n, q) = 0,\ \forall q > \delta, m, n, \quad p_m^a(n, \delta) = 1, \forall m, n, \qquad (6)$$

i.e.,"greedy" strategies where players only make the minimum offers of δ, and all players accept all minimum offers.

Given the characterizations above, if P_m was to switch to the strategy where

$$p_m^a(n, \underline{q} - 1) = 1,\ \forall n, \qquad (7)$$

then all players would make all offers to P_m who would gain $(N-1)(\underline{q}-1)$ per round incoming offers which is greater than $\underline{q}$, for $N \geq 3$. Thus, any strategy that can be "undercut" in this manner cannot yield a Nash equililibrium outcome. We note that if we relax the assumption that offers are made uniformly among players that maximize expected reward from outgoing offers, then there will exist some player who will be making at most $\underline{q}$ per round, and that player will still have incentive to "undercut". By a similar argument, if all players are accepting a particular value of q, then the likelihood of accepting that offer will gravitate to 1. Thus, all players, will be driven down to accepting all offers $q = \delta$. Given, this players will only make offers for $q = \delta$, and thus, the "greedy" strategy is the only Nash equilibrium. ■

It is interesting that this outcome, while similar to the Ultimatum Game, is not due to the first player leveraging their position as the offerer and being "greedy", but instead from the "rational" players competing to maximize gains from received offers.

4.2 Distribution-Based Agents

One way to create agents that satisfy a set of metrics is to use the metrics to generate the agent behavior. Using only time-collapsed metrics, one could create a distribution-based agent (DBA) as follows. Learn distributions of offer value, target recipient and rejection percentage from human data. Find the appropriate target-recipient distribution based on number of participants and assign agents to each position (i.e., most likely to least likely). In offer phases of each round, choose a target by sampling from the target-recipient distribution and an offer value by sampling from the offer distribution. For received offers, decide via Bernoulli trial based on the rejection percentage for that offer value.

The DBA has no notion of reciprocity. We also investigated a class of distribution-based reciprocal agents (DBRA) which behave like the DBA agents in all aspects other than target selection. If DBRA agents receive an offer it will decide to reciprocate based on a reciprocation percentage that is learned from human data. If multiple offers are received, the target is chosen using a relative likelihood based on the target-recipient distribution. Similarly, if it doesn't receive any offers, it uses the target-recipient distribution. While the distribution-based agents act on the basis of data of human play, they do not have models of other agents and consequently execute an open-loop static policy. The following model introduces an adaptive model that is not based simply on fitting the metrics.

4.3 Adaptive Agents

In order to create adaptive agent models of human players for the Social Ultimatum Game, we need to incorporate some axioms of human behavior that may be considered "irrational". The desiderata that we address include assumptions that people will

1. start with some notion of a fair offer,
2. adapt these notions over time at various rates based upon their interactions,
3. have models of other agents, and
4. choose the best option while occasionally exploring for better deals.

Each player P_m is characterized by three parameters: $\alpha_m^0 : P_m$'s initial acceptance threshold, $\beta_m : P_m$'s reactivity and $\gamma_m : P_m$'s exploration likelihood

The value of $\alpha_m^0 \in [0, e]$ is P_m's initial notion of what constitutes a "fair" offer and is used to determine whether an offer to P_m, i.e., $q_{n,m}^k$, is accepted or rejected. The value of $\beta_m \in [0, 1]$ determines how quickly the player will adapt to information during the game, where zero indicates a player who will not change anything from their initial beliefs and one indicates a player who will solely use the last data point. The value of $\gamma_m \in [0, 1]$ indicates how much a player will deviate from their "best" play in order to discover new opportunities where zero indicates a player who never deviates and one indicates a player who always does.

Each player P_m keeps a model of other players in order to determine which player to make an offer to, and how much that offer should be. The model is composed as follows: $a_{m,n}^k$: P_m's estimate of P_n's acceptance threshold; $\bar{a}_{m,n}^k$: Upper bound on $a_{m,n}^k$; and $\underline{a}_{m,n}^k$: Lower bound on $a_{m,n}^k$. Thus, P_m has a collection of models for all other players $\{[\underline{a}_{m,n}^k a_{m,n}^k \bar{a}_{m,n}^k]\}_n$ for each round k. The value $a_{m,n}$ is the P_m's estimate about the value of P_n's acceptance threshold, while $\underline{a}_{m,n}^k$ and $\bar{a}_{m,n}^k$ represent the interval of uncertainty over which the estimate could exist. Each player P_m initializes these values as follows:

- $a_{m,n}^0 = \alpha_m$
- $\bar{a}_{m,n}^k = \lceil e/2 \rceil$
- $\underline{a}_{m,n}^k = 0$

This denotes that each player begins with the assumptions that other players in the game (1) have acceptance thresholds that are the same as theirs, (2) will always accept an equal split of the endowment, and (3) may be willing to accept an arbitrarily low offer.

During the course of the game, each player will engage in a variety of actions and updates to their models of agents. Below, we present our model of how our adaptive agents address those actions and model updates. For simplicity, we will assume that $\delta = 1$.

4.3.1 Making Offers

In each round k, P_m may choose to make the best known offer, denoted $\tilde{q}_m^k$, or explore to find someone that may accept a lower offer. If there are no gains to be made from exploring, i.e., the best offer is the minimum offer ($\tilde{q}_m^k = \delta = 1$), a player will not explore. However, if there are gains to be made from exploring, with probability γ_m, P_m chooses a target P_n at random and offers them $q_{m,n}^k = \tilde{q}_m^k - 1$. With probability $1 - \gamma_m$, P_m will choose to exploit. The target is chosen from the players who have the lowest value for offers they would accept, and the offer is that value:

$$q_{m,n}^k = \lceil a_{m,n}^k - \epsilon \rceil \text{ where } n \in \arg\min_{\tilde{n} \neq m} \lceil a_{m,\tilde{n}}^k \rceil \tag{8}$$

The previous equation characterizes an equivalence class of players from which P_m can choose a target agent. The ϵ parameter is used to counter boundary effects in the threshold update, discussed below. The target agent from the equivalence class is chosen using *proportional reciprocity*, by assigning likelihoods to each agent with respect to offers made in some history window.

4.3.2 Accepting Offers

For each offer $q_{m,n}^k$, the receiving player P_n has to make a decision $d_{m,n}^k \in \{0, 1\}$ to accept or reject it, based on its threshold:

$$\text{If } q_{m,n}^k \geq \lceil \alpha_m^k - \epsilon \rceil, \text{ then } d_{m,n}^k = 1, \text{ else } d_{m,n}^k = 0 \quad (9)$$

4.3.3 Updating Acceptance Threshold

The acceptance threshold is a characterization of what the agent considers a "fair" offer. Once an agent is embedded within a community of players, the agent may change what they consider a "fair" offer based on the received offers. We model this adaption using a convex combination of the current threshold and the offers that are received, with adaptation parameter β_m. Let the set of offers that are received be defined as: $R_m^k = \{q_{i,j}^k : j = m, q_{i,j}^k > 0\}$. If $|R_m^k| \geq 1$, then $\alpha_m^{k+1} =$

$$(1-\beta_m)^{|R_m^k|}\alpha_m^k + \frac{(1-((1-\beta_m)^{|R_m^k|})}{|R_m^k|}\sum_i q_{i,m}^k \quad (10)$$

If $|R_m^k| = 0$, then $\alpha_m^{k+1} = \alpha_m^k$. Thus, offers higher than your expectation will raise your expectation and offers lower than your expectation will lower your expectation at some rate.

4.3.4 Updating Threshold Estimate Bounds

As a player makes an offer $q_{m,n}^k$ and receives feedback on the offer $d_{m,n}^k$, they learn about P_n's acceptance threshold. Using this information, we can update our bounds for our estimates of their threshold, with the following rules.

If you make an offer and it is rejected, then the lower bound for the acceptance threshold for that player must be at least the offer that was rejected:

$$q_{m,n}^k > 0, d_{m,n}^k = 0 \;\Rightarrow\; \underline{a}_{m,n}^{k+1} = \max\{q_{m,n}^k, \underline{a}_{m,n}^k\} \quad (11)$$

If you make an offer and it is accepted, then the upper bound for the acceptance threshold for that player must be at most the offer that was rejected:

$$q_{m,n}^k > 0, d_{m,n}^k = 1 \;\Rightarrow\; \bar{a}_{m,n}^{k+1} = \min\{q_{m,n}^k, \bar{a}_{m,n}^k\} \quad (12)$$

The next two conditions occur because acceptance thresholds are dynamic and the bounds for estimates on thresholds for other players may become inaccurate and may need to be reset. If you make an offer, it is rejected and that offer at least your current upper bound, then increase the upper bound to the "fair" offer that you expect that the other player will accept:

$$q_{m,n}^k > 0, d_{m,n}^k = 0, q_{m,n}^k \geq \bar{a}_{m,n}^k \;\Rightarrow\; \bar{a}_{m,n}^{k+1} = \lceil e/2 \rceil \quad (13)$$

If you make an offer, it is accepted and that offer is lower than your current lower bound, then decrease the lower bound to zero:

$$q_{m,n}^k > 0, d_{m,n}^k = 1, q_{m,n}^k \leq \underline{a}_{m,n}^{k+1} \;\Rightarrow\; \underline{a}_{m,n}^{k+1} = 0 \quad (14)$$

4.3.5 Updating Threshold Estimates

Once the threshold bounds are updated, we can modify our estimates of the thresholds as follows. If the player accepts the offer, we move the estimate of their threshold closer to the lower bound and if the player rejects the offer, we move our estimate of their threshold closer to the upper bound using a convex combination of the current value and the appropriate bound as follows.

$$\begin{aligned} &d_{m,n}^k = 1 \;\Rightarrow \\ &\quad a_{m,n}^{k+1} = \min\{\beta_m\ \underline{a}_{m,n}^{k+1} + (1-\beta_m)a_{m,n}^k, \bar{a}_{m,n}^{k+1}\} \quad (15) \\ &d_{m,n}^k = 0 \;\Rightarrow \\ &\quad a_{m,n}^{k+1} = \max\{\beta_m\ \bar{a}_{m,n}^{k+1} + (1-\beta_m)a_{m,n}^k, \underline{a}_{m,n}^{k+1} + 2\epsilon\} \quad (16) \end{aligned}$$

The *min* and *max* operators ensure that we don't make unintuitive offers (such as repeating a just rejected offer), if our adaptation rate is not sufficiently high. The adaptive agent described above fulfills the properties of the desiderata prescribed to generate behavior that is more aligned with our expectations in reality.

5. EXPERIMENTS

Thus far we have introduced several autonomous agent models, and metrics to evaluate their verisimilitude to actual human behavior. In this section, we first discuss the collection of human data, and the use of this data to fit the described agent models. We then evaluate the agent performance using our proposed metrics.

5.1 Human Play

Data was collected from human subjects recruited from undergraduates and staff at the University of Southern California. In each round, every player is given the opportunity to propose a \$10 split with another player of their choosing. Games ranged from 20 to 50 rounds. A conversion rate of 10 ultimatum dollars to 25 U.S. cents was used to pay participants, i.e., \$5 per 20 rounds per player in an egalitarian social-welfare maximizing game, leading to total U.S. denominated splitting opportunities of \$5 per player per game. Each game lasted approximately 20 minutes, once regulations and training were completed. The subjects participated in organized game sessions and a typical subject played three to five games in one session. Between three and seven players participated in each game. During each session, the players interacted with each other exclusively through the game's interface on provided iPads, shown in Figure 1. No talking or visual communication was allowed. The rules of the game were as outlined in Section 2.

As shown in Figure 1, players were also randomly assigned an avatar from one of two "cultures": monks or warriors. Monk avatars tend to look similar, while warriors have more individualistic appearances. Names for each cultures also follow a naming convention. We were interested whether such small cultural cues would have any noticeable effect on game behavior – thus far this does not appear to be the case. If anything, there is a slight tendency for all players, regardless of culture, to make offers to warriors, perhaps because their avatars are more eye-catching and memorable.

The collected data includes every GUI command input by each player, with corresponding timestamp. For example, this includes not only the offers made and accepted, but also provides information about length of time a player deliberated about an offer, and occasions where a player may have changed his mind about the recipient of an offer or the amount of an offer.

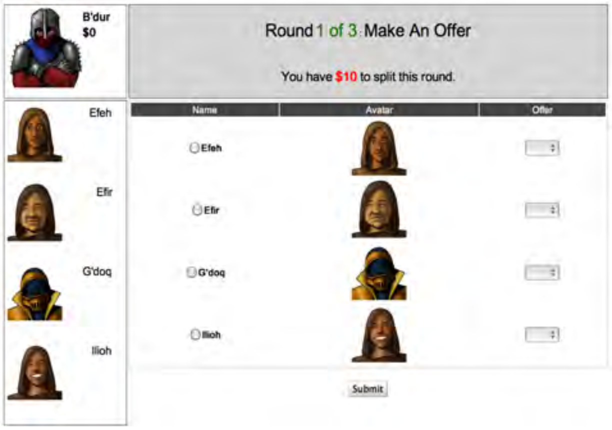


Figure 1: The Social Ultimatum Game Interface

After each game, a written survey was completed by each participant. They were asked to provide answers regarding their own game play strategies during the game, the observed strategies of the other players, and any additional comments. We have collected data from 27 human subject games thus far. In this paper, we focus on the seven 5-person games in the dataset. By restricting our attention to five-player games, we avoid biases that may be introduced if we attempted to normalize the data from the other games to reflect a five-person composition. Analysis on the games of other sizes yields similar results.

5.2 Autonomous Agents

To create the Distribution-Based Agent and Distribution-Based Reciprocal Agent using the collected data, we calculated the appropriate distributions (offer value, rejection percentage by value, targeted-recipient) by counting and averaging over all games and all players. The agents then selected target-recipients and offers based on these distributions, and made their acceptance decisions based on the rejection-by-value distributions.

For the Adaptive Agents, we analyzed the traces of each game, and estimated game-specific α, β, γ parameters of each of the participating players, as follows. For each player P_m in the game,

- α_m : This is set as the player's first offer in this game.
- β_m : When a player decreases his offer to a specific player from q_1 to q_2 after K steps (not necessarily consecutive), we find and store the best β value such that K applications of $\beta q_2 + (1-\beta)q_1$ yields a result less than $\frac{q_1+q_2}{2}$ (so that the next offer should be closer to q_2 then it is to q_1). We then take β_m to be this stored β value.
- γ_m : This is the likelihood that a player's offer is less than the minimum known accepted offer, where the minimum accepted offer at a given round k is the minimum offer known to be accepted by any player at time $k-1$.

Having estimated the population parameters of each game, we then use them as input to create an autonomous agent for each player, and simulate each game ten times to produce ten traces. Within each of these games, each of the five players uses the parameters corresponding to one of the five original human players.

6. EVALUATION

These experiments and simulations result in a collection of game traces for each of the five types of agent discussed: Human, Adaptive, DBA, DBRA, and Game-theoretic (GT). Table 1 shows the similarity between the collection of human traces and each of the four collections of autonomous agent traces, according to the metrics discussed earlier.

	Adaptive	DBA	DBRA	GT
d^O	0.57	0.008	0.008	33.26
d^R	0.21	0.0005	0.01	0.19
d^B	11.74	0.008	0.11	32.83
d^{Y_8}	4.22	16.34	20.10	97.02

Table 1: Similarity to human play, based on various metrics.

The DBA and DBRA agents score very well on the three metrics based on offer value, rejection percentage, and target-recipient. We fully expect this result as both these agents generate their behavior by sampling from these distributions. It is also clear that the GT agent performs very differently from the human data, based on most of the metrics. It is only close to the human trace data when compared on d^R, the metric based on the target-recipients distribution. This is because we assumed that the game-theoretic agent would distribute its offers uniformly across the other players, and human play roughly approximates this phenomenon. It is worth noting that the Adaptive Agent scores approximately the same as the GT agent on this metric. Naturally, the Adaptive Agent scores worse than the distribution-based agents on the temporally-independent distribution metrics d^O, d^R, and d^B, but its behavior is still relatively close to human behavior. On the temporally-dependent reciprocation-chain metric d^{Y_8}, the Adaptive Agent scores much better in similarity to the human traces.

To get a more intuitive sense of the differences in the trace data, we also display the actual distributions that underlie the metrics in Figures 2-5, which shows the distributions of offer amounts for each of the agent types, the probability of rejection given each offer amount, the distribution of offer recipients, ordered from most likely to least likely, and the probability that an offer will be reciprocated, given that a chain of c offers have been made between the players in the past $c = 1, 2, \ldots, 8$ time periods.

While the Adaptive Agent may not have been the most human-like agent according to the other three metrics, the form of its distributions still reasonably resembled the distributions produced by human play. However, on the time-dependent reciprocation-based metric, it is very clear that the Adaptive Agent is the only one that exhibits behavior that is similar to human play. This temporal dependence is crucial to creating agent behavior that emulates human behavior.

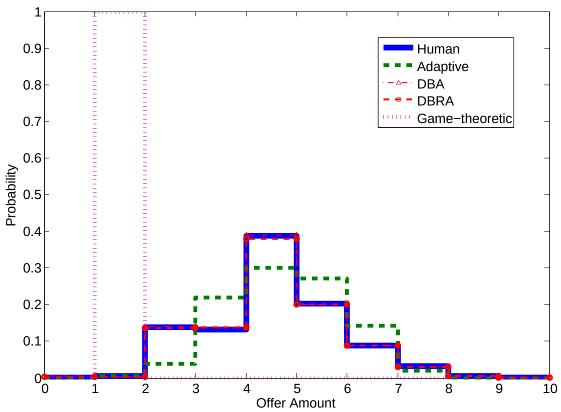


Figure 2: (Top to bottom) Distribution of offer amounts, for each of the five types of agents discussed. d^O is based on these distributions.

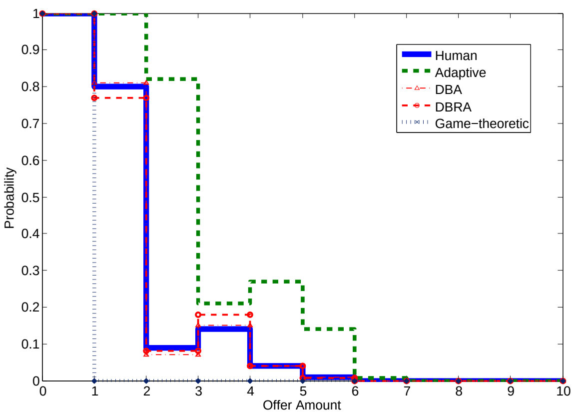


Figure 3: Rejection probabilities given offer amounts, for each of the five types of agents discussed. In our game-theoretic agent, we assumed that offers of $0 would be rejected. d^B is based on these probabilities.

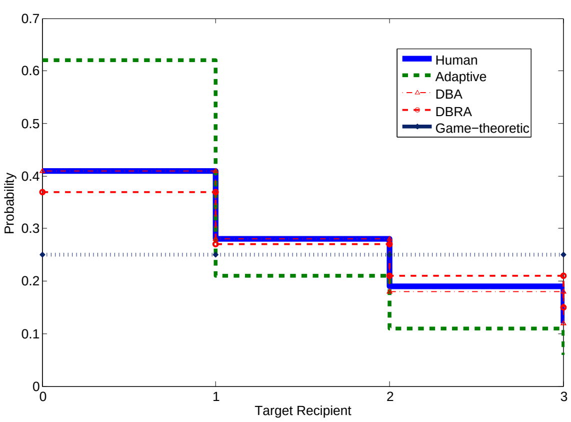


Figure 4: Target recipient distribution, for each of the five types of agents discussed. The game-theoretic agent was assumed to distribute its offers uniformly across the other agents. d^R is based on these distributions.

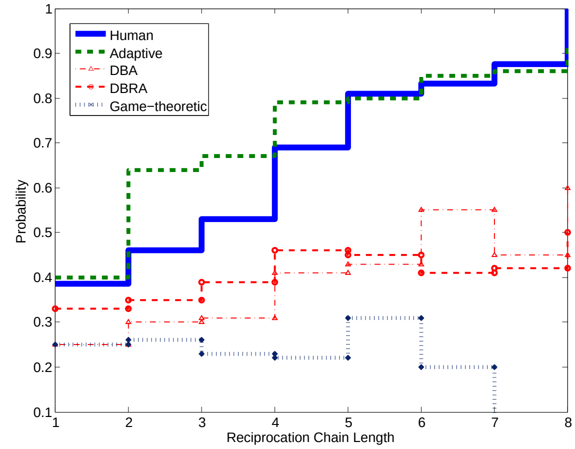


Figure 5: Graph showing the probability that an offer is reciprocated, given that a chain of reciprocation of length $c = 1, 2, \ldots, 8$ has just occurred. d_8^Y is based on these probabilities.

7. RELATED WORK

Our choice to investigate the Ultimatum Game was motivated by its long history in the field and the fact that it is a leading example of where game-theoretic reasoning fails to predict consistent human behaviors [5, 12, 6]. Economists and sociologists have proposed many variants and contexts of the Ultimatum Game that seek to address the divergence between the "rational" Nash equilibrium strategy and observed human behavior, for example, examining the game when played in different cultures, with members of different communities, where individuals are replaced by groups, where the players are autistic, and when one of the players is a computer. Interestingly, isolated non-industrialized cultures, people who have studied economics, groups, being autistic, and playing with a computer all tend to lead to less cooperative behavior [5, 12, 10, 7, 2, 4]. Learning human game data is a promising approach for quickly learning realistic models of behavior. In the paper, we have demonstrated this approach in SUG, and proposed metrics that evaluate the similarity between autonomous agents' game traces and human game traces.

Recently, there has also been other work attempting to model human behavior in multi-agent scenarios, primarily in social network and other domains modeled by graphical relationship structures [8]. In contrast, our work focuses on multi-agent situations where motivated agents make sequential decisions, thus requiring models that include some consideration of utilities and their interplay with psychological effects. Our Adaptive Agent is a simple model, with parameters that are fit to the collected data, that demonstrates this approach.

Finally, a critical aspect of this line of work must include the development of appropriate metrics for evaluating the verisimilitude of the autonomous agent behaviors to human behavior. While there is a long literature on time-series metrics [11], in this paper, we show that these metrics do not capture the temporal causality patterns that are key to evaluating human behaviors, and thus are insufficient to evaluate agent behaviors when used alone.

8. CONCLUSION

Our goal is to develop approaches to create autonomous agents that replicate human behavior in multi-agent domains where humans make sequential decisions over time. To create and evaluate these agents, one needs appropriate metrics to characterize the deviations from the source behavior. The challenge is that a single source behavior in dynamic environments produces not a single decision but instead multiple traces where each trace is a sequence of decisions. A single source can produce a diverse collection of traces. Thus, the challenge is to find a way to compare collections of traces.

We introduced the Social Ultimatum Game and in that context, developed time-collapsed and time-dependent metrics to evaluate a collection of autonomous agents. We showed that agents built on time-collapsed metrics can miss key characteristics of human play, in particular an accurate model of temporal reciprocity. While our adaptive agent was able to perform closer to this metric, the key is the identification of time-dependent metrics as a key factor in evaluating emulation agents. This also has implications on the type of agent model necessary to have as a substrate upon which one can learn from human data.

Going forward, we will consider more complex domains and potential corresponding models. We will require both general, parameterized models that can be learned from data, as well as more formal methods for constructing appropriate temporal metrics to automatically evaluate the realism of the learned behaviors.

9. REFERENCES

[1] D. Carmel and S. Markovitch. *Learning Models of Intelligent Agents*. AAAI Press, 1996.

[2] C. R. P. J. Carnevale. Group choice in ultimatum bargaining. *Organizational Behavior and Human Decision Processes*, 72(2):256–279, 1997.

[3] P. Davidsson. Agent-based social simulation: A computer science view. *Journal of Artificial Societies and Social Simulation*, 2002.

[4] R. H. Frank, T. Gilovich, and D. T. Regan. Does studying economics inhibit cooperation? *The Journal of Economic Perspectives*, 7(2):159–171, 1993.

[5] J. Henrich. Does culture matter in economic behavior? ultimatum game bargaining among the machiguenga. *American Economic Review*, 90(4):973–979, 2000.

[6] J. Henrich, S. J. Heine, and A. Norenzayan. The weirdest people in the world? *Behavioral and Brain Sciences*, 33(2-3):61–83, 2010.

[7] E. Hill and D. Sally. Dilemmas and bargains: Theory of mind, cooperation and fairness. Working paper, University College, London, 2002.

[8] S. Judd, M. Kearns, and Y. Vorobeychik. Behavioral dynamics and influence in networked coloring and consensus. In *Proceedings of the National Academy of Science*, 2010.

[9] W. Mao and J. Gratch. Social judgment in multiagent interactions. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '04, pages 210–217, Washington, DC, USA, 2004. IEEE Computer Society.

[10] A. G. S. Mascha van't Wout, René S. Kahn and A. Aleman. Affective state and decision-making in the ultimatum game. *Experimental Brain Research*, 169(4):564–568, 2006.

[11] T. Mitsa. *Temporal Data Mining*. CRC Press, 2010.

[12] H. Oosterbeek, R. Sloof, and G. van de Kuilen. Differences in ultimatum game experiments: Evidence from a meta-analysis. *Experimental Economics*, 7:171–188, 2004.

[13] J. Rickel. Intelligent virtual agents for education and training: Opportunities and challenges. In A. de Antonio, R. Aylett, and D. Ballin, editors, *Intelligent Virtual Agents*, volume 2190 of *Lecture Notes in Computer Science*, pages 15–22. Springer Berlin / Heidelberg, 2001.

A Qualitative Ascending Protocol for Multi-Issue One-to-Many Negotiations

Liviu Dan Şerban, Cristina Maria Ştefanache
Gheorghe Cosmin Silaghi, Cristian Marius Litan
Babeş-Bolyai University, Cluj-Napoca, Romania
{liviu.serban, cristina.stefanache,
gheorghe.silaghi, cristian.litan}@econ.ubbcluj.ro

ABSTRACT

Many practical distributed systems environments require novel automatic mechanisms, including multi-attribute reverse auctions, for efficient partner selection and contracts negotiation. Recent results [2] show that the property of transferable utilities is not of vital importance, as qualitative versions of the standard auctions are proved to exhibit nice efficiency properties as well. Such auctions require that the preferences of the auctioneer are publicly known. Practical protocols of multi-bilateral closed negotiations are experimentally shown [5] to approximate the Pareto-efficient best-seller QVA outcome, without requesting that any of the parties explicitly reveals their preferences. The only condition is to enable bidders to learn preferences. In this paper we introduce and discuss a novel protocol that tries to implement a qualitative ascending English auction, overcoming some restrictions imposed by the non-transferable utilities environment and without being needed that auctioneer reveals his preferences. Our auction-like protocol is designed for fully automatic environments and, when learning agents play the bidders' roles, we expect to reach the QEA outcome.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - intelligent agents, multi-agent systems

General Terms

Algorithms, Economics, Theory

Keywords

Automated multi-issue negotiation, Non-transferable utilities, English auction

1. INTRODUCTION

Classical auctions, especially English and Vickrey are widely studied in the literature and existing theoretical results can guide a mechanism designer to select the proper protocol for trading services in a distributed environment. In the context of independent private-value models, the English and Vickrey auctions exhibit several important properties [6]: (i) the two auctions are strategically equivalent, (ii) the outcome of both is Pareto-optimal and (iii) the winner is the bidder who values the object most highly. In the English auction the dominant strategy of a player is to bid actively until the price announced by the auctioneer reaches the value of the object to him. In the Vickrey auction, if a bidder knows the value of the object to himself, the dominant strategy is to submit a sealed bid equal to that value. These observations are valid in auction models with *transferable utilities*, i.e. there is a good established as currency in the community, bids are expressed in the form of price quotations and the auctioneer simply prefers a higher price to a lower one. From these fundamental assumptions, auctions evolved to environments where a center holds some amount of money, being interested to buy the best good or service he can get for that amount.

In this paper we consider such a reversed auction setup, with one buyer against several sellers, while bids are expressed as bundles of characteristics of the good / service under discussion. Such setups are of interest in (e.g.) service-based computing, where service level agreements can be either individually negotiated or auctions can be employed to select the provider that supplies the best SLA [8].

To address this reversed auction setup, several novel practical and theoretical results exist. Harrenstein et al. [2] defines a *qualitative Vickrey auction* (QVA) protocol that generalizes the classical second-price Vickrey auction for environments without payments. They show that the dominant strategy of each bidder is to make that offer, which among the ones that are acceptable to him, is most preferred by the center. If all bidders adhere to this strategy, the outcome is weakly Pareto-optimal. They also suggest that in an English-like auction the straightforward strategy for each bidder is to offer the highest alternative in his preference order, such that the new bid is preferred by the center to the last submitted bid. In both cases, the results are derived under the main assumption that the center publicly announces his preference profile at the beginning of the auction. Considering this issue and referring to the QVA setup of [2], Hindriks et al. [5] provides a practical protocol that approximates the QVA outcome.

Within the research framework just described, this paper introduces a protocol to practically implement a qualitative ascending English auction (QEA), overcoming the restrictions imposed by the non-transferable utilities environment.

We construct around the principles of ascending English auctions in order to bypass the limitation that the buyer is unable to explicitly elicit his value function or he might not know the complete domain of possible outcomes of the sellers [5]. For a fully automatic environments, we provide an auction-like protocol that is expected to enable learning agents playing the bidders' roles to reach the QEA outcome. The structure of the paper is the following. Section 2 defines the qualitative multi-issues auction environment, including the definition of the theoretical QEA. Section 3 presents a practical protocol implementing the QEA with learning agents on the bidders' side, while the auctioneer's preferences is not public information. Section 4 presents the experimental results. Section 5 concludes the paper.

2. DEFINITIONS

We tackle with a virtual environment where sellers and buyers negotiate over a good or service. Each service has m issues of interest $x = (x_1, x_2, \ldots, x_m)$ with $x \in X = X_1 \times X_2 \times \cdots \times X_m$. Buyers and sellers associate an utility value $u_i(x) \in [0, 1]$ for each outcome x, and each player i has a reservation value v_i below which he does not accept any outcome. The utility functions u_i can be written as *linear combinations* of the individual utility functions $u_{i,k}$: $u_i(x) = \sum_{k=1}^{m} w_k u_{i,k}(x_k)$, with $\sum_{k=1}^{m} w_k = 1$, where $u_{i,k}(x_k)$ represents the utility that the agent i obtains by receiving the value x_k for the issue k and $0 \leq w_k \leq 1$ represent the weights measuring the importance of a given issue k for the agent. Agents prefer a higher utility to a lower one. An outcome x is *weakly Pareto-efficient* if there is no other outcome under which all players are strictly better off.

Further in this paper, without loss of generality, we shall assume a reverse-auction setup with one buyer and many sellers. The goal of the buyer is to design a mechanism that provides an efficient outcome and which is the best possible for him. Formally, the buyer (with the utility function u_0) is interested in selecting the seller i^* such that:

$$i^* = \arg\max_{i \in \{1,\ldots,n\}} \max\{u_0(x) | x \in X, u_i(x) \geq v_i\} \quad (1)$$

As in [2, 5], the above-defined environment is called a *qualitative* one, since no general accepted currency is defined and the bids and outcomes are represented as vectors x.

The classical English auction as described by [6, 2] implies that the center announces the price level (or the acceptable bid in a qualitative setup) and bidders adhere or not to the center's announcements. Variants of the classical English auction exist and among them, the one in which the bidders themselves call the bids and the auction ends when no one is willing to raise the bid. In such an English auction, in the case of independent private-values, the dominant strategy of a bidder is to always bid a small amount over the current highest bid and to stop when his private value is reached [7]. The equivalence between this sort of English auction and the second-price auction holds in a transferable-utility environment with private-value agent models [6]. In non-private English auctions with at least three bidders, the auctioneer is better off than in the Vickrey case, because other bidders' willing to raise the price causes any bidder to increase his own valuation of the item [7].

Given these theoretical assertions, we define the qualitative English auction protocol as in Alg. 1. In QEA, the buyer accepts only bids in increasing order (of the global ordering induced by his preferences) until no bidder is interested any

Algorithm 1 The qualitative English auction

1 The buyer announces his preferences.
While only one bidder remains or no bidder is willing to raise the bid utility of the buyer during a time frame
2. Bidders submit offers with a higher utility for the buyer than the best offer announced by the other bidders.
3. At any time, a bidder can withdraw (without a re-entering possibility)
End while
4. The last remaining bidder wins with his last bid.

more to submit. Therefore, a straightforward seller strategy is to offer the highest alternative in his preference order, that it is higher (in the global ordering of the buyer) than the last submitted bid [2]. On the other hand, in the QVA protocol described in [2], the sellers' dominant strategy is to bid an offer that is just acceptable to himself and ranks highest in the buyer's preferences. Furthermore, QVA selects the best seller and let him to adjust his offer up to the utility of the second best seller.

Fig. 1 presents a comparison between the QEA and the QVA outcomes. Let a buyer asking for an outcome x and let us consider two sellers with reservation values $v_1 > v_2$. In both auctions, seller 2 wins, because it can supply better outcomes for the buyer. In the first round of the QVA, seller 2 announces a bid which is the closest (or identical if possible) to u_2 for the buyer (and to v_2 for himself). This bid is preferred by the auctioneer to the bid offered by seller 1. In the second round, seller 2 is allowed to adjust his bid up to u_1 (the utility for the buyer induced by seller 1's offer). Thus, the game will end in the outcome denoted by QVA.

On the other hand, in QEA both sellers will announce bids giving increasing utility to the buyer. When the outcome reaches the point (u_1, v_1), the first seller will exit, as he is not able to further improve his bid. Thus, only seller 2 remains in the game, with his offer inducing utility u_e for the buyer, i.e. representing the next offer ranked highest in seller 2's preferences, but with increasing utility for the buyer.

We note that the QEA outcome is the next outcome on the Pareto frontier for the above winning seller, such that the utility of the buyer is higher than the one induced by the QVA outcome. If the domain is continuous, the utility u_e

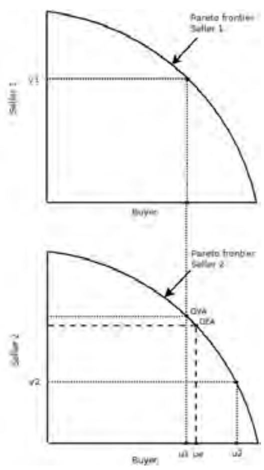


Figure 1: QEA and QVA outcomes

of the QEA outcome is only ϵ-better ($0 < u_e - u_1 < \epsilon$ for a very small $\epsilon > 0$) than the one of the QVA outcome and practically, both outcomes coincide. However, if the domain is discrete (and sparse) and the Pareto frontier is defined by discrete points, a significant difference between the utility of the QEA outcome and the one of the QVA may occur. Therefore, in the discrete case, a center willing to improve his outcome, might select the QEA.

3. IMPLEMENTING THE QEA

In this section we present a protocol that tries to practically implement the QEA. In the theoretical QEA (see alg. 1), to submit bids with increasing utility for the buyer, sellers need to know the buyer's preferences or the buyer's utility function. If, because of various reasons, the buyer cannot reveal his preferences, another signaling procedure should be used, such that the sellers to be notified to deliver increasing value bids to the buyer. Thus, we propose the auction-like protocol of Alg. 2.

We assume that each seller *learns* the buyer's profile from his bids, exactly like in an one-to-one bilateral negotiation. Initially, all buyer's profiles from sellers' point of view are blank, thus, each seller initializes to zero the threshold $U_{0,i}^{exp}$ for the expected utility for the buyer. In each round, each seller i who has not reached his reservation value v_i, submits a bid $x^{(i)}$ such that this bid is better off for the buyer: $u_{0,i}^{exp}(x^{(i)}) > U_{0,i}^{exp}$. More precisely, for a rational seller, this bid will be exactly the next one in decreasing order, following his preference ordering, which increases the expected utility for the buyer. If, for a given seller i, there are no more bids which satisfy the condition above, that particular seller should withdraw from the negotiation.

We notice that the buyer reveals little information, in each round he just signals the best submitted bid to all sellers. Each seller, who is capable of learning the opponents' profiles, will infer both the buyer's and the best competitor's profile. Further, the round winning seller will know that he won the round, because his bid was repeated by the buyer. To keep the similarity with the QEA, we translated the exiting condition of line 3 in Alg. 1 in the `while` condition of Alg. 2. This condition states that a seller will remain in the protocol while he has offers, acceptable to him, that are greater than the threshold of the round. If, within a round, a seller does not submit an offer, this means that the seller exited the auction.

One possible problem with this protocol is that each bidder should submit an offer in a reasonable amount of time. In this first version of the protocol, we assume that the sellers are not trying to exploit the time restrictions, thus, if a seller has an available offer, it will present this offer, if not, he will immediately withdraw. A similar behaviour is obtained if we impose a short time limit for every round. If a seller does not respond with an offer within the time limit, he is considered withdrawn from the negotiation.

Note that the protocol proposed here has many nice practical properties. First, it does not require the buyer to make public his profile. Furthermore, even if the buyer does not a priori know the range of the possible offered outcomes, this fact does not represent a problem for the protocol, as the buyer is not required (at any time) to produce an outcome. All what buyer is requested is that he is able to rank the possible incoming offers. Second, the protocol is ascending - simulating an English auction; at every round sellers are being presented the highest bid up to the moment. Third, the protocol can simulate a non-private value setup, the sellers being actually able to learn competitors' profiles by observing the best bid in each round. Thus, the protocol might yield a higher utility for the buyer.

The existence of learning agents is essential for the success of the protocol presented in Alg. 2. Learning agents usually perform two steps when producing an offer: (i) process all information from the environment to infer a model for the environment, auctioneer and competitors; (ii) select and propose the next offer. We consider that a learning agent of this type can be easily adapted to the protocol presented in Alg. 2. This is because, from his point of view, the selling agent plays a game like in the one-to-one negotiation.

Algorithm 2 The practical implementation of the QEA

1. Each seller i sets $U_{0,i}^{exp} = 0$

While sellers i have offers $x^{(i)}$ such that $u_{0,i}^{exp}(x^{(i)}) > U_{0,i}^{exp}$ and $u_i(x^{(i)}) \geq v_i$, for those sellers i perform:

2.1 Each seller submits an offer $x^{(i)}$ with expected buyer utility higher than $U_{0,i}^{exp}$

2.2 The buyer selects the offer $x^{(i^*)}$ with the highest utility $u_0(x^{(i^*)})$

2.3 The buyer announces the bid $x^{(i^*)}$ to every seller

2.4 Each seller updates its $U_{0,i}^{exp}$ to the expected utility value for the bid $x^{(i^*)}$

end while

3. The last remaining bidder wins with the his last bid

4. EXPERIMENTS AND RESULTS

In this section we present the experimental results of the protocol presented in Alg. 2 compared to the theoretical outcome of the QEA

4.1 The experimental setup

In this subsection we present the experimental setup. To assure the comparability of results, we used the same negotiation domain and only two sellers, as in [5]. In the future, we shall extend the results to many sellers, as the simulation software allows it.

To simulate the negotiation, we used a modified version of the Genius simulator [3], allowing for one-to-many negotiations. The service-oriented negotiation domain described in [1] was used as testbed. This domain refers to negotiating a service with four issues of interest: delivery time, quality, duration and penalty, each issue having 30 possible values. Thus, there are 810000 total number of possible agreements.

Adapted for the one-to-many negotiation, we used the 12 preference profiles per role that were considered in [5]. Playing one buyer against two sellers results a total of 792 possible games. We conducted all our experiments on a sample of size 50 out of the full population of 792 possible auctions. To work with the Genius simulator, we adapted the Bayesian learning agent [4] for playing the protocol described in Alg. 2 as seller. This adapted Bayesian agent will never accept a bid proposed by the buyer and will withdraw when no further possible bids are available. The condition to select the next bid is further restricted by considering for selection only bids that have the estimated utility for the buyer higher than

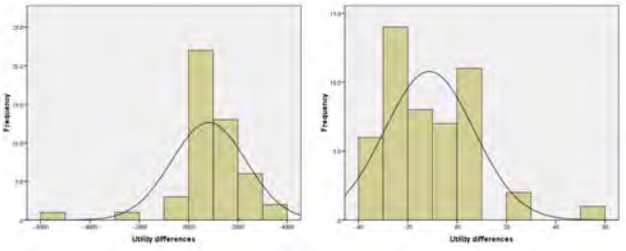


Figure 2: Utility difference between the practical protocol and the QEA on the buyer side (left) and on the seller side (right)

			95% conf. interval	
Agent	Mean	Std. Dev.	Lower	Upper
buyer	0.0821	0.1499	0.0386	0.1256
seller	-0.1096	0.1801	-0.1619	-0.0573

Table 1: Synthetic statistical results for utility differences between the practical protocol and the theoretic QEA

buyer's last announced bid. When winning the round, the Bayesian agent will repeat the winning bid, up to the moment that one competitor submits a better bid.

4.2 The practical protocol against QEA

This subsection presents a comparison between the outcomes of the practical protocol and those generated by the theoretical QEA. Fig. 2-left presents the distributions of utility differences for the buyer and Fig. 2-right for the winning seller[1]. Table 1 presents synthetic statistical results. In all 50 runs, the outcome is situated on the Pareto frontier.

In Fig. 2 one can notice that the practical protocol statistically produces a better outcome for the buyer than the theoretical QEA. Table 1 shows the synthetic statistical results and the 95% confidence intervals for the buyer and for the winning seller. The average of the differences are significantly different from 0, clearly positively biased for the buyer and negatively biased for the seller. Out of the 50 runs, only in 5 cases the outcome is situated on the left side of the QEA outcome. Therefore, the practical protocol produces results under which the buyer is better off, while the winning seller is worse off than in the theoretical case.

This may be due to several reasons. First, the protocol does not entirely fit the private-value model. The sellers are learning both buyer's preferences and those of the competitors (mostly the latter preferences). Thus, as indicated in the literature [7], we can expect the buyer to obtain better results than in the private-value case, and clearly better than in the second-price auction.

The second possible explanation for these results come from the fact that the sellers are learning agents that are meant to only estimate the buyer's profile. This estimation may not be perfect; it often happens that the Bayesian agent to retrieve an offer better for him than the theoretical QEA outcome. In this case, if the buyer returns back this offer, the ascending property of the protocol does not allow the selling agent to propose the bid again, even if, based on the re-estimation of buyer's profile within the future rounds, the offer would represent for the seller a maximizing behavior fulfilling the right constraints.

[1]The solid line of Fig. 2 represents the standard normal distribution with mean 0 and variance 1

5. CONCLUSIONS

In this paper we propose a protocol to practically implement the qualitative English auction for multi-issue auction setups with non-transferable utilities. Qualitative Vickrey auction and, by extension, qualitative English auction are proved [2] to possess nice theoretical properties, generalizing the classical Vickrey auction to environments without an established currency. The proposed protocol is expected to enable learning agents playing the bidders' roles, to reach the QEA outcome in practical experiments. We consider this protocol of practical usage, as the center is not required to reveal his profile at the start of the auction. Experiments show that this version of our protocol produces outcomes that are still Pareto-efficient, however with a better off buyer and worse off winning seller than in the case of theoretical QEA (or QVA). This may be due to the fact that the protocol is not entirely fit into the private-value model. Here, the sellers are learning both the buyer's preferences and also those of the competitors.

Acknowledgements

We acknowledge support from the Romanian Authority for Scientific Research under project IDEI_2452. C.M. Stefanache acknowledges support from the European Social Fund under the Operational Sectoral Programme Human Resources Development 2007-2013 "Investing in People".

6. REFERENCES

[1] P. Faratin, C.Sierra, and N. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205–237, 2002.

[2] B. P. Harrenstein, M. M. de Weerdt, and V. Conitzer. A qualitative vickrey auction. In *The 10th ACM Conf. on Electronic Commerce*, pages 197–206. ACM, 2009.

[3] K. Hindriks, C. M. Jonker, S. Kraus, R. Lin, and D. Tykhonov. Genius: negotiation environment for heterogeneous agents. In *AAMAS'09: The 8th Intl. Conf. on Autonomous Agents and Multiagent Systems*, pages 1397–1398. IFAAMAS, 2009.

[4] K. Hindriks and D. Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *AAMAS'08: The 7th Intl. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 331–338. IFAAMAS, 2008.

[5] K. Hindriks, D. Tykhonov, and M. de Weerdt. Qualitative one-to-many multi-issue negotiation: Approximating the qva. *Group Decision and Negotiation*, pages 1–29, 2010.

[6] P. Milgrom and R. Webber. A theory of auctions and competitive bidding. *Econometrica*, 50(5):1089–1122, 1982.

[7] T. W. Sandholm. *Distributed rational decision making*, pages 201–258. MIT Press, 1999.

[8] O. Waeldrich, D. Battre, F. Brazier, K. Clark, M. Oey, A. Papaspyrou, P. Wieder, and W. Ziegler. Ws-agreement negotiation 1.0. In *OGF Grid Resource Allocation Agreement Protocol (GRAAP) WG*, 2010.

EAF-based Negotiation Process

Paulo Maio
GECAD – ISEP - IPP
School of Engineering
Polytechnic of Porto
Porto, Portugal
pam@isep.ipp.pt

Nuno Silva
GECAD – ISEP - IPP
School of Engineering
Polytechnic of Porto
Porto, Portugal
nps@isep.ipp.pt

José Cardoso
University of Trás-os-Montes
and Alto Douro
Vila Real, Portugal
jcardoso@utad.pt

ABSTRACT
Agents participating in a negotiation dialogue may use argumentation to support their position, hence achieving a better agreement. The Extensible Argumentation Framework (EAF) provides modularity and extensibility features that facilitates its adoption by agents in MAS. In order to emphasize the EAF potential and applicability, this paper proposes an argument-based negotiation process grounded on the EAF adoption.

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

Keywords
Negotiation, Argumentation, Agents, MAS

1. INTRODUCTION

Internally agents may use argumentation for both (i) reasoning about what to believe (i.e. theoretical reasoning) and/or (ii) for deciding what to do (i.e. practical reasoning). Despite existing differences between both, from a standpoint of first-personal reflection, a set of considerations for and against a particular conclusion are drawn on both [1]. On the other hand, concerning the types of agents' dialogues (e.g. Deliberation, Negotiation, Persuasion, Inquiry, Information-seeking dialogues), while a clear distinction between each one exist, most of agents' dialogue occurrences involve mixtures of dialogue types. Within this context, argumentation is seen as an activity where each participant tries to increase (or decrease) the acceptability of a given standpoint for the others participants by presenting arguments. In particular, agents participating in a negotiation dialogue may use argumentation to support their position and by that achieve a better agreement. Therefore, argumentation is foreseen as an adequate modeling formalism to reduce the gap between models governing the internal and external agent behavior. Grounded on that, this paper presents a generic negotiation process that exploits the expressivity, modularity and extensibility features of the Extensible Argumentation Framework (EAF) [2]. The core idea behind the EAF-based process is: while a common argumentation vocabulary is shared by all agents, internally each agent is able to extend that vocabulary to fit its own needs and knowledge.

The rest of this paper is organized as follows: the next section describes the main structures and concepts of the EAF. Section 3 presents the proposed negotiation process based on the adoption of EAF in MAS. Section 4 presents a brief summary of performed experiments in the domain of ontology alignment [3] applying the proposed negotiation process. Finally, section 5 draws conclusions and comments on future work.

2. The EAF

This section describes briefly and informally the main features of the Extensible Argumentation Framework (EAF). The EAF comprehends three modeling layers as depicted in Figure 1.

The Meta-model layer defines the core argumentation concepts and relations holding between them. EAF adopts and extends the minimal definition presented by Walton in [4] where "an argument is a set of statements (propositions), made up of three parts, a conclusion, a set of premises, and an inference from premises to the conclusion". For that, the meta-model layer defines the notion of *Argument*, *Statement* and *Reasoning Mechanism*, and a set of relations between these concepts. An argument *applies* a reasoning mechanism (such as rules, methods, or processes) to *conclude* a conclusion-statement from a set of premise-statements. Intentional arguments are the arguments corresponding to intentions ([5], [6]).

The Model layer defines the entities and their relations for a specific domain according to a community's perception. The resulting model is further instantiated at the Instance-pool layer. The R relation is established between two argument types (e.g. $(C, D) \in R$) when C supports or attacks D. Through R it is also determined the types of statements that are admissible as premises of an argument. Additionally, arguments, statements and reasoning mechanisms can be structured through the H_A, H_S and H_M relations respectively. These are acyclic transitive relations established between similar entity types (e.g. arguments), in the sense that in some specific context entities of type e_1 are understood as entities of type e_2. While these relations are vaguely similar to the specialization relation (i.e. subclass/superclass between entities) it does not have the same semantics and it is constrained to 1-1 relationship.

The Instance-Pool layer corresponds to the instantiation of a particular model layer for a given scenario. A statement instance B_1 is said to be in conflict with another statement instance B_2 when B_1 states something that implies or suggests that B_2 is not true. The statement conflict relation is asymmetric (in Figure 1 B_2 conflicts with B_1 too). The support and attack relationships (R_{sup} and R_{att} respectively) between argument instances are automatically inferred exploiting (i) the R relations defined at the model layer and (ii) the existing *premise* relations and the statements conflicts at this level.

An EAF model may reuse and further extend the argumentation conceptualizations of several existing EAF models. Inclusion of an EAF into another EAF is governed by a set of modularization constraints ensuring that no information of included EAF is lost.

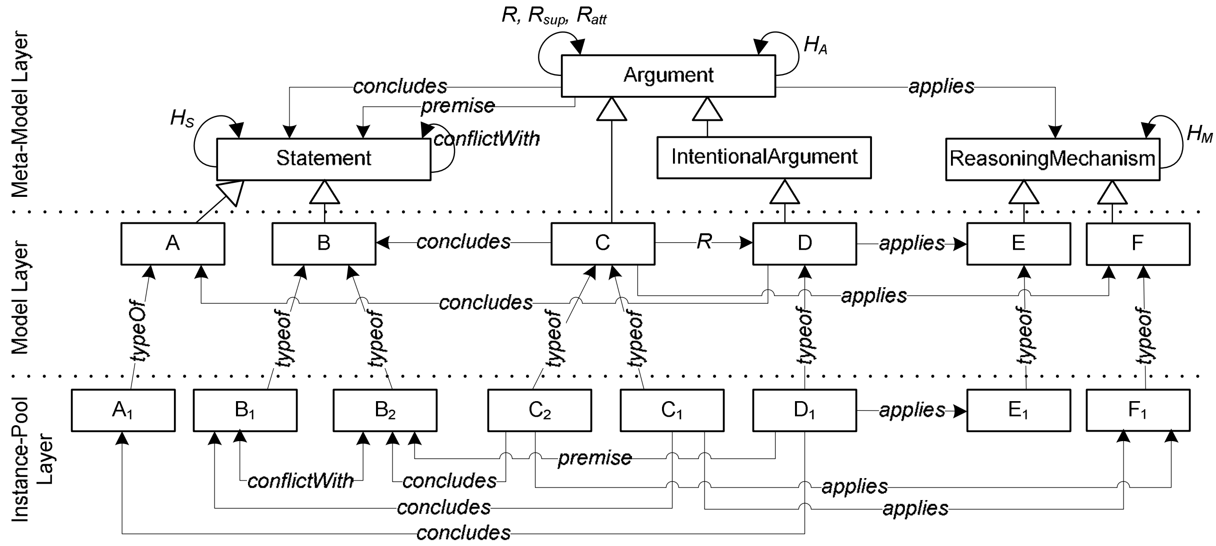


Figure 1. The three modeling layers of EAF

3. NEGOTIATION PROCESS

This section proposes a negotiation process based on the adoption of EAF by agents in MAS. While other negotiation processes using EAF are admissible, we aim to provide an end-to-end negotiation process that emphasizes its potential and applicability.

The proposed negotiation process relies on the following assumptions: (i) a negotiation process between two or more agents always occurs in the scope of a given community of agents, (ii) the agents' community is able to define an EAF model (i.e.EAF_c) representing the community minimal common understanding about the domain of discourse that all agents of that community are able to understand, and (iii) each agent is able to exploit the modularization and extensibility features of the EAF such that each agent is free to internally extend the common argumentation model so it better fits its own needs and knowledge. Concerning latter assumption, it is especially relevant the application of the H_A, H_S and H_M relations so that the agent explicitly states the specialization of its individual EAF model (i.e. EAF_{Ag}) in respect to EAF_c. These relations will provide a minimal common classification of arguments, statements and reasoning mechanisms introduced individually by each agent. Based on these assumptions, we propose the EAF-based negotiation process to be adopted by each negotiating agent.

The negotiation process specifies nine phases (Figure 2).

3.1 Setup

The Setup phase encompasses a set of domain-dependent interactions between agents such as: (i) the identification of the (possible) negotiation participants, (ii) the identification of the negotiation object, (iii) the identification of which is the community minimal common understanding (i.e. EAF_c) between all participants, (iv) the definition of the required negotiation parameters/constraints such as deadline for achieving an agreement, (v) the specification of the arguments exchanging method used by each agent, (vi) the specification of the negotiation method to compute a possible agreement between participants (e.g. by consensus between all participants or by the majority of participants opinions), (vii) the establishment of special rights for some of the participants, (viii) sharing the

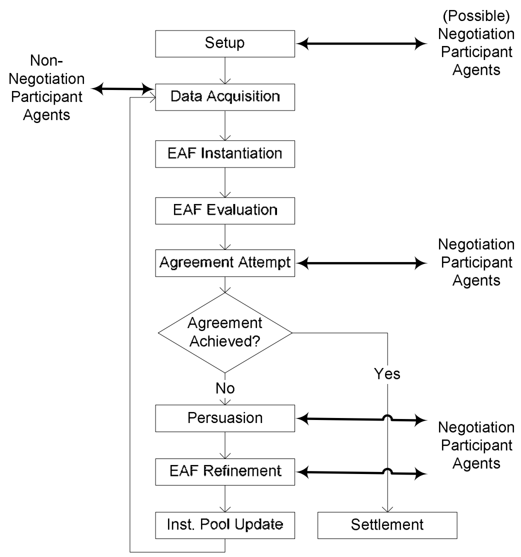


Figure 2. EAF-based negotiation process

data/information that is required by the agents in order to participate in the negotiation. At the end of this phase, the context of the EAF-based negotiation is completely defined and known by all participating agents. Therefore, such context must be uniquely identified and further participants' interactions related with that content must be clearly stated as so. Yet, such context defines a set of constraints called negotiation parameters (i.e. NP). Moreover, each participant creates an instance-pool of its own EAF (i.e. $IP(EAF_{Ag})$) that will capture the argumentation data. Contrary to the other phases, this phase occurs only once.

3.2 Data Acquisition

During the Data Acquisition phase the agent collects, from the environment, a set of data/information (called D_{Ag}) that constitutes the grounds to generate arguments. The agent may rely on a communication process with other agents (non-participating directly in the ongoing negotiation), namely specialized agents on the subject under discussion.

3.3 EAF Instantiation

The goal of the EAF Instantiation phase is to analyze and to process the collected data (i.e. D_{Ag}) in order to add and/or update the instances (e.g. argument-instances) in the respective EAF instance-pool. For that, the agent makes use of one or more data transformation processes whose output is a set of unclassified (or partially classified) EAF_{Ag} instances. Next, those instances are properly (re)classified as required by the EAF. An EAF instances (re)classification process is also needed further in the Instance-Pool Update phase. In that sense, it is envisaged that the instances (re)classification process might be the same in both phases, however that is not mandatory.

3.4 EAF Evaluation

In the EAF Evaluation phase, each agent extracts a *preferred extension*, i.e. a consistent position within EAF_{Ag} which is defensible against all attacks and cannot be further extended without introducing a conflict. According to the agent's $IP(EAF_{Ag})$ one or more possible *preferred extensions* may be extracted. If the EAF-evaluation process extracted more than one *preferred extension* then it is necessary to select one. Notice that the selection criterion has a special relevance during the negotiation process, because it directly defines the public agent's position about the subject under discussion (i.e. its *intentions* and the *beliefs* behind those *intentions*). Given that, instead of a simple criterion such as "selection of the preferred extension that is maximal with respect to set inclusion", a more elaborated selection criterion may take into consideration the *preferred extension* previously selected (if there is any) in order to select, for example, the one that differs less. This phase may occur more than once due to new data/information acquisition and especially due to the exchange of arguments between the agents during the persuasion phase. Thus, any change made to $IP(EAF_{Ag})$ suggest that the agent's consistent position may change, hence requiring a re-evaluation of the *preferred extension* by the agent.

3.5 Agreement Attempt

In the Agreement Attempt phase each participant makes a proposal of agreement to the other agents in order to find out an overall common agreement (called candidate agreement) which can be accepted and further settled by all participants. It comprehends two steps. In the first step, each agent makes its proposal of agreement by exchanging the intentional argument of its *preferred extension* only (called *intentional preferred extension*). As a result of all proposals, two sets of arguments are derived and then shared by all agents: (i) the set of arguments agreed/proposed by all agents ($AgreedArgs$) which represents a candidate agreement and (ii) the set of arguments which at least one agent disagrees ($DisagreedArgs$). For a negotiation between n agents where $iprefext_{Ag_i}$ is the *intentional preferred extension* of agent i, these sets can be computed differently depending on the agents and according to the setup phase. One of the simplest agreement evaluation forms is based on their intersection:

$$AgreedArgs = \bigcap_{i=1}^{n} iprefext_{Ag_i}$$

$$DisagreedArgs = \left(\bigcup_{i=1}^{n} iprefext_{Ag_i}\right) - AgreedArgs$$

In the second step, each participant evaluates its level of satisfaction of the current candidate agreement. For that the agent considers the defined negotiation parameters/constraints (i.e. NP) and the content of the $DisagreedArgs$ set. According to the level of satisfaction, the participants must decide to either:

- Continue the negotiation, and therefore proceed to the persuasion phase, or
- Conclude the negotiation, which is either:
  - successful if all agents accept the candidate agreement ($AgreedArgs$). In such case the process proceeds to the settlement phase, or
  - unsuccessful if the candidate agreement is not accepted by all agents and it was considered that it is not worth keep trying to achieve another candidate agreement. The negotiation ends without an agreement.

3.6 Persuasion

From previous phase it has been identified a set of intentional arguments that are not accepted by at least one participant (i.e. $DisagreedArgs$). In the this phase, each agent, first selects from its *preferred extension* a (sub-) set of arguments supporting or attacking the intentional arguments in $DisagreedArgs$, which will further be exchanged with the opponent agents to persuade them. There are two forms to exchange the arguments:

1. The arguments are exchanged according to the EAF_c and not according to EAF_{Ag}, so the other agents can understand them. Due to the H_A, H_S and H_M relations, the transformation of the instances respecting the agent's EAF to the community EAF is straightforward.
2. The arguments are exchanged according to the EAF_{Ag} along with the EAF_{Ag} parts that allow the other agent to transform the arguments to EAF_c.

The way the arguments are exchanged is defined in the setup phase, and will have implications in next phase.

Yet, independently of the exchanged method, at the end of this phase, each agent has collected a new set of information (ED_{Ag}), corresponding to the received arguments presented by the other negotiating agents.

3.7 EAF Refinement

This phase concerns the refinement of the community's EAF model according to the exchanged arguments and the agents' EAF models. Therefore, if the exchange of arguments does not include exchanging parts of the agent's EAF model, this phase is more difficult and therefore may be skipped. It is not the aim of this description to present an EAF evolution process, nor the agents' reasoning process leading to such evolution. Instead, this description intends to emphasize the ability of the EAF to be extended according the agent's needs, by exploiting the modularization features of the proposed argumentation framework.

3.8 Instance Pool Update

In this phase, the agent reclassifies the ED_{Ag} data according to its EAF_{Ag} applying an instance (re)classification process, which might be the same used in the EAF Instantiation phase. The reclassified data that do not exist into $IP(EAF_{ag})$ is added while duplicated arguments are discarded. Added arguments are taken into consideration by the agent in the next round of proposals. The negotiation process proceeds to the Data Acquisition phase.

3.9 Settlement

The goal of the settlement phase is to transform the candidate agreement into a definitive agreement according to the settlement parameters of NP. Depending of the domain of application and the negotiation object (e.g. a good or a service) as well as the participating partners, the settlement phase can have a varying of sub-functions. In this respect, this phase is seen as an initiator of a set of transactions that must occur after the agreed terms are known in order to fulfill the terms. For example, in an e-commerce scenario, to fulfill an agreement for selling a physical good may imply to carry on logistic and financial services.

4. EXPERIENCES

Since the proposed negotiation approach is domain independent, to carry out some experiments we need to choose a domain of application. We applied the EAF-based negotiation approach to address conflicts arising between agents when they are reconciling the vocabulary used in their ontologies. The result of the vocabulary reconciliation is a set of correspondences (i.e. an alignment) between entities of agents' ontologies. Such conflicts arise because each agent may have its own perspective about what are the best correspondences. In that sense, the experiments aim to measure the improvement produced on the accuracy (in terms of precision, recall and f-measure) of the agreed alignment by the negotiation process when compared to each agent's initial proposal, i.e. before the negotiation process runs. For this purpose, we adopted an empirical approach using (i) a set of pairs of publicly available ontologies used in several ontology alignment experiences as, for example, the Ontology Alignment Evaluation Initiative (OAEI) and (ii) for each pair of ontologies a widely accepted reference alignment that will be exploited to evaluate the negotiation results.

For the sake of brevity and simplicity, the results are presented considering the negotiation of all individual alignments as just one huge alignment. The reference alignment contains 1402 correspondences (also referred as matches) corresponding to the sum of all correspondences of all reference alignments. We have configured three agents, each one using a distinct set of matching algorithms and a distinct EAF model (extended from a common one).

Table 1 summarizes and characterizes the automatic alignment of each agent before the negotiation process runs. Correct matches are those that exist in the reference alignment.

Table 1. Agents' alignment before the negotiation process

Agent	Matches		Accuracy %		
	Proposed	Correct	Precision	Recall	F-Measure
A	1358	1296	95.4	92.4	93.9
B	2025	1266	62.5	90.3	73.9
C	1290	1219	94.5	86.9	90.6

Table 2 summarizes and characterizes the agreed alignment between each possible pair of agents. It also shows (i) on column "U." the amount of matches under discussion on the beginning of the negotiation process i.e. the agents' contradictory initial position, (ii) on column "U.C" the amount of correct matches under discussion, (iii) on column "R." the amount of matches that even after the negotiation process remain contradictory, (iv) on column "R.C." the amount of correct matches remaining with contradictory position and (v) on column "G.P." the percentage of good persuasion occurred, i.e. one of the agents concede its initial position in favor of the opponent agent's position, and that concession contributes positively for the quality of the achieved agreement.

Table 2. Agreed alignment between agents

Agent	Matches						Accuracy (%)			G.P. %
	P.	C.	U.	U.C.	R.	R.C.	Pr.	Re.	F-M.	
A-B	1294	1243	813	78	308	67	96.1	88.7	92.2	96.4
A-C	1250	1214	200	119	130	90	97.1	86.6	91.6	67.1
B-C	1387	1234	779	75	220	39	89.0	88.0	88.5	82.6

Examination of results shows that independently of the amount of resolved conflicts, the percentage of good persuasion is always high and consequently the negotiation process is beneficial to the overall accuracy of the agreed alignment. Yet, it also perceived that it is very hard for an agent to successfully persuade its opponent to change position about a correct match proposed by its opponent.

5. CONCLUSIONS

This paper describes a novel, generic and domain independent argument-based negotiation process based on the adoption of the Extended Argumentation Framework. Due to the modeling, modularity and extensibility features of the EAF, agents are able to share an external common argumentation model which is further extended internally by each agent to better fit its own needs and knowledge. Yet, the common argumentation model may continuously evolve along the time profiting from occurring negotiation interaction between agents (section 3.7). The proposed negotiation process also promotes the use of argumentation as a common formalism either for (i) agents' internal reasoning and (ii) agents interactions (namely negotiation interactions).

Experiences in the ontology alignment field show that the adoption of the EAF-based negotiation process leads to a substantial improvement in the quality of the agreed ontology alignment when compared with the intersection of agents' individual ontology alignment. The good persuasion is achieved both by persuading the opponent by accepting a correct match and by rejecting an incorrect match.

An interesting research direction concerns providing agents with the ability (i) to learn and improve their argumentation strategies based on their past experiences and (ii) to learn (and understand) new arguments used by other agents in order to apply in the Community's EAF Update phase.

6. ACKNOWLEDGMENTS

This work is partially supported by the Portuguese MCTES-FCT project COALESCE (PTDC/EIA/74417/2006).

7. REFERENCES

[1] R. Moran, *Authority and Estrangement: An Essay on Self-Knowledge*. Princeton University Press, 2001.

[2] P. Maio and N. Silva, *Technical Report: The Extensible Argumentation Framework*. Porto, Portugal: ISEP, 2011.

[3] J. Euzenat and P. Shvaiko, *Ontology Matching*, 1st ed., vol. 1, 1 vols. Heidelberg, Germany: Springer-Verlag, 2007.

[4] D. Walton, "Argumentation Theory: A Very Short Introduction," in *Argumentation in Artificial Intelligence*, Springer Publishing Company, Incorporated, 2009.

[5] M. Bratman, *Intention, Plans and Practical Reason*. Cambridge, MA: Harvard University Press, 1987.

[6] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Wiley, 2009.