

# Effective Acceptance Conditions in Real-time Automated Negotiation

Tim Baarslag, Koen Hindriks, Catholijn Jonker

*Interactive Intelligence Group  
Delft University of Technology  
Mekelweg 4, Delft, The Netherlands  
{T.Baarslag, K.V.Hindriks, C.M.Jonker}@tudelft.nl*

---

## Abstract

In every negotiation with a deadline, one of the negotiating parties must accept an offer to avoid a break off. As a break off is usually an undesirable outcome for both parties, 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 before time runs out. 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 perform extensive experiments to compare the performance of various acceptance conditions in combination with a broad range of bidding strategies and negotiation scenarios. Furthermore we propose new acceptance conditions and we demonstrate that they outperform the other conditions. We also provide insight into why some conditions work better than others and investigate correlations between the properties of the negotiation scenario and the efficacy of acceptance conditions.

*Keywords:* Automated negotiation, Real-time bilateral negotiation, Acceptance criteria, Acceptance conditions, When to accept

---

## 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, 19, 24]). The strategic-negotiation model has a wide range of applications, such as resource and task allocation mechanisms, conflict resolution mechanisms, and decentralized information services [19, 32].

A number of successful negotiation strategies have already been established both in literature and in implementations, (e.g. [6, 8, 7, 13, 14, 23]). And more recently, in 2010 seven new negotiation strategies were created to participate in the first automated negotiating agents competition (ANAC 2010) [5] 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. It is important for every negotiator to employ such a mechanism to decide under which conditions to accept, because in every negotiation with a deadline, one of the negotiating parties has to accept in order to avoid a break off. 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 [33].

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 acceptance 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. The reason for this is straightforward: we want to be able to re-incorporate the acceptance condi-

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

Table 1: An overview of the rank of every agent in ANAC 2010 and the type of acceptance conditions that they employ. Agents using time and utility based acceptance conditions were ranked at the top, except for *Agent Smith*, which had a faulty acceptance mechanism.

tions that have been found most effective into new agent designs; therefore, the acceptance conditions under investigation should not be coupled with a specific agent implementation.

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 large selection of current generic acceptance conditions and evaluate them in an experimental setting.
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 6 and 7 outline 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 that 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  $\Omega$  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  $\Omega$ . 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 [28] for bilateral negotiation, in which the negotiating parties exchange offers in turns.

As in [31], we assume a common global time, represented here by  $\mathcal{T} = [0, 1]$ . We supplement the alternating-offers protocol with a deadline at  $t = 1$ , at which moment both agents receive utility 0. This is the same setup as [9], with the exception that issues are not necessarily real-valued and both agents have the same deadline equal to  $t = 1$ . We represent by  $x_{A \rightarrow B}^t$  the negotiation outcome proposed by agent  $A$  to agent  $B$  at time  $t$ . A *negotiation thread* (cf. [8, 31]) between two agents  $A$  and  $B$  at time  $t \in \mathcal{T}$  is defined as a finite sequence

$$H_{A \leftrightarrow B}^t := (x_{p_1 \rightarrow p_2}^{t_1}, x_{p_2 \rightarrow p_3}^{t_2}, x_{p_3 \rightarrow p_4}^{t_3}, \dots, x_{p_n \rightarrow p_{n+1}}^{t_n}), \quad (1)$$

which satisfies the following constraints:

1.  $t_k \leq t_l$  for  $k \leq l$ , 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  $t_n \leq t$ ,
4.  $x_{p_k \rightarrow p_{k+1}}^{t_k} \in \Omega$  for  $k \in \{1, \dots, n\}$ , the agents exchange complete offers.

Additionally, the last element of  $H_{A \leftrightarrow B}^t$  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  $x_{B \rightarrow A}^t$  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  $x_{A \rightarrow B}^{t'}$ . Given a negotiation thread  $H_{A \leftrightarrow B}^t$  between agents  $A$  and  $B$ , we can formally express the action performed by  $A$  with an *action function*  $X_A$ :

$$X_A(t', x_{B \rightarrow A}^t) = \begin{cases} End & \text{if } t' \geq 1 \\ Accept & \text{if } \mathbf{AC}_A(t', x_{A \rightarrow B}^{t'}, H_{A \leftrightarrow B}^t) \\ Offer\ x_{A \rightarrow B}^{t'} & \text{otherwise} \end{cases} \quad (2)$$

Note that we extend the setting of [9, 31] by introducing the *acceptance condition*  $\mathbf{AC}_A$  of an agent  $A$ . When used in this way, the model enables us to study arbitrary decoupled acceptance conditions. The acceptance condition  $\mathbf{AC}_A$  takes as input

$$\mathcal{I} = (t', x_{A \rightarrow B}^{t'}, H_{A \leftrightarrow B}^t), \quad (3)$$

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

The resulting action given by the function  $X_A(t', x_{B \rightarrow A}^t)$  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  $x_{A \rightarrow B}^{t'}$  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 [31]. 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

$$H_{A \leftrightarrow B}^t = \left( x_{p_1 \rightarrow p_2}^{t_1}, x_{p_2 \rightarrow p_3}^{t_2}, \dots, x_{A \rightarrow B}^{t_{n-1}}, x_{B \rightarrow A}^{t_n} \right),$$

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

As defined by Eq. (1) 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, 10, 31], use the following implementation of  $\mathbf{AC}_A(\mathcal{I})$ :

$$\mathbf{AC}_A(t', x_{A \rightarrow B}^{t'}, H_{A \leftrightarrow B}^t) \iff U_A(x_{B \rightarrow A}^t) \geq U_A(x_{A \rightarrow B}^{t'}).$$

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  $x_{A \rightarrow B}^{t'}$ . For  $\alpha, \beta \in \mathbb{R}$  this may be generalized as follows:

$$\mathbf{AC}_{\text{next}}^{\mathcal{I}}(\alpha, \beta) \stackrel{\text{def}}{\iff} \alpha \cdot U_A(x_{B \rightarrow A}^t) + \beta \geq U_A(x_{A \rightarrow B}^{t'}). \quad (4)$$

We can view  $\alpha$  as the scale factor by which we multiply the opponent's bid, while  $\beta$  specifies the minimal 'utility gap' [14] that is sufficient to accept.

Analogously, we have acceptance conditions [11, 14, 30, 36] that rely on the agent's *previous* offer  $x_{A \rightarrow B}^{t_{n-1}}$ :

$$\mathbf{AC}_{\text{prev}}^{\mathcal{I}}(\alpha, \beta) \stackrel{\text{def}}{\iff} \alpha \cdot U_A(x_{B \rightarrow A}^t) + \beta \geq U_A(x_{A \rightarrow B}^{t_{n-1}}). \quad (5)$$

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  $t$ . However, it is important to bear in mind that the behavior of the acceptance condition may still be influenced implicitly by these factors, because of the possibility that the bidding strategy takes such factors into account.

Other acceptance conditions may rely on other measures, such as the remaining negotiation time or a utility threshold. For example, there is a very simple acceptance criterion [30, 33, 36] that only compares the opponent's previous offer with a threshold  $\alpha$ :

$$\mathbf{AC}_{\text{const}}^{\mathcal{I}}(\alpha) \stackrel{\text{def}}{\iff} U_A(x_{B \rightarrow A}^t) \geq \alpha. \quad (6)$$

Last but not least, instead of considering utility, agents (such as [33]) may employ a time-based condition to accept after a certain amount of time  $T \in \mathcal{T}$  has passed:

$$\mathbf{AC}_{\text{time}}^{\mathcal{I}}(T) \stackrel{\text{def}}{\iff} t' \geq T. \quad (7)$$

We will omit the superscript  $\mathcal{I}$  in equations (4) to (7) when it is clear from the context. We will use these general acceptance conditions to classify existing acceptance mechanisms in the next section.

<b>AC</b>	$\alpha$	$\beta$	<b>Agent</b>
<b>AC<sub>prev</sub></b> ( $\alpha, \beta$ )	1.03	0	<i>FSEGA</i> , <i>Bayesian Agent</i>
	1	0	<i>Agent Smith</i>
	1.02	0	<i>IAM(crazy)Haggler</i>
	1	0.02	<i>ABMP</i>
<b>AC<sub>next</sub></b> ( $\alpha, \beta$ )	1	0	<i>FSEGA</i> , <i>Boulware</i> , <i>Conceder</i> , <i>Trade-off</i> , <i>Equilibrium strategies</i>
	1.02	0	<i>IAM(crazy)Haggler</i>
	1.03	0	<i>Bayesian Agent</i>
<b>AC<sub>const</sub></b> ( $\alpha$ )	1	-	<i>FSEGA</i>
	0.9	-	<i>Agent Smith</i>
	0.88	-	<i>IAM(crazy)Haggler</i>
<b>T</b>			
<b>AC<sub>time</sub></b> ( $T$ )	0.92	-	<i>Agent Smith</i>

Table 2: A selection of existing decoupled acceptance conditions found in literature and current agent implementations.

### 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*, *Nozomi* [15], *Yushu* [1], *IAM(crazy)Haggler* [36], *FSEGA* [30], and *Agent Smith* [33]. We also examined well-known agents from literature, such as the *Trade-off* agent [7], the *Bayesian learning agent* [11], *ABMP* [14], *equilibrium strategies of* [10], and *time dependent negotiation strategies as defined in* [27], i.e. the *Boulware* and *Conceder* tactics.

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

Some agents also use logical combinations of different acceptance conditions

at the same time. This explains why some agents are listed multiple times in the table. For example, both *IAMHaggler* and *IAMcrazyHaggler* [36] 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* [14] between the parties is smaller than  $\beta$ . 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 an acceptance condition when time is running short. For example, many ANAC agents such as *Yushu*, *Nozomi* and *FSEGA* [1, 30, 15] 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 similar. 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. However, 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

$$H_{A \leftrightarrow B}^t = \left( x_{p_1 \rightarrow p_2}^{t_1}, x_{p_2 \rightarrow p_3}^{t_2}, \dots, x_{A \rightarrow B}^{t_{n-1}}, x_{B \rightarrow A}^{t_n} \right),$$

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

$$H_{B \rightarrow A}^W = \{x_{B \rightarrow A}^s \in H_{A \leftrightarrow B}^t \mid s \in W\}$$

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  $H = H_{B \rightarrow A}^W$ :

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

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  $\alpha$ ) given the remaining time:

$$\begin{aligned} & \mathbf{AC}_{\text{combi}}(T, \alpha) \\ & \quad \stackrel{\text{def}}{\iff} \\ & \mathbf{AC}_{\text{next}} \vee \mathbf{AC}_{\text{time}}(T) \wedge (U_A(x_{B \rightarrow A}^t) \geq \alpha). \end{aligned} \tag{8}$$

Note that equation (8) defines  $\mathbf{AC}_{\text{combi}}(T, \alpha)$  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.  $\mathbf{AC}_{\text{combi}}(T, \text{MAX}^T)$ : 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 the 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. This average utility score is then an indication of the efficacy of  $\mathbf{AC}_A$ .

For our experimental setup we employed GENIUS (General Environment for Negotiation with Intelligent multi-purpose Usage Simulation) [21]. 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) [5]. 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 [34]. The seven agents that participated in ANAC 2010 have been implemented by various international research groups of negotiation experts. We used these strategies in our experiments as they are representative of the state-of-the-art in automated negotiation at the time of writing. Firstly, we removed the built-in acceptance mechanism from this representative group of agents; this left us with its pure bidding tactics. As outlined in our negotiation model, this procedure allowed 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 [12]. Acceptance conditions have to be assessed on negotiation domains of different size and complexity. Negotiation results also depend on the *opposition* of the parties' preferences. The notion of weak and strong opposition can be formally defined [16]. 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 a good spread of negotiation characteristics by selecting four different negotiation scenarios with two preference profiles each (see Table 3 and Figure 1). We picked two domains from the three that were used in ANAC 2010 (cf. [4, 5]). We have also taken two negotiation scenarios from the ANAC 2011 competition [3] to include both a smaller and a larger domain to our experimental setup.

Some agents participating in ANAC 2010 did not scale well and could not deal with very large bid spaces; therefore, we omitted the even larger domains that featured in ANAC 2010 and 2011, as the agents had too many difficulties with them to make them reliable testing domains. Additionally, in contrast to the 2010 competition, ANAC 2011 introduced discount factors for some of the scenarios. We removed these discount factors to ensure compatibility with the ANAC 2010 agents.

Our smallest scenario is called *Laptop*. In this scenario, a seller and a buyer are negotiating the specifications of a laptop. An agreement in the

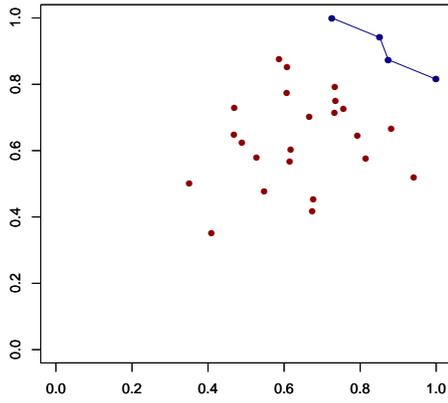
	Laptop	Itex–Cyp	Zim–Eng	Grocery
<b>Size</b>	27	180	576	1600
<b>Opposition</b>	Weak	Strong	Medium	Medium
<b>Mean utility</b>	0.67	0.48	0.58	0.44
<b>Nash Point</b>	(1.00, 0.82)	(0.72, 0.67)	(0.91, 0.73)	(0.84, 0.90)
<b>K-S Point</b>	(0.87, 0.87)	(0.72, 0.67)	(0.82, 0.79)	(0.84, 0.90)

Table 3: The eight preference profiles used in the experiments, as used in ANAC 2010 [5] and ANAC 2011 [3]. The rows indicate respectively: the size of the outcome space, the level of opposition, the arithmetic mean utility that can be obtained in the scenario, and the location of the Nash point and Kalai Smorodinsky point.

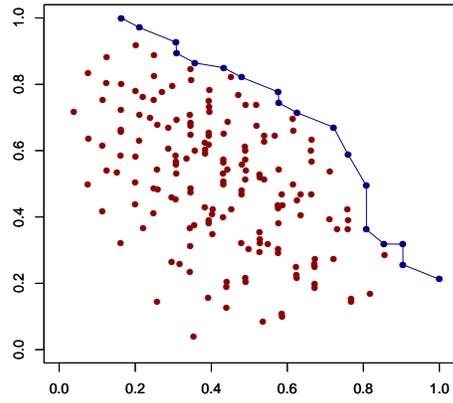
negotiation reconciles their differences and results in a purchase. The scenario has three issues: the laptop brand, the size of the hard disk, and the size of the external monitor. Each issue has only three options, making it a very small scenario with only 27 possible outcomes. Unbeknownst to each other, the buyer and seller actually both prefer to buy (and sell, respectively) a laptop with a small screen. The buyer prefers this because it is cheaper, and the seller prefers to sell laptops with small screens because s/he has more of those in stock. If the two parties are able to find the outcomes that are mutually beneficial to both, then they are happy to do business together with high utility scores on both sides. This can be confirmed in Table 3: the scenario has the highest arithmetic mean utility, and the most favorable Nash and Kalai-Smorodinsky point.

Our second scenario is taken from [17], 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. Even the Nash point utilities are quite low for both parties. Altogether, there are 180 potential offers that contain all combinations of values for the four issues.

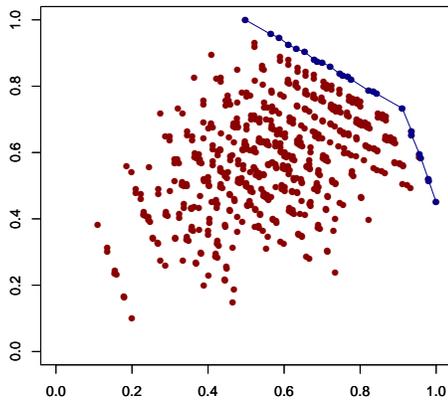
Third, the domain taken from [20, 22] involves a case where *England and Zimbabwe* negotiate an agreement on tobacco control. The leaders of both countries must reach an agreement on five issues. England and Zimbabwe



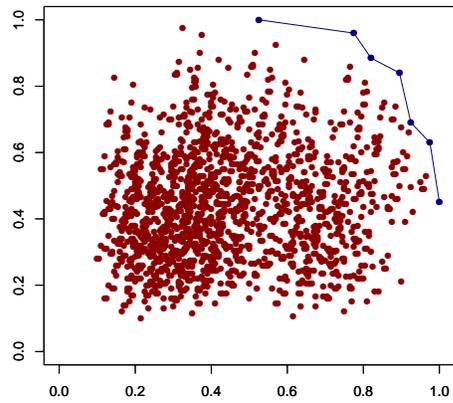
(a) Laptop



(b) Itex-Cypress



(c) England-Zimbabwe



(d) Grocery

Figure 1: The Pareto frontier of the outcome space of the four scenarios used in the experiments.

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.

Our final negotiation case concerns the *Grocery* scenario, which models a shopping negotiation in a local supermarket. The negotiation is between two persons having different tastes, who wish to buy groceries together. The discussion is about five product categories: bread, fruit, snacks, spreads, and vegetables. Each category consists of four to five possible options, resulting in a scenario with 1600 possible outcomes. Apart from their differences in taste, the two parties also differ in what category of product they find more important. The preferences are modeled in such a way that a good outcome is achievable for both, so the Nash and Kalai-Smorodinsky point utilities are high for both parties; however, the outcome space is scattered (resulting in a relatively low mean utility), so agents must explore it considerably to find the jointly profitable ones.

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 8 different preference profiles. We ran every experiment a total of  $N = 15$  times, so that altogether each acceptance condition was tested  $21 \times 8 \times 15 = 2520$  times in total. This resulted in running as many negotiations, and as every negotiation lasts 3 minutes, the experiments took 126 hours of cpu time. We selected a wide range of 102 acceptance conditions for experimental testing, as shown in Table 4. The different values of parameters will be discussed in the section below.

Additionally, we ran five more experiments with agents having their original, built-in acceptance mechanism in place. That is, we also tested the original agents' *coupled* acceptance mechanism for comparison purposes. 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.

To evaluate the hypotheses below, we have carried out a large number of experiments. A small selection of the results is summarized in Table 5.

<b>AC</b>	<b>Ranges</b>	<b>Increments</b>
$\mathbf{AC}_{\text{prev}}(\alpha, \beta)$ , and $\mathbf{AC}_{\text{next}}(\alpha, \beta)$	$\alpha \in [1, 1.05)$ , and $\beta \in [0, 0.1)$	for $\alpha$ : 0.01, and for $\beta$ : 0.02
$\mathbf{AC}_{\text{const}}(\alpha)$	$\alpha \in [0, 1)$	0.05 increments
$\mathbf{AC}_{\text{time}}(T)$	$T \in [0, 1)$	0.05 increments
$\mathbf{AC}_{\text{combi}}(T, \text{MAX}^W)$ , and $\mathbf{AC}_{\text{combi}}(T, \text{AVG}^W)$	$T \in [0.95, 1)$	0.01 increments
$\mathbf{AC}_{\text{combi}}(T, \text{MAX}^T)$	$T = 0.99$	-

Table 4: The selected ranges and increments for the parameters of different acceptance conditions in the experimental setup.

The table shows the average utility obtained by the agents, and the standard deviation (of the  $N = 15$  experiments), 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 reaching the deadline), they get zero utility. Thus, as a general observation, the following holds:

*(The acceptance dilemma)*

$$\text{Total average utility} = \text{Agreement percentage} \times \text{Average utility of agreements.}$$

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.

Our first hypothesis is about the simplest condition,  $\mathbf{AC}_{\text{const}}(\alpha)$ , and reads as follows:

**Hypothesis 1.** There is no single choice for  $\alpha$  that makes  $\mathbf{AC}_{\text{const}}(\alpha)$  an effective acceptance condition; this is mainly because the optimal choice of  $\alpha$  is very domain-dependent.

AC	$\alpha$	$\beta$	Util	SD	Agt %	Agt Util
$\mathbf{AC}_{\text{prev}}(\alpha, \beta)$	1	0	0.680	0.0084	80%	0.851
	1	0.04	0.711	0.0094	84%	0.842
	1	0.08	0.722	0.0076	87%	0.827
	1.02	0.04	<b>0.723</b>	0.0085	86%	0.837
$\mathbf{AC}_{\text{next}}(\alpha, \beta)$	1	0	0.683	0.0112	81%	0.843
	1	0.04	0.727	0.0067	87%	0.833
	1	0.08	0.731	0.0057	89%	0.819
	1.02	0.04	<b>0.737</b>	0.0060	89%	0.830
$\mathbf{AC}_{\text{const}}(\alpha)$	0.20	-	0.492	0.0025	100%	0.492
	0.55	-	<b>0.619</b>	0.0027	92%	0.671
	0.80	-	0.501	0.0078	60%	0.842
	0.90	-	0.343	0.0080	36%	0.952
Built-in mechanism	-	-	<b>0.737</b>	0.0057	89%	0.774
<b>T</b>						
$\mathbf{AC}_{\text{time}}(T)$	0.10	-	0.533	0.0035	100%	0.533
	0.40	-	0.548	0.0064	100%	0.548
	0.70	-	0.602	0.0062	100%	0.602
	0.95	-	<b>0.648</b>	0.0063	100%	0.648
$\mathbf{AC}_{\text{combi}}(T, \text{MAX}^W)$	0.97	-	0.756	0.0019	100%	0.756
	0.98	-	<b>0.762</b>	0.0031	100%	0.764
	0.99	-	0.761	0.0046	98%	0.776
$\mathbf{AC}_{\text{combi}}(T, \text{AVG}^W)$	0.97	-	0.739	0.0050	100%	0.739
	0.98	-	0.754	0.0037	100%	0.757
	0.99	-	<b>0.759</b>	0.0056	98%	0.774
$\mathbf{AC}_{\text{combi}}(T, \text{MAX}^T)$	0.99	-	<b>0.737</b>	0.0083	93%	0.796

Table 5: A small selection of the various acceptance conditions that were tested, together with average utility obtained and standard deviation. The utility of the best scoring AC of each category is in **bold**. The two right-hand side columns show agreement percentages and the utility obtained when an agreement is reached.

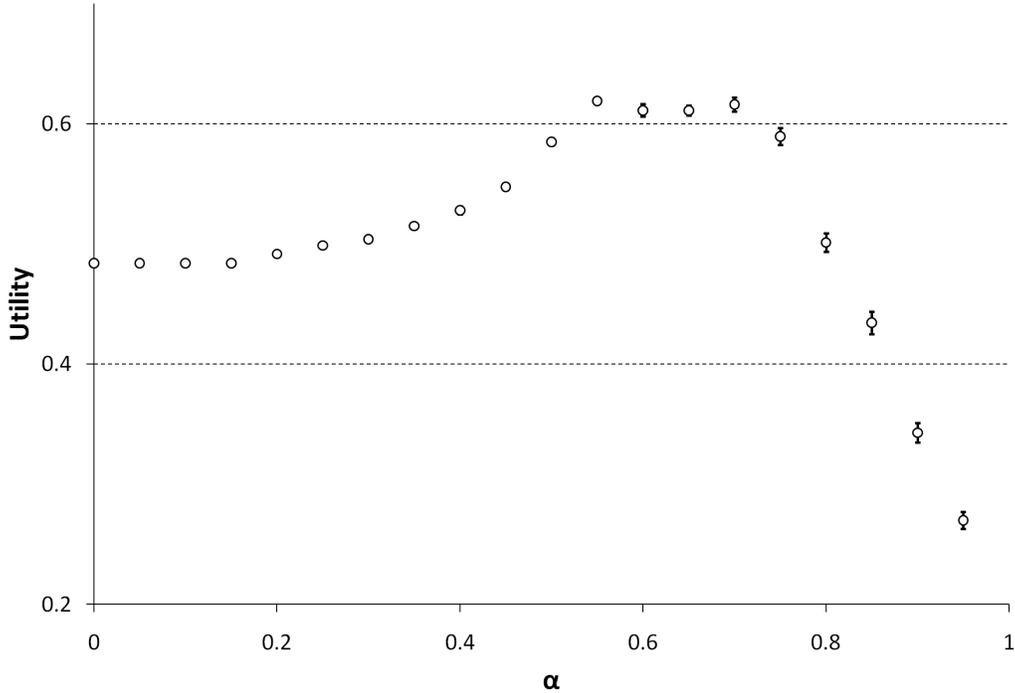


Figure 2: The average utility obtained by agents using  $\mathbf{AC}_{\text{const}}(\alpha)$ . The vertical error bars indicate one standard deviation to the mean.

First, consider  $\mathbf{AC}_{\text{const}}(0.9)$  and  $\mathbf{AC}_{\text{const}}(0.8)$  by consulting Table 5. When they reach an agreement, they receive a very high utility (at least 0.9 or 0.8 respectively), but this happens so infrequently (resp. 60% and 36% of all negotiations), that they are ranked at the bottom when we consider total average utility. On the other hand, choosing a low value for  $\alpha$ , such as using  $\mathbf{AC}_{\text{const}}(0.2)$ , will always result in an immediate agreement, but with one of the lowest possible scores of 0.492.

The best possible choice for  $\alpha$  should therefore be somewhere in the middle between zero and one, and is found to be 0.55 (see Figure 2), yielding a payoff of 0.619. Firstly, this is still a suboptimal outcome compared to other AC's, such as the  $\mathbf{AC}_{\text{next}}$  and  $\mathbf{AC}_{\text{combi}}$  variants.

Moreover, it is worth noting that this optimal value may be best on average, but in this case, averaging over all scenarios also hides a lot of information. When we break down our analysis and look at the four domains

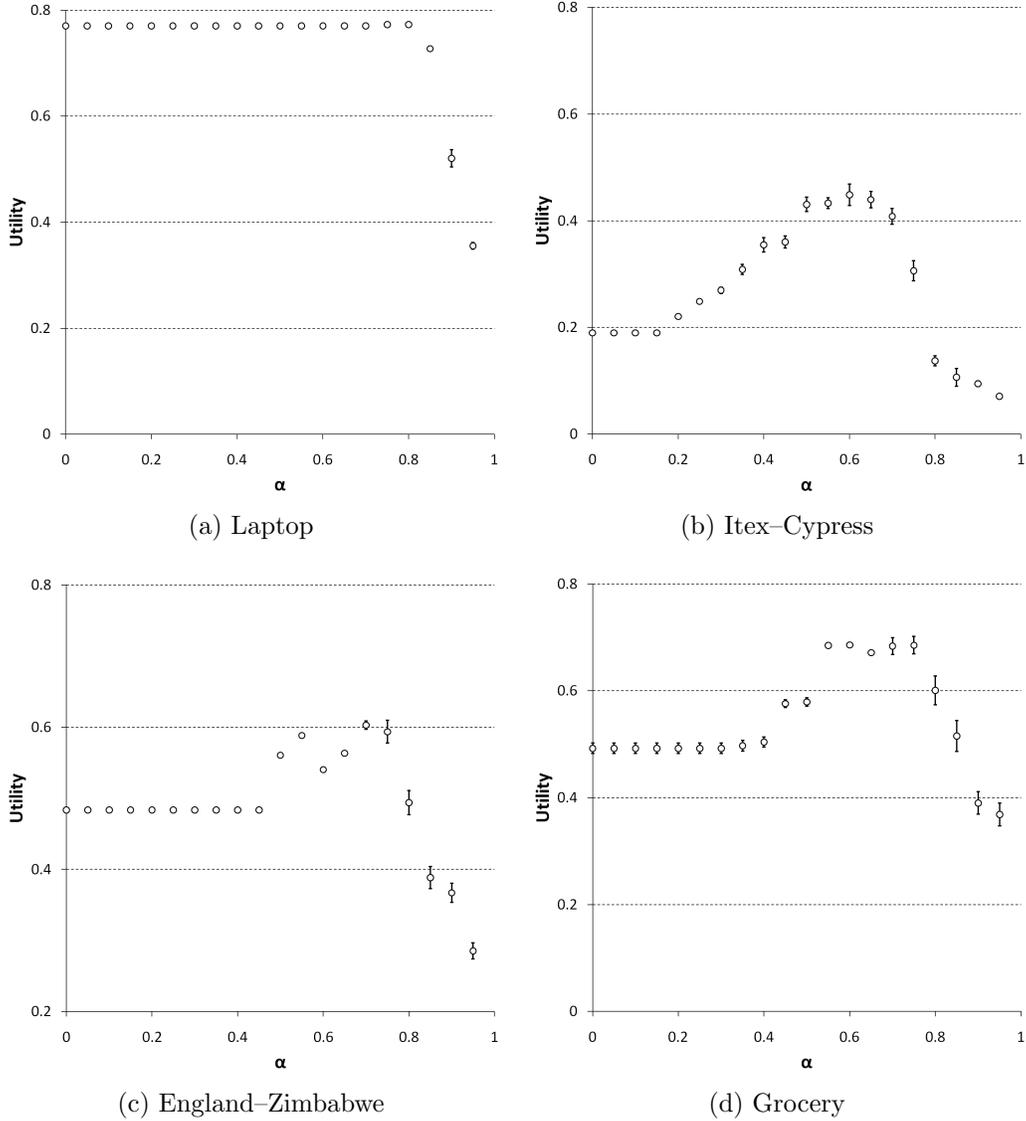


Figure 3: The average utility of  $\mathbf{AC}_{\text{const}}(\alpha)$  per negotiation scenario, for  $\alpha \in [0, 1)$ .

separately (see the four figures of Figure 3), we see that the optimal range of  $\alpha$  differs greatly per domain. For example, on *Itex vs. Cypress*, the optimal choice for  $\alpha$  is around 0.6, while on *Grocery*, the best performing value is in the range of  $[0.7, 0.8]$ . On the *Laptop* domain, any choice for  $\alpha \in [0, 0.8]$  is the best  $\mathbf{AC}_{\text{const}}(\alpha)$  can do in this scenario, and will cause the agent to

instantly accept most offers.

We 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  $\alpha$  is highly domain-dependent. A very cooperative scenario may have multiple win–win outcomes with utilities above  $\alpha$ .  $\mathbf{AC}_{\text{const}}(\alpha)$  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  $\mathbf{AC}_{\text{prev}}(\alpha, \beta)$  and  $\mathbf{AC}_{\text{next}}(\alpha, \beta)$  are standard in literature for  $\alpha \in [1, 1.03]$  and  $\beta \in [0, 0.2]$ . Many agents tend to use these acceptance conditions, as they are well-known and easy to implement. We have formed the following hypothesis about them:

**Hypothesis 2.**  $\mathbf{AC}_{\text{next}}(\alpha, \beta)$  will outperform  $\mathbf{AC}_{\text{prev}}(\alpha, \beta)$  for all  $\alpha$  and  $\beta$ . However, both conditions will perform worse than combined acceptance conditions, which also take the remaining time into account.

To test this hypothesis, we considered many different values for  $\alpha$  and  $\beta$  in our experiments, with ranges chosen around the values we had found in existing agents (cf. Table 2).

Consulting Table 5, the first observation is that  $\mathbf{AC}_{\text{prev}}(\alpha, \beta)$  as well as  $\mathbf{AC}_{\text{next}}(\alpha, \beta)$  already perform much better than  $\mathbf{AC}_{\text{const}}$  for all tested values of  $\alpha$  and  $\beta$ . Higher values for  $\alpha$  and  $\beta$  generally yield a better result, although the differences are quite small. However, given that we average the utility over 15 runs, we are able to statistically distinguish the performance for different values of  $\alpha$  and  $\beta$ . We have found  $\mathbf{AC}_{\text{next}}(\alpha, \beta)$  does indeed outperform  $\mathbf{AC}_{\text{prev}}(\alpha, \beta)$  for all tested values of  $\alpha$  and  $\beta$ , except for  $\beta = 0$  (*two-tailed t-test*,  $p < 0.01$ ), thereby partially confirming the hypothesis.

As an example, we have plotted  $\mathbf{AC}_{\text{next}}(1, \beta) = \mathbf{AC}_{\text{gap}}(\beta)$  and  $\mathbf{AC}_{\text{prev}}(1, \beta)$  for  $\beta \in [0, 1)$  in Figure 4. We can confirm that  $\mathbf{AC}_{\text{next}}(1, \beta)$  obtains scores that are significantly higher (using  $p < 0.01$ ) scores than  $\mathbf{AC}_{\text{prev}}(1, \beta)$ , for  $\beta \neq 0$ .

It makes sense that comparing the opponent’s offer to our upcoming offer is more beneficial than comparing it to our previous offer, as  $\mathbf{AC}_{\text{next}}$  is always

‘one step ahead’ of  $\mathbf{AC}_{\text{prev}}$ . In general,  $\mathbf{AC}_{\text{next}}$  is never worse than  $\mathbf{AC}_{\text{prev}}$ , and therefore there seems no reason to use the latter.

One of the top choices for both  $\mathbf{AC}_{\text{next}}$  and  $\mathbf{AC}_{\text{prev}}$ , is setting  $\alpha = 1.02$ , and  $\beta = 0.04$  (interestingly, *IAM(crazy)haggler* makes the same choice for  $\alpha$ , cf. Table 2). However, even for this choice, the combined acceptance conditions  $\mathbf{AC}_{\text{combi}}(T, \text{MAX}^W)$  outperform both of them for all tested values of  $T$  (*two-tailed t-test*,  $p < 0.01$ ). This also settles the second part of the hypothesis.

The reason for the relatively bad performance of  $\mathbf{AC}_{\text{next}}$  and  $\mathbf{AC}_{\text{prev}}$  is that many bidding strategies focus on the ‘negotiation dance’ [26]. 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 these 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.**  $\mathbf{AC}_{\text{time}}(T)$  always reaches an agreement, but of relatively low utility. This utility improves when  $T$  gets closer to the deadline.

To evaluate this hypothesis we tested  $\mathbf{AC}_{\text{time}}(T)$  for many possible values of  $T \in [0, 1)$ , a selection of which can be examined in Table 5. We have found that the obtained utility increases monotonously with larger  $T$ , i.e.: it is optimal to choose the value of  $T$  sufficiently close to the deadline, while still allowing enough time to reach a win-win agreement. The fact that one has to accept as late as possible when using  $\mathbf{AC}_{\text{time}}(T)$  clearly stems from the fact that we are dealing with undiscounted domains only; see Section 7 for a discussion on possible extensions in this regard.

From observing the acceptance probability of  $\mathbf{AC}_{\text{time}}(T)$  in the experimental results, we see that the agent will always reach an agreement, therefore we consider this part of the hypothesis confirmed.

Regarding the utility of the agreement,  $\mathbf{AC}_{\text{time}}(T)$  with  $T < 1$  is a sensible criterion to avoid a break off at all costs. It is rational to prefer any outcome over a break off of zero utility. However, the resulting deal can

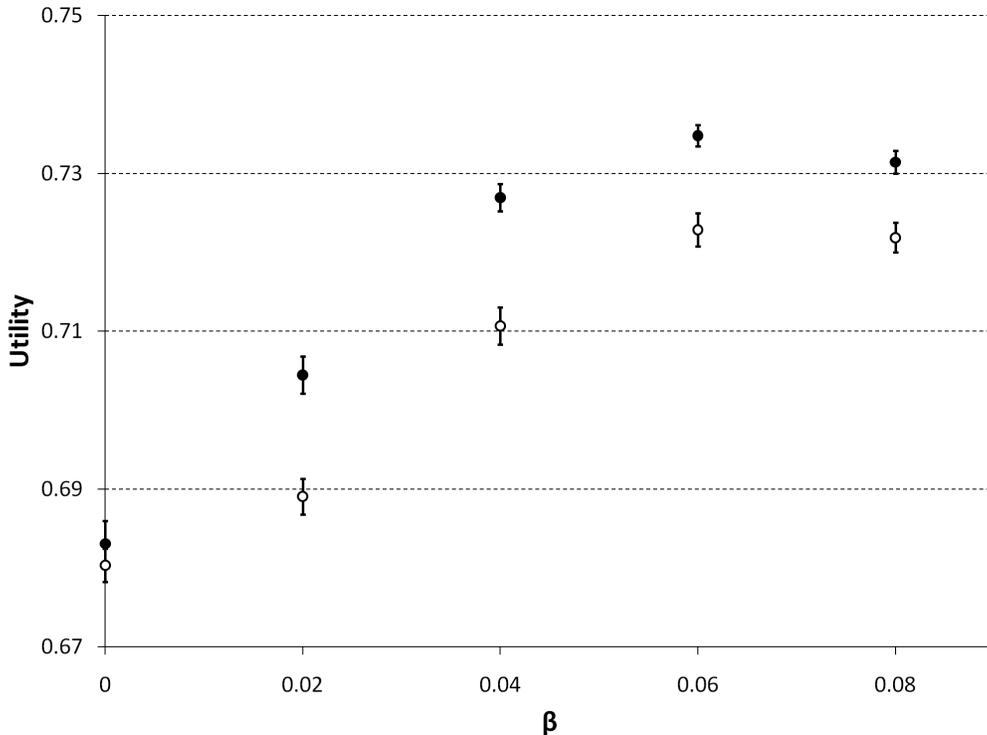


Figure 4: The average utility obtained by agents using  $\mathbf{AC}_{\text{next}}(1, \beta)$  (in black), and  $\mathbf{AC}_{\text{prev}}(1, \beta)$  (in white). The vertical error bars indicate one standard deviation to the average utility of  $N = 15$  different runs.

be anything. As we can see from the table, this is the reverse situation of  $\mathbf{AC}_{\text{const}}(0.9)$ :  $\mathbf{AC}_{\text{const}}(0.9)$  rarely gets a deal, but when it does, it is of high utility. Conversely,  $\mathbf{AC}_{\text{time}}(T)$  yields a low agreement score (0.648 for  $T = 0.95$ ), but with certainty of agreement. The overall score is the same (0.648), but it is interesting to note that this score is worse than all scores by either  $\mathbf{AC}_{\text{next}}$  or  $\mathbf{AC}_{\text{prev}}$  (*two-tailed t-test*,  $p < 0.01$ ). This phenomenon can again be explained by the acceptance dilemma: by accepting any offer near the deadline, it reaches more agreements, but of relatively low utility.

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 $\mathbf{AC}_{\text{combi}}(T, \alpha)$

When evaluating  $\mathbf{AC}_{\text{combi}}(T, \alpha)$ , we expected the following characteristics. First,  $\mathbf{AC}_{\text{combi}}(T, \alpha)$  is an extension of  $\mathbf{AC}_{\text{next}}$  in the sense that it will accept under broader circumstances. It alleviates some of the mentioned drawbacks of  $\mathbf{AC}_{\text{next}}$  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  $x_{A \rightarrow B}^{t_{n-1}}$ , my next bid  $x_{A \rightarrow B}^t$ , the remaining time, and the opponent’s bid  $x_{B \rightarrow A}^t$ , this condition employs the entire bidding history  $H_{A \leftrightarrow B}^t$  to compute the acceptability of an offer. Therefore we expect better results than with  $\mathbf{AC}_{\text{next}}$ , with more agreements, and when it agrees, we expect a better deal than by using  $\mathbf{AC}_{\text{time}}(T)$ .

We capture this last statement in our final hypothesis:

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

As is evident from the experimental results, there are two acceptance conditions that dominate the others, namely  $\mathbf{AC}_{\text{combi}}(T, \text{MAX}^W)$ , as well as  $\mathbf{AC}_{\text{combi}}(T, \text{AVG}^W)$  with  $T$  close to the deadline. The results are not statistically different for the different values of  $T$ , but any of the tested values performs quite well. One of the best AC’s of the test is  $\mathbf{AC}_{\text{combi}}(0.98, \text{MAX}^W)$  with a score of 0.762, which is even better than the built-in mechanisms of the agents, and also surpasses the performance of  $\mathbf{AC}_{\text{next}}(\alpha, \beta)$  for any  $\alpha$  and  $\beta$  (significantly so, using a *two-tailed t-test*,  $p < 0.01$ ). In particular, it is at least 12% better than  $\mathbf{AC}_{\text{next}}$  (*two-tailed t-test*,  $p < 0.01$ ).

Similar to  $\mathbf{AC}_{\text{time}}$ , the combined conditions still get a deal almost every time, but with a higher payoff. However, the average utility of an agreement is not the highest: the built-in mechanisms and several  $\mathbf{AC}_{\text{const}}(\alpha)$  conditions 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. This settles our last hypothesis.

Finally, aiming for the highest utility that has been offered so far (i.e., using  $\mathbf{AC}_{\text{combi}}(T, \text{MAX}^T)$ ) is not as successful, 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.

## 5. 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 [27] (e.g. the Boulware and Conceder tactics). The same principle is used in the equilibrium strategies of [10] and for the Trade-off agent [7], although this concerns a setting where the deadline can be different for both agents. In our work, we consider strategies that do not always reach an agreement, and we have concentrated on acceptance conditions that yield better results in such cases.

Of all ANAC 2010 participants, we shortly discuss *Agent K* [15] 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 5) when equipped with our combined acceptance conditions than with its built-in mechanism.

This work builds upon earlier research [4], which also experimentally tested various acceptance conditions, albeit in a more limited setting. In this paper, we extend results in [4] and gain additional insights by exploring a larger class of acceptance conditions in a wider range of negotiation scenarios. Although we do not focus on negotiation tactics and convergence results, our negotiation model also builds upon the model of [31]. 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.

## 6. Conclusion

In this paper, we aimed to classify current approaches to generic acceptance conditions and to compare a selection of acceptance conditions in a

real-time 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), for which we have defined a formal negotiation model.

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 presented and classified many of the standard acceptance criteria that are currently used by negotiating agents, including  $\mathbf{AC}_{\text{next}}$ ,  $\mathbf{AC}_{\text{prev}}$ , and  $\mathbf{AC}_{\text{const}}$ . From our results, it is apparent that they do not always yield optimal agreements, and we established that they perform 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. In particular, they outperform the acceptance mechanisms that are used by the top ANAC 2010 agents.

## 7. Discussion and Future Work

We have examined the effectivity of acceptance conditions in a setting with two key elements: a bilateral alternating offers protocol, and a real-time deadline. We briefly discuss our results and possible lines of future work in light of different negotiation contexts.

The adoption of the alternating offers protocol imposes an important restriction on the negotiation process, because the agents only exchange information in one of three possible forms: an offer, an accept, or a withdrawal. Normally it is irrational to withdraw from a negotiation (i.e., by sending a message ending the negotiation) without any outside options, as it leaves the agent with nothing. However, recently there has been interest [3] in real-time settings with reservation values and discount factors. When both contract utility and outside options devaluate with the passing of time, novel acceptance conditions are required that give more consideration to the negotiation

timeline. For example, it can be advantageous for an agent to end the negotiation prematurely and receive its reservation value, rather than continuing an exchange of offers while the contract diminishes in value. This adds an additional dimension to the acceptance dilemma, as prolonging the negotiation does not necessarily increase the agent’s chances of a good outcome.

In a multi-party setting, the problem of when to accept is even more complex, as the outside options become dynamic; however, the presence of a mediator can reduce some of the complexity by taking over the role of finding acceptable agreements, for example through letting the agents vote on whether a proposed contract is acceptable [18]. It may then be sufficient for an agent to simply accept anything above its reservation value. In the same way, when richer protocols are employed (e.g., when communication is possible, for instance in persuasive, or argumentation-based negotiation [25, 32]), the acceptance dilemma may be easier to resolve, as agents have more knowledge about the acceptability of offers. Lastly, in traditional negotiation protocols such as ours, once a contract is settled upon, it is binding. However, a more general approach is to allow decommitment, i.e. backing out of the negotiation after finding a superior option elsewhere, usually at the cost of a penalty [29]. This requires complex acceptance strategies for committing and decommitting to agreements in a concurrent way; there has been recent work in the same negotiation setting that we employ, which opens up possible research in this area [35].

Finally, the real-time setting presents an additional challenge to finding effective acceptance mechanisms. For example, in a round-based setting, results are usually less ambiguous, as this usually concerns bargaining games with perfect information where a unique subgame-perfect equilibrium exists. An optimal acceptance strategy can then be adopted through backward inductive reasoning; the most well-known solution being that agreement is reached immediately in the first round [28]. In a real-time setting, it is generally unknown when the last offer has been made, and this makes it difficult to find optimal acceptance conditions for this setting; this is why our approach is heuristic in essence.

For future work, we plan to test acceptance conditions in more dynamic settings with more agents and on more complex scenarios, using the resources of ANAC 2011 and 2012.

Secondly, 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. We plan to examine such extensions in future work.

## Acknowledgments

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.

## 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*, pages 145–149, Berlin, Heidelberg, 2012. Springer-Verlag.
- [2] Robert J. Aumann and S. Hart, editors. *Handbook of Game Theory with Economic Applications*, volume 1. Elsevier, 1 edition, 1992.
- [3] Tim Baarslag, Katsuhide Fujita, Enrico H. Gerding, Koen Hindriks, Takayuki Ito, Nicholas R. Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, Valentin Robu, and Colin R. Williams. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, 198(0):73 – 103, 2013.
- [4] Tim Baarslag, Koen Hindriks, and Catholijn Jonker. Acceptance conditions in automated negotiation. In Takayuki Ito, Minjie Zhang, Valentin Robu, and Tokuro Matsuo, editors, *Complex Automated Negotiations: Theories, Models, and Software Competitions*, volume 435 of *Studies in Computational Intelligence*, pages 95–111. Springer Berlin / Heidelberg, 2013.

- [5] 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*, pages 113–135, Berlin, Heidelberg, 2012. Springer-Verlag.
- [6] Chi-Bin Cheng, Chu-Chai Henry Chan, and Kun-Cheng Lin. Intelligent agents for e-marketplace: Negotiation with issue trade-offs by fuzzy inference systems. *Decision Support Systems*, 42(2):626 – 638, 2006.
- [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] P. Faratin, C. Sierra, and N.R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.
- [9] S. Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. Optimal negotiation strategies for agents with incomplete information. In *Revised Papers from the 8th International Workshop on Intelligent Agents VIII, ATAL '01*, pages 377–392, London, UK, UK, 2002. Springer-Verlag.
- [10] 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.
- [11] Koen V. Hindriks and Dmytro Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 1, AAMAS '08*, pages 331–338, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.
- [12] Koen V. Hindriks and Dmytro Tykhonov. Towards a quality assessment method for learning preference profiles in negotiation. In Wolfgang Ketter, Han Poutr, Norman Sadeh, Onn Shehory, and William Walsh, editors, *Agent-Mediated Electronic Commerce and Trading Agent Design*

- and Analysis*, volume 44 of *Lecture Notes in Business Information Processing*, pages 46–59. Springer Berlin Heidelberg, 2010.
- [13] Takayuki Ito, Hiromitsu Hattori, and Mark Klein. Multi-issue negotiation protocol for agents: exploring nonlinear utility spaces. In *Proceedings of the 20th international joint conference on Artificial intelligence, IJCAI'07*, pages 1347–1352, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
  - [14] Catholijn Jonker, Valentin Robu, and Jan Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and Multi-Agent Systems*, 15:221–252, 2007.
  - [15] Shogo Kawaguchi, Katsuhide Fujita, 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*, pages 137–144, Berlin, Heidelberg, 2012. Springer-Verlag.
  - [16] G. E. Kersten and S. J. Noronha. Rational agents, contract curves, and inefficient compromises. *Trans. Sys. Man Cyber. Part A*, 28(3):326–338, May 1998.
  - [17] 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*, 11(3):297–316, 2003.
  - [18] Mark Klein, Peyman Faratin, Hiroki Sayama, and Yaneer Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation*, 12:111–125, 2003. 10.1023/A:1023068821218.
  - [19] Sarit Kraus. *Strategic Negotiation in Multiagent Environments*. MIT Press, October 2001.
  - [20] 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)*, volume 319, page 69–87. Springer, Springer, 2011.

- [21] Raz Lin, Sarit Kraus, Tim Baarslag, Dmytro Tykhonov, Koen Hindriks, and Catholijn M. Jonker. Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 2012.
- [22] 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.
- [23] 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.
- [24] Martin J. Osborne and Ariel Rubinstein. *Bargaining and Markets (Economic Theory, Econometrics, and Mathematical Economics)*. Academic Press, April 1990.
- [25] Iyad Rahwan, Sarvapali Ramchurn, Nicholas R. Jennings, Peter McBurney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(04):343–375, 2003.
- [26] H. Raiffa. *The Art and Science of Negotiation*. Harvard University Press, 1982.
- [27] Raquel Ros and Carles Sierra. A negotiation meta strategy combining trade-off and concession moves. *Autonomous Agents and Multi-Agent Systems*, 12:163–181, 2006.
- [28] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982.
- [29] Tuomas Sandholm and Victor R. Lesser. Advantages of a leveled commitment contracting protocol. In William J. Clancey and Daniel S. Weld, editors, *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, August 4-8, 1996, Volume 1*, pages 126–133. AAAI Press / The MIT Press, 1996.

- [30] 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*, pages 159–165, Berlin, Heidelberg, 2012. Springer-Verlag.
- [31] 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, MAAMAW97*, volume 1237 of *Lecture Notes in Artificial Intelligence*, pages 17–35. Springer-Verlag, 1997.
- [32] Katia P. Sycara. Machine learning for intelligent support of conflict resolution. *Decision Support Systems*, 10(2):121 – 136, 1993.
- [33] Niels van Galen Last. Agent smith: 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*, pages 167–174, Berlin, Heidelberg, 2012. Springer-Verlag.
- [34] 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.
- [35] Colin R. Williams, Valentin Robu, Enrico Gerding, and Nick Jennings. Towards a platform for concurrent negotiations in complex domain. In *Proceedings of The Fifth International Workshop on Agent-based Complex Automated Negotiations (ACAN 2012)*, In Press.
- [36] Colin R. Williams, Valentin Robu, Enrico H. Gerding, 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*, pages 151–158, Berlin, Heidelberg, 2012. Springer-Verlag.