Using Gaussian Processes to Optimise Concession in Complex Negotiations against Unknown Opponents

Colin R. Williams, Valentin Robu, Enrico H. Gerding and Nicholas R. Jennings
School of Electronics and Computer Science,
University of Southampton,
Southampton, SO17 1BJ, UK
{crw104,vr2,eg,nrj}@ecs.soton.ac.uk

Abstract

In multi-issue automated negotiation against unknown opponents, a key part of effective negotiation is the choice of concession strategy. In this paper, we develop a principled concession strategy, based on Gaussian processes predicting the opponent's future behaviour. We then use this to set the agent's concession rate dynamically during a single negotiation session. We analyse the performance of our strategy and show that it outperforms the state-of-the-art negotiating agents from the 2010 Automated Negotiating Agents Competition, in both a tournament setting and in self-play, across a variety of negotiation domains.

1 Introduction

A key aspect to effective negotiation in multi-issue negotiations is the choice of concession strategy. Specifically, there exists a trade-off between conceding quickly, thereby reaching an agreement early with a low utility, versus a more patient approach, that concedes slowly, therefore reaching an agreement later, but potentially with a higher utility. This trade-off is particularly relevant in domains with a discounting factor, where unnecessary delays result in a lower utility. Furthermore, where the negotiation has a deadline, there is a risk that delaying concession may result in no agreement being reached. Against this background, our aim is to develop a concession strategy for use in negotiations against unknown opponents. Such opponents have neither a known utility function, nor a known type of behaviour. Furthermore, the strategy will be used in domains which consist of multiple issues and realistic time constraints. That is, rather than having fixed rounds, we assume that agents can make offers in real time. Furthermore, the time constraints we consider are based on the amount of real time that has elapsed, and consist of both a deadline and discounting. All of these features combine to create a complex but realistic negotiation setting.

As an example, consider a scenario in which two parties are negotiating over the provision of a computing resource in order to compute the solution to a problem. The issues that they are negotiating over include the price per unit, the processor speed and the amount of memory. The user and the service provider have not previously done business together, and therefore there are no previous negotiation encounters that can be used by either party to inform their negotiation strategy. Furthermore, both parties need to reach an agreement before a fixed deadline. There is a benefit to reaching an agreement promptly, as this will allow the task to be completed sooner, freeing up the service provider's resources to be used by another user, whilst providing the user with a result in shorter time.

There exists extensive literature that deals with different aspects of such a scenario. Specifically, [Faratin *et al.*, 1998] developed time dependent concession, in which the utility level is calculated as a function of time. This approach uses a parameter that affects the rate of concession, and it is necessary for the agent to set this parameter appropriately in advance. However, it is not possible to tune these parameters in advance without any knowledge of the opponent's behaviour. In addition, they propose behaviour dependent concession, such as the tit-for-tat approach, in which the agent chooses its concession based on the ground given by its opponent in previous rounds. However, this assumes that the concession made by the opponent can be observed. In negotiations where the opponent's utility function is unknown, this may not be the case, and it may be necessary to measure the concession in terms of the agent's own utility function. In this case, a tit-for-tat strategy may concede more than it needs to, resulting in a lower utility for the agent. To address the shortcomings of these strategies, more recent literature specifically deals with uncertainty. To this end, [Brzostowski and Kowalczyk, 2006] propose a strategy that performs on-line prediction of opponent behaviour, by assuming that the opponent uses a strategy that is a weighted combination of time- and behaviour-dependent concession. However, their negotiation environment contains no discounting factor, and so their solution assumes that the best approach is to reach an agreement at the deadline. In negotiations where the opponent is unknown, many existing strategies attempt to model the opponent. Commonly, this involves learning either the opponent's preferences [Coehoorn and Jennings, 2004; Hindriks and Tykhonov, 2008; Robu *et al.*, 2005] or classifying the opponent [Lin *et al.*, 2008] using techniques such as Bayesian updating or kernel density estimation. However, in domains with many issues, this approach becomes computationally expensive, and may be unnecessary in automated negotiation where time constraints are based on real

time rather than the number of interactions. Under such constraints, there is no benefit to limiting the number of offers exchanged, hence it may be possible for automated agents to exchange thousands of offers in order to explore the negotiation space. We found this to be the case in experiments carried out with automated agents using the GENIUS negotiation environment [Hindriks *et al.*, 2009].

In addition to the above empirical work, there is considerable previous work that looks at computing game theoretic equilibria in agent-based negotiation. However, this tends to require additional assumptions and/or a less complex environment than the one we consider. For example, in [An *et al.*, 2010], there is one-sided uncertainty, in which one of the agents has uncertainty about its opponent's behaviour, whilst the other agent has complete information. In contrast, in our work there is two-sided uncertainty. Alternatively, in [Fatima *et al.*, 2007], the negotiation model assumes that time constraints are based on the number of offers made, rather than the real time used. In such an environment, it is possible to perform backwards induction in order to find a game theoretic solution, but this would be infeasible under real-time constraints, as the number of steps would be unknown.

Given this context, we have developed a principled approach, which first predicts the opponent's behaviour through a Gaussian process and then uses this to find the optimal concession. Optimal in this context means that our strategy maximises the agent's expected utility. Unlike previous work, our concession strategy considers the observed concession of the opponent and negotiation constraints which are based on the elapsed real time, rather than the number of negotiation rounds. In addition, our strategy can be adjusted to negotiate more aggressively by altering the risk profile of the agent. Furthermore, we make no assumptions about the opponent.

In developing this strategy, our work advances the state-of-the-art in multi-issue negotiation in the following ways:

- We provide the first concession strategy that uses an estimate of the opponent's future concession, including a measure of the uncertainty of that estimate, in order to optimally set the rate of concession, in a negotiation setting that contains multiple issues, uncertainty and real-time constraints.
- We show that this strategy outperforms other state-of-the-art strategies in a complex environment containing independently developed benchmark domains taken from the First Automated Negotiating Agents Competition (ANAC 2010) [Baarslag *et al.*, 2010].
- We show, using empirical game theory, that our strategy is stable and that other agents have an incentive to deviate towards it.

The remainder of this paper proceeds as follows. Section 2 describes the negotiation setting that we have considered, then Section 3 describes the strategy that we have designed for use in this setting. We evaluate the strategy in Section 4 and finally, we conclude in Section 5.

2 The Negotiation Setting

Our strategy is designed to participate in multi-issue, bilateral negotiation, in which two parties negotiate over multiple issues in order to reach an agreement. The negotiation protocol that is used is based on the alternating offers protocol [Rubinstein, 1982], with each offer, o, representing a complete package, in that it specifies the values for all issues. Formally, $o = (v_1, v_2, ..., v_n)$, where v_i is the value for issue i. Furthermore, the utility function, $U(\cdot)$, is defined as a function which maps all possible offers, o in the outcome space O, to a value. Specifically:

$$\forall o \in O, U(o) \in [0, 1] \tag{1}$$

Our agent knows its own utility function, but the utility function of the opponent is unknown. In each negotiation round, our agent receives an offer, o_{opp} from the opponent (except on the first round, if our agent starts the negotiation). It can choose to accept this offer or make a counter-offer, o_{own}.

Our model differs from that of [Rubinstein, 1982], in that we use a different model of time. Rather than having fixed rounds, agents can make offers in real time. This means that the number of negotiation steps that can be made in a given time period is unknown, and depends on the time required to compute the offer. Especially when using automated negotiation, this allows many offers to be exchanged in a short period of time. Furthermore, in order to encourage the agents to reach an agreement in a timely fashion, we apply a discounting factor, δ, which is based on the amount of real time that has elapsed, rather than the number of negotiation steps. Note that this value may be different for each agent and is privately known. We furthermore assume that there is a deadline, t_{max}, common to and known by both agents, which is a limit to the amount of time that the negotiation can last. If the deadline is reached without an agreement being made, the negotiation results in conflict and both agents receive zero utility. Typically, in our experimental setting, during a negotiation, agents can exchange thousands of offers. Specifically, the discounted utility function $D(\cdot, \cdot)$ is given by:

$$D(u, t) = ue^{-\delta t / t_{\text{max}}} \tag{2}$$

where u is the undiscounted utility of a particular agreement, δ is the discounting factor, t is the time of agreement and t_{max} is the deadline.

In a tournament setting, the primary aim of our strategy is to 'win' the negotiation by achieving a higher utility than that of its opponent. In such a setting, reaching an agreement with a fairly high utility is not always good enough, since it is possible that the opponent may have achieved an even higher utility, thereby winning the negotiation. Consequently, the strategy may need to take a more aggressive approach than it would if it were simply maximising its own utility.

In order to deal with this trade-off, we include the concept of risk attitude in the design of our concession strategy. Formally, the risk function used by our agent is:

$$R(u) = u^r \tag{3}$$

where r is the risk parameter. If $r = 1$, the strategy is risk-neutral, for $r > 1$, the strategy is risk-seeking and for $r < 1$, it is risk-averse. A risk-seeking strategy would result in more aggressive behaviour, since such an agent will concede more slowly, as it will regard lower utilities to be of even lower value than their true value.

Algorithm 1 Let δ be the discounting factor, r be the risk parameter, $t_{\max}$ be the negotiation deadline. o_{own} and o_{opp} represent our own and the opponent's latest offers, respectively. μ and σ are vectors, representing the mean and variance respectively, as output from the regression function. t^* is the time at which the estimation of our utility is maximised. u^* is the utility that we should offer at that time in order to maximise the received utility. u_τ is the target utility at time t_c.

```
Require: δ, r, t_max
  while t_c < t_max do
    o_opp ⇐ RECEIVEOFFER()
    RECORDOFFER(o_opp, t_c)
    if REGRESSIONREQUIRED(t_c) then
      μ, σ ⇐ PERFORMREGRESSION()
      t*, u* ⇐ GETBEST(μ, σ, t_c, δ, r)
    end if
    u_τ ⇐ GETTARGET(t*, u*, t_c)
    if GETUTILITY(o_opp, t_c, δ) ≥ u_τ then
      ACCEPTOFFER(o_opp)
      return
    end if
    o_own ⇐ GENERATEOFFER(u_τ)
    PROPOSEOFFER(o_own)
  end while
```

3 Design of the Negotiation Strategy

Our strategy for the above setting consists of three distinct phases. Algorithm 1 gives an overview of our approach and, in the remainder of this section, we describe these phases in turn. The first stage, described in Section 3.1, is to predict the concession of the opponent (represented by the PERFORMREGRESSION function). The second stage, described in Section 3.2, is to set the concession rate such that it optimises the expected utility given the prediction of the opponent's concession (represented by the GETBEST function). The final stage, described in Section 3.3 is to generate an offer according to the concession rate (represented by the GENERATEOFFER function).

3.1 Predicting the Opponent's Concession

To estimate the utility that we can expect to achieve later in the negotiation, our approach begins by predicting the concession of the opponent, but only using the information we can actually observe. This works as follows. First, for each offer made by the opponent, we record the time at which the offer was made, along with the utility that it offers to our agent, according to our agent's utility function. Then, we estimate the future concession of our opponent using a regression technique. Specifically, our agent uses a Gaussian process, with a Matérn covariance function and a linear mean function [Rasmussen and Williams, 2006]. We use a Gaussian process as it provides both a prediction and a measure of the level of confidence in that prediction. The covariance and mean functions were selected due to their robustness, as well as the fact that they can be computed in real time during the negotiation.

The output of the Gaussian process is a Gaussian distribution, for each time t, of the form:

$$f(u; \mu_t, \sigma_t) = \frac{1}{\sigma_t \sqrt{2\pi}} e^{\frac{-(u-\mu_t)^2}{2\sigma_t^2}} \tag{4}$$

where μ_t and σ_t are the mean and standard deviation, respectively. The mean, μ_t, gives an indication of the most likely value for u at time t, whilst the standard deviation, σ_t is an indication of how accurate the prediction of μ_t is likely to be.

We note that alternative regression techniques can be used in place of a Gaussian process, provided their output contains mean and confidence measures. However, in this work, we have only evaluated our approach using a Gaussian process.

As input to the Gaussian process, we use the maximum value offered by the opponent in a particular time window of duration t_{window}, and the time of that window. The reason for using this windowed approach is twofold. Firstly, it reduces the effect of noise on the Gaussian process. Since we measure the utility of the opponent's offers in terms of our agent's utility function, it is possible that this value may vary significantly in a given window. This is due to the offers consisting of multiple issues, with the negotiation partners having different utility functions. In such an environment, it is possible that a small change in utility for the opponent can be observed as a large change by our agent. Secondly, it reduces the amount of input data for the Gaussian process. If all of the observed offers were used, there could be thousands of data points, which could significantly slow down the regression process and therefore delay the negotiation. We use the maximum value in each time window, rather than the average, as the maximum represents the best offer that we have observed, and can therefore expect to reach agreement at. Figure 1 shows an example of the input to and output from the Gaussian process performed at time $t_{\text{c}} = 0.25 \cdot t_{\max}$ during a negotiation against *Agent K* in the U1 domain (see Section 4.1 for more regarding the domains and opponents). The figure shows that there is relatively high confidence in the prediction at times close to t_{c}, with the uncertainty increasing at later times.

Based on the prediction of the opponent's future concession which was generated using the regression technique, our strategy then aims to set its concession by optimising the expected utility given that prediction.

3.2 Setting the Concession Rate

Having introduced our use of Gaussian processes in predicting the future concession of the opponent, we now discuss the main contribution of this work to the literature, which is to show how the output of a Gaussian process can be used in setting the concession rate. Specifically, we discuss the way in which we use both the mean, μ, and standard deviation, σ, output by the Gaussian process in setting an optimal concession rate. The aim of this stage of our strategy is to calculate the best time, t^* and utility value, u^* at which to reach agreement. We consider the best time, t^*, to be the point in future time ($t \in [t_{\text{c}}, t_{\max}]$) at which the expected utility of the opponent's offers is maximised, using:

$$t^* = \text{argmax}_{t \in [t_{\text{c}}, t_{\max}]} E_{\text{rec}}(t) \tag{5}$$

where t_c is the current time and $E_{\text{rec}}(t)$ is the expected utility, adjusted by the agent's risk attitude and discounting, of reaching an agreement at time t, given by:

$$E_{\text{rec}}(t) = \int_0^1 D(R(u) f_{[0,1]}(u; \mu_t, \sigma_t), t) \text{d}u \tag{6}$$

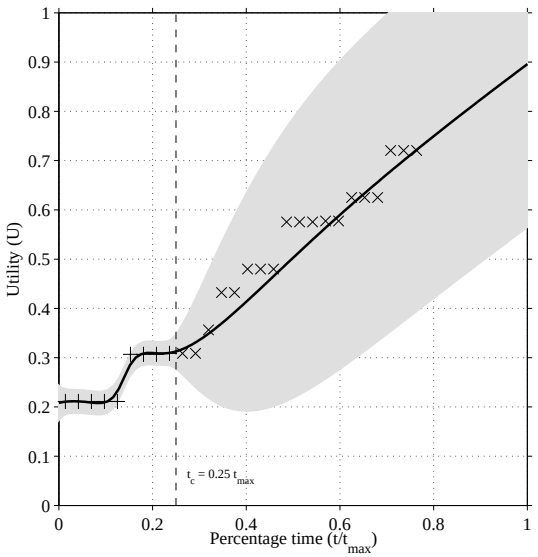

Figure 1: Illustration of the Gaussian process used in a negotiation with *Agent K* in the *Itex vs Cypress* domain, at time $t_c = 0.25 \cdot t_{\max}$. The plus signs are the input data, based on the observed offers. The crosses are based on the actual future offers (which are unknown at time t_c). The mean of the Gaussian output is shown as a solid line, with the shaded area representing the 95% confidence interval.

where $D(\cdot, \cdot)$ is the effect of the discounting factor, given by Equation 2, $R(\cdot)$ is the risk function, given by Equation 3, and $f_{[0,1]}(\cdot)$ is the probability distribution over the values of u, as given by the Gaussian process. The effect of σ_t in this equation is as follows. If, at two points in time, t_1 and t_2, the mean values are the same ($\mu_{t_1} = \mu_{t_2}$), but the standard deviation differs such that $\sigma_{t_1} < \sigma_{t_2}$, then a risk-seeking agent will consider the expected utility at time t_2 to be greater than at time t_1. That is, the risk-seeking agent is prepared to wait for the less certain offer at time t_2, as there is a higher chance that the utility may exceed the value of μ_{t_2}, than it would for μ_{t_1}. The converse applies for risk-averse agents, with risk-neutral agents being indifferent between the two solutions.

In practice, we calculate the optimal time, t^*, to reach agreement, by discretising the values of t, with up to 101 steps, such that $t \in \{0, 0.01 \cdot t_{\max}, 0.02 \cdot t_{\max}, \ldots, 0.99 \cdot t_{\max}, t_{\max}\}$. Integration is carried out in a similar way, by discretising $u \in \{0, 0.01, 0.02, \ldots, 0.99, 1.00\}$

As given by Definition 1, we assume that the utility of the opponent's offers must lie in the range $[0, 1]$. Therefore we use a truncated normal distribution, constrained to fit in the utility range $[0, 1]$, as follows:

$$f_{[0,1]}(u; \mu_t, \sigma_t) = \frac{f(u; \mu_t, \sigma_t)}{f(1; \mu_t, \sigma_t) - f(0; \mu_t, \sigma_t)} \qquad (7)$$

where $f(u; \mu_t, \sigma_t)$ is as given in Equation 4. The mean, μ_t, and variance, σ_t, are those given by the Gaussian process.

Having selected the time, t^*, at which the expected utility of the opponent's offers is maximised, our agent needs to choose a utility, u^*, to offer at that time. The approach that our strategy takes here is to maximise the expected utility of making an offer of utility u. Therefore, the utility, u^*, which should be offered at time t^*, is given by:

$$u^* = \text{argmax}_{u \in [0,1]} E_{\text{offer}}(u, t^*) \qquad (8)$$

The expected utility is calculated based on the probability that an offer of a given utility will be accepted. We assume that an offer of utility u will be accepted at time t^* if $u \leq u_{t^*}$. Since we have a probability distribution over u_{t^*}, we can calculate the probability that $u \leq u_{t^*}$ using the cumulative distribution $F(u; \mu_t, \sigma_t)$. Therefore, our agent's expectation for the adjusted utility (taking into consideration the risk attitude and time discounting) of an offer with utility u is given by:

$$E_{\text{offer}}(u, t^*) = D(R(u) F_{[0,1]}(u; \mu_{t^*}, \sigma_{t^*}), t^*) \qquad (9)$$

where $D(\cdot, \cdot)$ and $R(\cdot)$ are as before, and $F_{[0,1]}(\cdot)$ is the cumulative distribution for $f_{[0,1]}(\cdot)$.

Due to the effect of the standard deviation, note that the risk-seeking agent will find a higher optimal utility than a risk-neutral or risk-averse agent would. In common with the way we find t^*, in practice, in order to find u^*, we also discretise the values of u, with 101 steps, such that $u \in \{0, 0.01, 0.02, \ldots, 0.99, 1.00\}$.

Finally, having determined u^* as the utility to offer at time t^*, our agent needs to choose a utility to offer at the current time, t_c. The agent should not concede immediately to offer u^* at the current time, nor should it wait until t^*. Either of those approaches is likely to result in concession behaviour which is too extreme, especially since they are based on predictions which may be inaccurate. Therefore, and to avoid any additional parameters, our agent's approach is simply to concede linearly between $[t_{\text{lr}}, u_{\text{lr}}]$ and $[t^*, u^*]$, where t_{lr} is the time at which the regression was last performed and u_{lr} is the target utility at that time. Consequently, the target utility, u_τ is given by:

$$u_\tau = u_{\text{lr}} + (t_c - t_{\text{lr}}) \frac{u^* - u_{\text{lr}}}{t^* - t_{\text{lr}}} \qquad (10)$$

Note that the overall concession will not generally be linear, as the predictions of t^* and u^* are revised at the end of each time window, throughout the negotiation.

Once a target utility level has been selected for the current time, our agent needs to select an offer to make.

3.3 Selecting an Offer

Having selected a target utility, u_τ, our strategy needs to generate an offer which has a utility close to that target. In a multi-issue negotiation, there may be many different packages with a similar utility. Under the real-time constraints, the goal is to reach an agreement within a shorter time period, but not necessarily to limit the number of offers made. Therefore, if our agent can generate an offer quickly (even if it does so at random), it can explore more of the outcome space in the available time.

Consequently, our strategy chooses an offer which has a utility close to the target, u_τ, by generating a random offer with a utility in the range $[u_\tau - 0.025, u_\tau + 0.025]$. If an offer cannot be found within this range, the range is expanded, until

a solution is found. In addition, if the target drops below the highest value of the offers made by the opponent, we instead propose the package with that utility that was offered by the opponent. This is since we assume that, for a set of possible offers with utility greater than u_τ, the one which is most likely to be accepted is the one which has previously been offered by the opponent. While we believe that it can be improved, we found that using this simple approach to selecting an offer produced very good results.

4 Empirical Evaluation

In order to evaluate the performance of our strategy, we use the Generic Environment for Negotiation with Intelligent multi-purpose Usage Simulation (GENIUS) [Hindriks *et al.*, 2009]. Using this environment, our agent can be tested under a variety of negotiation domains, where each domain is defined by the utility function of each agent. Each negotiation takes place between a pair of agents, which negotiate within one of these domains. However, the overall performance is calculated for a complete tournament. We continue this section as follows. In Section 4.1 we describe the settings we used in our experiments, then in 4.2, we give the results of those experiments according to the metrics used in ANAC 2010. Finally, Section 4.3 provides an empirical game theoretic analysis of our results.

4.1 Experimental Settings

We compare our agent against the entries submitted to the ANAC 2010 [Baarslag *et al.*, 2010]. As part of this evaluation, we use the domains that were created for the competition, all of which had no discounting factor ($\delta = 0$) (we refer to the *Itex vs Cypress* domain as U1, the *England vs Zimbabwe* domain as U2, and the *Travel* domain as U3). In addition, to test the performance in domains with a discounting factor, we use the same domains, but with each agent having a discounting factor, $\delta = 1$. We refer to these domains as D1, D2 and D3. We get similar results for other values of δ.

In this context, each domain specifies the number and types of issues that are negotiated over by the agents. Furthermore, each domain has two preference profiles which specify the utility functions of each participant. These functions allow each agent to calculate the utility value of any offer that can be made in a particular domain. Each preference profile is represented by a linear additive utility function of the form:

$$U(o) = \sum_{i=1}^{n} w_i U_i(o_i) \tag{11}$$

where w_i is the weight of issue i, $U_i(\cdot)$ is the utility function for issue i, and o_i is the value for issue i in the offer o. The actual values of w_i, U_i and o_i for each preference profile are included in version 3.1 of the GENIUS platform[1]. Each agent knows its own utility function, but not that of its opponent.

For each of these six domains, we run a tournament containing 7 agents, as in ANAC 2010. The other agents that we use in the tournament are the top 6 agents according to the results of ANAC 2010. Whilst we do not know the exact behaviour of the other agents, we suppose that some of them use a combination of stationary and non-stationary (adaptive) strategies. The whole tournament run (all combinations of agent pairs and domains) was repeated 20 times, so as to obtain results with high statistical confidence.

[1]Available from `http://mmi.tudelft.nl/negotiation/index.php/Genius`

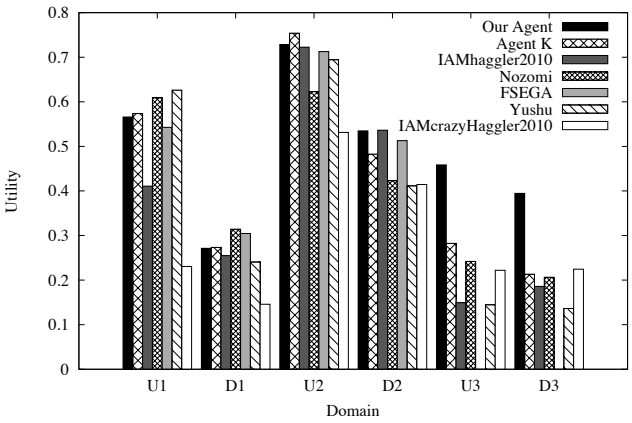

Figure 2: Average results of all agents in all domains, for the risk-neutral ($r = 1$) version of our agent.

Furthermore, we repeat these experiments using two variants of our agent. The first uses a risk-neutral approach ($r = 1$), whilst the other uses a risk-seeking approach ($r = 4$). In all of these experiments, there is a deadline of $t_{\max} = 360$s, and we set our regression window to $t_{\text{window}} = 10$s. These values work well in practice, but to avoid over-fitting, we have not attempted to fine tune them.

4.2 Results in the Competition

Figure 2 shows the results achieved by each agent in each domain. In the domains with the largest outcome spaces (U3 and D3), our agent reaches a score which is considerably higher than any of the other agents (the other agents reach a score no higher than 62% of ours). In the smaller domains, our agent reaches a lower score than some of the other agents, but by a much smaller difference (no more than 16%).

The scores were normalised using the maximum and minimum utility achieved by all agents, using the same technique as [Baarslag *et al.*, 2010]. Averaged over the six tournaments, our risk-neutral agent finished in first place, with an average normalised score of 0.846, which is 23% higher than that of the agent which finished in second place. Table 1 shows the raw and normalised scores of each agent, over the six domains. Using our risk-seeking agent, a slightly higher average normalised score of 0.887 was obtained, which is 33% higher than that of the agent which finished in second place. Table 2 shows the raw and normalised scores of each agent, over the six domains.

It should be noted that, although the risk-seeking version achieves a higher normalised score than the risk-neutral one, it actually achieves a lower raw score. The latter is because the more aggressive nature of the risk-seeking agent causes it to generally concede more slowly, thereby reaching agreements later and with a lower discounted utility. However, the

Agent	Raw Score	Normalised Score
Our Agent	$\mathbf{0.492 \pm 0.007}$	$\mathbf{0.846 \pm 0.015}$
Agent K	0.430 ± 0.006	0.688 ± 0.018
IAMhaggler2010	0.377 ± 0.008	0.572 ± 0.021
Nozomi	0.403 ± 0.008	0.561 ± 0.020
FSEGA	0.345 ± 0.001	0.521 ± 0.009
Yushu	0.376 ± 0.007	0.508 ± 0.019
IAMcrazyHaggler2010	0.295 ± 0.006	0.215 ± 0.012

Table 1: Scores for $r = 1$, with 95% confidence intervals.

Agent	Raw Score	Normalised Score
Our Agent	$\mathbf{0.459 \pm 0.011}$	$\mathbf{0.887 \pm 0.032}$
Agent K	0.395 ± 0.007	0.665 ± 0.017
Nozomi	0.365 ± 0.009	0.562 ± 0.022
IAMhaggler2010	0.342 ± 0.007	0.531 ± 0.020
FSEGA	0.323 ± 0.002	0.495 ± 0.009
Yushu	0.327 ± 0.008	0.423 ± 0.024
IAMcrazyHaggler2010	0.270 ± 0.007	0.248 ± 0.014

Table 2: Scores for $r = 4$, with 95% confidence intervals.

risk-seeking version also causes all other agents in the tournament to achieve a lower utility. This is partially due to the effect of the discounting, but also because the slower concession requires the opponent to concede more in order to reach agreement. As a result, as we can see from the normalised score, this risk-seeking behaviour has a greater negative effect on most of the opponents.

Although not a factor in achieving a high tournament score, another measure of success of a strategy is whether it performs well in negotiations with agents using the same strategy. We therefore consider how the strategies perform in self-play. Our agent achieves the highest average self-play score. The maximum possible average self-play score over the six domains is 0.786, with our agent achieving 91.9% of that value. Table 3 shows the average score achieved by each agent over the six domains.

4.3 Results of Empirical Game Theoretic Analysis

A limitation of the tournament setting is that it measures average performance considering only one agent for each type of strategy among a set of possible strategies. Naturally, this raises the question: would our strategy still be robust (and would it still win) in tournaments where the mix of strategies was different? Would any of the agents in those tournaments have an incentive to deviate (switch to a different strategy)?

In order to answer such questions, we perform an empirical game theoretic analysis of the tournament results, using the technique developed by [Jordan *et al.*, 2007] to analyse the Trading Agent Competition. We consider single-agent

Agent	Self-play Score	% of maximum
Maximum	0.786 ± 0.000	100.0 ± 0.0
Our Agent ($\mathbf{r = 1}$)	$\mathbf{0.722 \pm 0.006}$	$\mathbf{91.9 \pm 0.7}$
Our Agent ($\mathbf{r = 4}$)	$\mathbf{0.605 \pm 0.028}$	$\mathbf{76.9 \pm 3.6}$
IAMhaggler2010	0.559 ± 0.026	71.1 ± 3.4
Agent K	0.547 ± 0.003	69.6 ± 0.4
Nozomi	0.505 ± 0.030	64.2 ± 3.8
Yushu	0.412 ± 0.044	52.4 ± 5.6
FSEGA	0.410 ± 0.000	52.2 ± 0.0
IAMcrazyHaggler2010	0.000 ± 0.000	0.0 ± 0.0

Table 3: Self-play scores, with 95% confidence intervals.

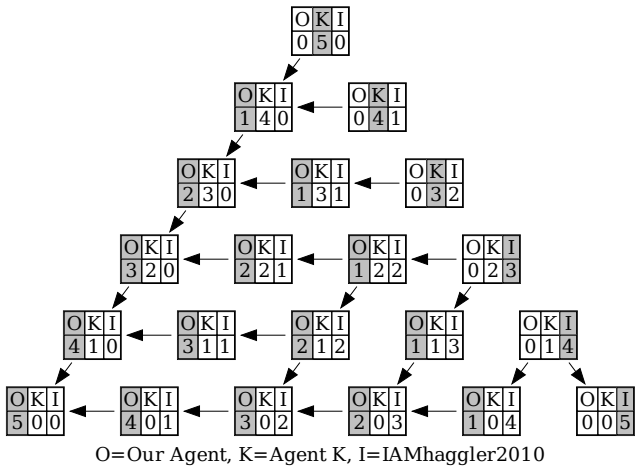

Figure 3: Deviation in domain D3, for our strategy (with $r = 1$), and the strategies of *IAMhaggler2010* and *Agent K*. The shaded strategies are those which achieve the highest scores.

deviations, that is, where there is an incentive for one agent to change its strategy, assuming that all other agents maintain their current strategy. We use this technique to search for Nash equilibria, in which there is no incentive for any single agent to change to another strategy.

We consider all tournaments consisting of 5 agents, each using one of the top three strategies found in the results in Section 4.2. Analysis of the larger 7-agent tournaments with the full combination of strategies was also performed, but due to space constraints in this paper, we report the similar results from a smaller tournament setup. Under this analysis technique we find two equilibria as follows:

1. All agents use our strategy.
2. All agents use the *IAMhaggler2010* strategy.

In Figure 3, we show these deviations for the strategies used by the top three agents in domain D3 (the *Travel* domain, with a discounting factor, $\delta = 1$), which is the largest and most complex domain used in the 2010 competition. Each node represents a possible mixture of the three strategies in a tournament. The corners of the triangle represent strategy mixtures in which all agents play the same strategy. Each arrow indicates a statistically significant single agent deviation to a different strategy. Furthermore, the shaded strategy at each node represents the one which achieves the highest score and the equilibria are the nodes with no outgoing arrow.

In this particular domain, from all non-equilibria strategy mixtures, there exists a path of statistically significant deviations which leads to the equilibria containing only our strategy. Furthermore, in all non-equilibria strategy mixtures except the one in which one agent uses the *Agent K* strategy and the rest use the *IAMhaggler2010* one, all such paths of deviations lead to the equilibria containing only our strategy. This represents 97.5% of the possible mixtures, demonstrating the robustness of our strategy.

We also performed the same analysis, with $r = 4$ instead of $r = 1$. For this setting, we observe a similar result, with the same pair of equilibria, but with slightly fewer (83.1%) of

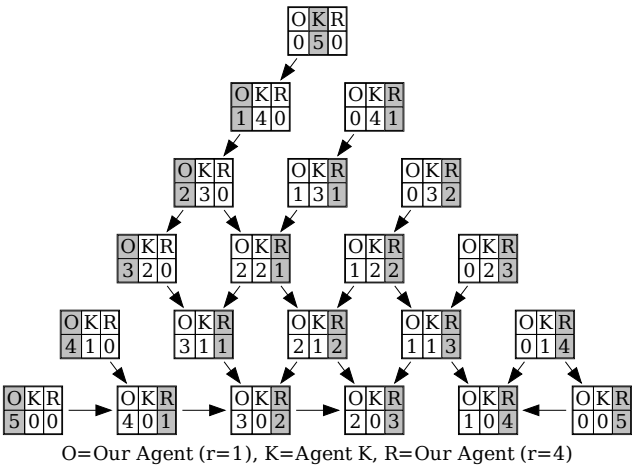

Figure 4: Deviation in domain D3, for our strategy (with $r = 1$ and with $r = 4$), and the strategy of *Agent K*. The shaded strategies are those which achieve the highest scores.

the possible mixtures having all paths of deviations leading to the equilibria containing our strategy.

Having shown that both versions of our strategy perform well against the other agents, we now consider how they perform in negotiation tournaments where both are present. We therefore repeat the analysis, but this time considering the risk-seeking version of our strategy ($r = 4$, denoted R), the risk-neutral version ($r = 1$, denoted O), and *Agent K*. We observe two equilibria, as shown in Figure 4. Both equilibria consist entirely of our strategy, as follows:

1. Two agents with $r = 1$ and three agents with $r = 4$.
2. One agent with $r = 1$ and four agents with $r = 4$.

There are no equilibria in which all agents use either O or R because, in this domain, the score achieved by strategy O against strategy R is greater than the self-play score of R. That is, there is an incentive for strategy R agents in negotiations against an opponent using the same strategy to change to strategy O. This is since, if both agents are risk-neutral, one of them can deviate to the risk-seeking strategy, and in doing so, extract a greater utility from the concessive, risk-neutral agent. Similarly, in a negotiation session where both agents use strategy O, there is an incentive for one of them to change to strategy R. This is because, if both agents are risk-seeking, they will be likely to take longer to reach an agreement, as they will concede slowly. In a discounted domain, slow concession reduces the utility of agreement. Therefore, by deviating to a more concessive strategy (with $r = 1$), an agent can obtain a higher utility by reaching agreements earlier. However, in doing so, the utility of the remaining risk-seeking agents will also increase, potentially by a greater amount than for the deviating agent. Therefore, in tournaments containing both strategies O and R, it is the less concessive, risk-seeking strategy, R that is more likely to 'win' the tournament (by achieving a higher average score than its opponents), even though the scores in individual negotiation sessions can be higher when the risk-neutral strategy, O is used instead.

5 Conclusions and Future Work

In this work, we have developed a negotiation strategy that uses a principled approach to concession, based on an estimate of the opponent's future concession. By developing an agent which uses this strategy, we have demonstrated the performance of our strategy in a setting containing multiple issues, uncertainty and real-time constraints. Using the benchmark of the Automated Negotiating Agent Competition, we have shown that our agent outperforms other state-of-the-art agents in both a tournament setting and in self-play.

There are a number of areas where we believe improvements could be made. Our current work has shown that the risk-neutral version of our strategy is able to reach a higher raw score than our purely risk-seeking version. We believe that our strategy could achieve an even higher score by using a hybrid risk function, which is risk-seeking for low utility values, but risk-averse for high utility values. Furthermore, our strategy's current approach to selecting an offer with a given value is to choose an offer at random from those which have a utility close to our agent's target. This may be improved by modelling the opponent's preferences, in order to choose packages at a given utility level which maximise the likelihood that they are accepted by the opponent.

References

[An *et al.*, 2010] B An, N Gatti, and VR Lesser. Searching for pure strategy equilibria in bilateral bargaining with one-sided uncertainty. *Proc. of the Ninth Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 1607–1608, 2010.

[Baarslag *et al.*, 2010] T Baarslag, K Hindriks, CM Jonker, S Kraus, and R Lin. The first automated negotiating agents competition (ANAC 2010). *New Trends in Agent-based Complex Automated Negotiations, Studies in Computational Intelligence*, 2010.

[Brzostowski and Kowalczyk, 2006] J Brzostowski and R Kowalczyk. Adaptive negotiation with on-line prediction of opponent behaviour in agent-based negotiations. *Proc. of the 2006 IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, pages 263–269, 2006.

[Coehoorn and Jennings, 2004] RM Coehoorn and NR Jennings. Learning on opponent's preferences to make effective multi-issue negotiation trade-offs. *Proc. of the Sixth Int. Conf. on Electronic Commerce*, pages 59–68, 2004.

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

[Fatima *et al.*, 2007] SS Fatima, M Wooldridge, and NR Jennings. Approximate and online multi-issue negotiation. *Proc. of the Sixth Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 156–170, 2007.

[Hindriks and Tykhonov, 2008] K Hindriks and D Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. *Proc. of the Seventh Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, 1:331–338, 2008.

[Hindriks *et al.*, 2009] K Hindriks, CM Jonker, S Kraus, R Lin, and D Tykhonov. GENIUS: negotiation environment for heterogeneous agents. *Proc. of the Eighth Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, 2:1397–1398, 2009.

[Jordan *et al.*, 2007] PR Jordan, C Kiekintveld, and MP Wellman. Empirical game-theoretic analysis of the tac supply chain game. *Proc. of the Sixth Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 1188–1195, 2007.

[Lin *et al.*, 2008] R Lin, S Kraus, J Wilkenfeld, and J Barry. Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence*, 172:823–851, 2008.

[Rasmussen and Williams, 2006] CE Rasmussen and CKI Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[Robu *et al.*, 2005] V Robu, DJA Somefun, and JA La Poutré. Modeling complex multi-issue negotiations using utility graphs. *Proc. of the Fourth Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 280–287, 2005.

[Rubinstein, 1982] A Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, 50(1):97–109, 1982.