IAMhaggler: A Negotiation Agent for Complex Environments

Colin R. Williams, Valentin Robu, Enrico H. Gerding, and
Nicholas R. Jennings

School of Electronics and Computer Science, University of Southampton,
University Road, Southampton, SO17 1BJ, UK

{crw104,vr2,eg,nrj}@ecs.soton.ac.uk

Abstract We describe the strategy used by our agent, IAMhaggler, which
finished in third place in the 2010 Automated Negotiating Agent Compe-
tition. It uses a concession strategy to determine the utility level at which
to make offers. This concession strategy uses a principled approach which
considers the offers made by the opponent. It then uses a Pareto-search al-
gorithm combined with Bayesian learning in order to generate a multi-issue
offer with a specific utility as given by its concession strategy.

1 Introduction

We present the negotiation strategy, called IAMhaggler, which we developed
and entered into the first Automated Negotiating Agent Competition [1].
The competition is a tournament between a set of agents which each perform
bilateral negotiation using the alternating offers protocol. The negotiation
environment consists of multiple issues (which may take continuous or dis-
crete values), where there is uncertainty about the opponent's preferences.
The environment also uses a real-time discounting factor and deadline.

Our negotiation strategy consists of two parts. First, it uses a principled
approach to choose a concession rate based on the concession of the opponent,
the real-time discounting factor and the negotiation deadline. Then, using the
utility level chosen according to this concession strategy, it generates a multi-
issue offer at that utility level, whilst attempting to maximise the likelihood
that the opponent will form an agreement by accepting the offer.

Our strategy makes two main contributions to the literature on multi-issue
negotiation with uncertainty about the opponent's preferences. Specifically,
we propose a novel approach to determining our agent's concession rate by
taking into account the real-time discounting factor and deadline in order to
choose a concession rate which is a best response to that of the opponent (see

Section 2). Furthermore, we combine two existing approaches (Pareto-search and Bayesian learning) which represent the state of the art in continuous and discrete domains respectively, to generate a multi-issue offer in domains containing both discrete and continuous issues (see Section 3).

2 Setting the Concession Rate

Our concession strategy is composed of two parts. The first tries to estimate the opponent's future concession based on observations of the opponent's offers. This estimate is then used to set an appropriate rate of concession. We discuss both parts in turn in Sections 2.1 and 2.2.

2.1 Learning the Opponent's Concession

Our agent's approach is to set its concession as a best response to the opponent's concession throughout the negotiation. Therefore, the first stage of our agent's concession strategy is to build an estimate of the opponent's future concession. To this end, our agent observes the offers that the opponent makes, recording the time that the offer was made, along with the utility of that offer according to our own agent's utility function. We assume that the opponent is concessive over time, and therefore, for any offer which gives us a lower utility than that of a previous offer from the opponent, we instead record the highest utility that we have observed so far. A further motivation for this behaviour is that we make the assumption that our opponent is likely to accept any offer that it has previously made. Therefore, if we were to propose the best offer that we have seen so far again, we assume that the opponent will accept. We believe this to be a reasonable assumption and one which holds for our agent. This gives us a function over time of the opponent's concession in terms of our own utility. At this stage we ignore the effect of the discounting factor on our utility.

Now, in order to predict the opponent's concession at any future point in time during the game, we apply non-linear regression to the opponent's concession curve, and in doing so assume that the observed points will roughly fit to a curve of the form:

$$U_o(t) = U_0 + e^a t^b, \tag{1}$$

where $U_o(t)$ is the utility of the offer made by the opponent at time t and U_0 is the utility of the opponent's first offer. The constants a and b are the parameters which we find using a least mean squares curve fitting algorithm.

Once the constants a and b have been found, we use Equation 1 to estimate the utility of the opponent's offers at any time during the negotiation session. After every offer from the opponent, we update our estimation, by repeating the process described in this section.

2.2 Choosing the Concession Rate

Once we have an approximation of our opponent's concession in terms of our utility, we are able to use this information to set our own rate of concession.

Our approach to this is to use standard time dependent concession, as described by [2]. Specifically, we use the polynomial time concession function:

$$U_p(t) = U_0 - (U_{\min} - U_0)\frac{t}{t_{\max}}^{1/\beta},$$ (2)

where U_0 is the initial utility, $U_{\min}$ is the reservation utility (which the agent will not concede beyond), $t_{\max}$ is the negotiation deadline and β is the parameter that affects the rate of concession. The β value can be partitioned into three types: tough ($\beta < 1$), linear ($\beta = 1$) or weak ($\beta > 1$). Our approach to finding an appropriate value for β proceeds as follows, and we give a graphical example of this approach in Fig. 1. We firstly apply our discounting factor to our model of the opponent's concession function (Equation 1), to create a function which gives us an estimate of the discounted utility of our opponent's offers at any point in the negotiation session. Therefore, the discounted utility function is given by:

$$U_d(t) = (U_0 + e^a t^b)e^{-\delta t},$$ (3)

where δ is the discounting factor.

If the opponent's future concession is as predicted, then our agent cannot achieve a utility higher than the maximum on the discounted opponent concession curve (given by Equation 3) for the time period at which an agreement can be reached. By this we mean that parts of the curve that represent times in the past, or times which are beyond the negotiation deadline should be ignored. Therefore, the next step is to solve:

$$t^* = \underset{t_{\mathrm{now}} \leq t \leq t_{\max}}{\arg\max} \, ((U_0 + e^a t^b)e^{-\delta t}),$$ (4)

where t_{now} is the current time.

By solving Equation 4, our agent has identified the time t^* at which the discounted utility to our agent of our opponent's offers is likely to be maximised. Given this, our agent chooses a target utility level, which matches the estimated utility of the opponent's offer (without any discounting) at that
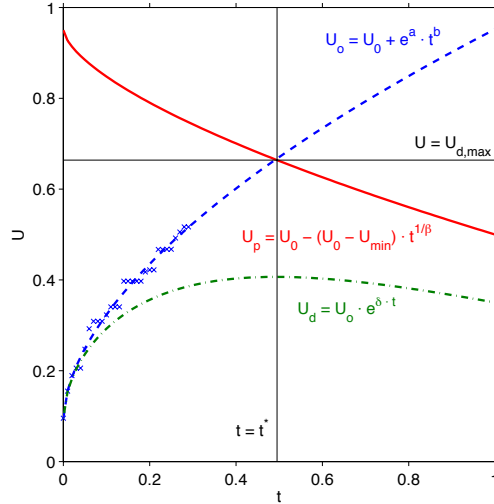
Fig. 1 Setting our Concession Rate. The crosses represent the best offers made by the opponent. The estimated future concession of the opponent is shown by the dashed (undiscounted) and dot-dashed (discounted) lines. The maxima on the dot-dashed line, indicated by the vertical line, represents the time (t^*) at which the expected discounted utility of the opponent's offers is maximised. The horizontal line shows the expected undiscounted utility of the opponent's offer ($U_{d,\max}$) at that time. The solid curve is our planned concession

time. This is given by:

$$U_{o,\max} = U_o(t^*) \tag{5}$$

We now have a point in time t^* at which we expect to reach agreement, and a target utility level $U_{o,\max}$ that we should use at that time. We then choose the value of β such that the concession curve (Equation 2) passes through $[t^*, U_{o,\max}]$. By rearranging Equation 2, we find β as follows:

$$\beta = \frac{\log(t^*)}{\log\left(\dfrac{1 - U_{o,\max}}{1 - U_{\min}}\right)} \tag{6}$$

A limitation of this strategy is that, at the beginning of a negotiation, the curve fitting is performed using a small number of points, and therefore the curve may not accurately reflect the future concession of the opponent. Thus, we set upper and lower bounds on our actual concession parameter (β) so that the agent does not use a concession strategy that is too extreme, either by conceding too quickly at the beginning of the session (thereby forming a low utility agreement), or by playing too tough (thereby conceding too little).

Having shown our concession strategy, we now discuss how the strategy produces a multi-issue offer at the utility level selected by the concession strategy.

3 Negotiating over Multiple Issues

In the previous section we showed how our agent can choose its desired utility level at any point during the negotiation session. However, in a multi-issue negotiation, there are likely to be a number of different offers at any given utility level. Our agent is indifferent between these offers, since they all result in the same utility. However, the opponent has a different utility function to ours. Therefore, our aim, and the basis of our negotiation strategy, is to select the offer which maximises the utility to the opponent whilst maintaining our desired utility level. The reason for this is that the opponent is more likely to accept offers with a higher utility, and, from a performance perspective, such outcomes are more Pareto-efficient. The challenge here is to try to do so without knowing the opponent's utility function.

In this section, we consider issues to belong to one of two classes: *ordered*, and *unordered*. Ordered issues have an ordering that is common to and known by both agents, though the agents may have different preferences over the issue values. Conversely, unordered issues do not have a common ordering.

In domains which consist solely of ordered issues, our agent does not attempt to learn the opponent's utility function. Instead, we use a Pareto-search approach to selecting an offer. In domains with unordered issues, we extend this approach by using Bayesian learning to learn the opponent's utility functions for the unordered issues.

To this end, in Section 3.1, we describe our approach in a domain without unordered issues, and in Section 3.2, we consider domains with such issues.

3.1 Domains without Unordered Issues

In this section, we present our approach to selecting the package with a given utility that we consider to be closest to the best offer that we have seen from our opponent. At this point, we only consider ordered issues.

Our strategy is based on the Pareto-search approach developed in [3, 5]. We consider our agent's utility function to be a mapping from a multi-dimensional space (in which there is a dimension representing each ordered issue) to a real value which represents the utility of the outcome. Our strategy treats integer based issues in the same way as continuous issues. For example, for a domain which consists of only one linear and one triangular issue, this multi-dimensional space is shown graphically in Fig. 2.

Now, by taking a cross-section of the utility space, we can construct an iso-utility space, which is a multi-dimensional space, with the number of dimensions equal to the number of ordered issues. This space represents all packages which result in a particular utility for our agent.

Moreover, the iso-utility space that is chosen at a particular time is the one which represents our current desired utility level (as decided by our concession
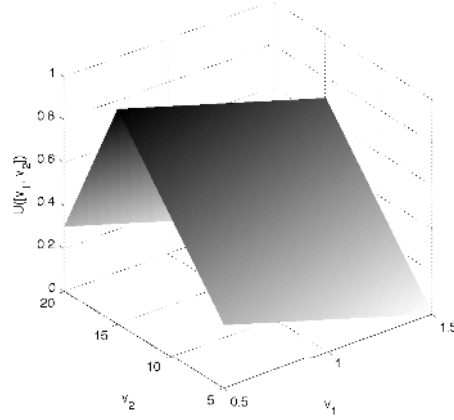
Fig. 2 Multi-dimensional space representing the utility of outcomes in a domain with one linear issue and one triangular issue. Values v_1 and v_2 are the values of the two issues. $U([v_1, v_2])$ is the utility of the offer represented by those values

strategy, which we detailed in Section 2). Based on the work of [5], we use projection to find the point on our iso-utility space which is closest to the best offer our agent has received from its opponent.

Specifically, in terms of closeness between two offers $[v_1, v_2, ..., v_n]$ and $[y_1, y_2, ..., y_n]$, we use the Euclidean distance, that is:

$$\sqrt{\sum_{i=1}^{n} \left(\left(\frac{v_i - y_i}{\text{range}_i} \right)^2 \right)}, \tag{7}$$

where range_i is the range of values allowed for issue i. The reason that we divide by the range is to ensure that the scale of the issue's values does not affect the distance measurement. In doing so, all of the issues are considered to have equal weightings. Otherwise, the distance calculation could be excessively biased by an issue with a large range relative to the other issues.

3.2 Domains with Unordered Issues

In order for our strategy to negotiate in domains with unordered issues, we need to make some modifications to it. In particular, we cannot treat the unordered issues as further dimensions in our space, since they cannot be ordered based on their similarity. To address this problem, we continue to create an iso-utility space to represent the ordered issues. However, to handle the additional complexity of unordered issues, we create an iso-utility space
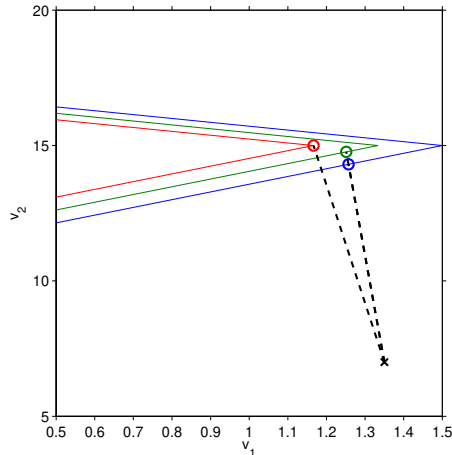
Fig. 3 Projection of a point representing an opponent's offer (at $[1.35, 7.0, v_3]$, marked with a cross) onto an iso-utility space for a domain with one linear, one triangular and a discrete unordered issue. The projections are shown for each of the three discrete combinations

for each combination of the unordered issues. As an example, consider a domain with one linear, one triangular and one discrete issue. The discrete issue can take the values '*red*', '*green*' or '*blue*'.

For each combination of the unordered issues, we create a multi-dimensional space, and use the iso-utility projection method (described in Section 3) to find a solution for each of those combinations. We demonstrate this projection in Fig. 3, where the opponent's offer $[1.35, 7.0, v_3]$ (we write v_3 to represent the value of the unordered issue 3, since it does not affect the projection) is projected to give the solutions $[1.17, 15.0, \text{red}]$, $[1.25, 14.8, \text{green}]$ and $[1.26, 14.3, \text{blue}]$. For some combinations, the maximum overall utility available from the package with the best values for the ordered issues may be lower than our current utility level. In this case, there will not be a solution which contains this combination.

Once a solution has been found for each combination of the unordered issues, it is necessary for our agent to choose one of these solutions as its offer. Our agent is indifferent between each of them, as they all belong to the same iso-utility space, resulting in them having an identical utility. However, their utility to the opponent may vary, and in order to negotiate efficiently, we should choose the one which maximises the utility to the opponent.

Now, since we assume that the opponent's utility function is unknown, we need a way to estimate the utility of the opponent. In our work, this is done using the approach taken by [4], using Bayesian updating to learn the preferences of the opponent in a scalable manner.

As the agent receives offers from the opponent, it uses Bayes' rule to update the probability of each hypothesis. It then evaluates the solution for each

unordered combination using our model of the opponent's utility function, in order to obtain an estimate of the utility of the offer to the opponent. The overall solution that is chosen is then the one which maximises the opponent's utility according to our model of its utility function.

In order to ensure that our approach remains computationally tractable even in domains with large outcome spaces, we limit the number of combinations of unordered issues that we perform the iso-utility projection method for. Specifically, we choose a maximum of 1000 such combinations, by identifying those which maximise the sum of our utility and our opponent's utility (according to our model of the opponent's utility function).

4 Results and Conclusions

In this work, we have described our negotiating agent, IAMhaggler, covering both the process that our agent uses to set its concession rate, and the way in which it selects an offer with a given utility. In doing so, we have combined and extended several existing approaches, and designed a new adaptive strategy.

Our agent finished in 3rd place in the 2010 competition (according to the official results[1]). After the competition, we identified some minor implementation problems with our agent. By fixing these problems and then re-running the competition setup, our agent was able to finish in 2nd place. Since the competition domains did not include a discounting factor, we performed a further re-run in which we used the competition domains but with a discounting factor ($\delta = 1$). In this re-run, our agent finishes in 1st place.

We plan to extend our work by investigating how the agent can better adjust its parameters (such as the reservation value) based on its knowledge of the domain.

References

1. Baarslag, T., Hindriks, K., Jonker, C.M., Kraus, S., Lin, R.: The first automated negotiating agents competition (ANAC 2010). This volume.
2. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. Robotics and Autonomous Systems **24**(3), 159–182 (1998)
3. Faratin, P., Sierra, C., Jennings, N.R.: Using similarity criteria to make issue trade-offs in automated negotiations. Artificial Intelligence **142**(2), 205–237 (2002)
4. Hindriks, K., Tykhonov, D.: Opponent modelling in automated multi-issue negotiation using bayesian learning. Proc. of the 7th Int. Joint Conference on Autonomous Agents and Multiagent Systems **1**, 331–338 (2008)
5. Somefun, D.J.A, Gerding, E.H., La Poutré, J.A.: Efficient methods for automated multi-issue negotiation: Negotiating over a two-part tariff. Int. Journal of Intelligent Systems **21**(1), 99–119 (2006)

---

[1] `http://mmi.tudelft.nl/negotiation/images/0/03/Results.pdf`