ProbaBot: an Agent Submitted to the ANAC 2025 ANL

Loes Peters,
Centrum Wiskunde & Informatica,
Eindhoven University of Technology,
loes.peters@cwi.nl

27th May 2025

Abstract

This report details the design of ProbaBot, an agent submitted to the Automated Negotiation League of the Automated Negotiating Agent Competition 2025. The coordination strategy of this agent is based on expected utility that can be gained in the remainder of the negotiation, while the bidding strategy tries to offer bids which give away as little information as possible to the opponent.

1 Introduction

The Automated Negotiating Agent Competition (ANAC) [1] is an international tournament that has been running since 2010. It brings together researchers from the negotiation community in hopes of advancing research in the field of autonomous agent design. ANAC is divided into two leagues; the Automated Negotiation League (ANL) and the Supply Chain Management League (SCML).

This year's challenge of ANL is sequential multi-deal negotiation. In particular, participants are tasked with the design of an agent that can negotiate with multiple opponents in sequence to reach a desired combination of deals from each separate subnegotiation. A negotiation consists of several agents, namely a center agent, which negotiates with several opponents in sequence, and edge agents, which are the opponents of the center agent. The center agent has a utility function over the domain of all combinations of deals made in the negotiation. In every subnegotiation the center agent may have a side utility function, which determines the utility of bids in that subnegotiation. The utility function of an edge agent is called the edge utility function. The designed agent is required to perform both as a center agent and as an edge agent.

ProbaBot is one of the agents submitted to the ANAC 2025 ANL. This report describes the design of ProbaBot, which can be divided into three strategies, in Section 2. A brief discussion of the design of ProbaBot is given in Section 3.

2 The Design of ProbaBot

The design of negotiation agent ProbaBot can be subdivided into three parts, namely the coordination strategy, the bidding strategy, and the acceptance strategy. While the bidding strategy and the acceptance strategy are used by ProbaBot in the role of both center agent and edge agent, ProbaBot only utilizes the coordination strategy in its role as center agent. In the remaining of this report, these three strategies are explained in more detail.

2.1 Coordination Strategy Center Agent

The main challenge of this year's ANAC 2025 Automated Negotiation League is the coordination of the sequential subnegotiations. The task of determining which deal to aim for in every subnegotiation is nontrivial. The idea behind ProbaBot's coordination strategy is to define the side utility of a bid as the expected center utility which can be obtained from this bid. This strategy should aim not necessarily for the best overall outcome, but rather for outcomes in a subnegotiation which have a good chance of reaching a high center utility score.

ProbaBot uses the expected complete outcome over all subnegotiations given the current bid in that subnegotiation as side utility function of the subnegotiation. To this end, ProbaBot needs to estimate the probability that a certain complete outcome over all subnegotiations is the outcome of the entire negotiation. We assume that every outcome is equally likely to become the outcome of the entire negotiation. Note that this is quite a strong assumption, which is not necessarily true in many scenarios. However, the agent can be adapted to work with an opponent model, which can be used as probabilities that the opponent accepts certain bids. These probabilities can be combined into probabilities of certain complete outcomes being the outcome of the entire negotiation.

Now the edge utility of a bid in subnegotiation j is defined as the sum over all complete outcomes (of the entire negotiation with multiple edge agents), which have this bid as the outcome for subnegotiation j, of the probability of this complete outcome times the utility of this complete outcome. Suppose the center agent is in subnegotiation j and the previous subnegotiations have resulted in deals $o_1, \ldots, o_{j-1}$. Let Ω_l be the outcome space of subnegotiation l for $l = 1, \ldots, n$. Then the side utility of some bid $b \in \Omega_j$ is defined as

$$u_j(b) := \sum_{\mathbf{b}=(o_1,\ldots o_{j-1},b,b_{j+1},\ldots,b_n)\in\Omega_1\times\cdots\times\Omega_n} \mathbb{P}(\mathbf{b} \text{ is outcome of entire negotiaton}) u(\mathbf{b}),$$

where u is the utility function of the center agent. Note that for ProbaBot the probabilities in this sum are all equal, and thus independent of $\mathbf{b}$.

2.2 Bidding Strategy

The bidding strategy of an agent determines the next bid it offers its opponent. The idea behind the bidding strategy of ProbaBot is to give away as little information about the agent's preferences as possible to the opponent. This should, in theory, make it harder for the opponent to model the agent's preferences and leave more room for deceiving the opponent.

The first question one might ask is; what constitutes "giving away information"? Consider, for example, a negotiation domain with two issues, each taking on three values, say 1, 2 and 3. If an agent first offers the bid $(1, 1)$, then $(1, 2)$ and after that $(2, 1)$, then a good next bid to offer can be $(2, 2)$, since the opponent might deduce from this (by comparing the last two bids) that 1 is preferred over 2 for the second issue. However, the opponent could already deduce this by comparing the first two bids. So this last bid does not give away any new information about the agent's preferences.

The reader might have noticed an implicit assumption underlying this bidding strategy in this example. Specifically, we assume that the bids an agent offers are nonincreasing in their utility value. Moreover, it is assumed that the utility function (whether edge or side utility) is linear and additive, or at the least that the change in utility from substituting a value in one issue is independent of the values of the other issues.

The bidding strategy of ProbaBot works as follows. If ProbaBot has not offered any bids yet, it uses the edge utility or the side utility to determine its best possible bid to make and offers this. Otherwise, at relative time t in the (sub)negotiation, a polynomially conceding aspiration level is calculated. The aspiration level $a(t)$ at a given relative time t is

$$a(t) = M(1 - t^x), \tag{1}$$

where M is the maximum aspiration (dependent on the side or edge utility function) and x is the chosen exponent. The exponent used for ProbaBot is $x = 3$, which gives a boulware aspiration function making ProbaBot concede less in the beginning of the (sub)negotiation and more quickly at the end. A set of bids which have utility at least $a(t)$ is generated. From these bids, the bid which has not been offered before in this (sub)negotiation and which does not give away new information is chosen. If such a bid does not exist, the condition that it has not been offered before is dropped. Note that not giving away new information means that when comparing this bid to the previous bid, the combination of different issues of the previous bid compared this bid has been observed before (like the the example described above).

2.3 Acceptance Strategy

The acceptance strategy of an agent determines when a bid from the opponent is accepted or rejected. In the worst case, an agent can also decide to end the (sub)negotiation altogether. The acceptance strategy of ProbaBot is based on the polynomially conceding aspiration curve from Equation (1).

Given a bid from the opponent at relative time t, ProbaBot ends the (sub)negotiation if offering its best bid afterwards is worse or as good as ending the (sub)negotiation at that point. Note that the best bid ProbaBot can make, is dependent on the side utility function (for center agents) or the edge utility function (for edge agents). If the (sub)negotiation is not ended, ProbaBot accepts the bid from the opponent if the (side or edge) utility of this bid is at least $a(t)$. Otherwise, ProbaBot rejects the bid from the opponent and offers its next bid according to the bidding strategy.

3 Discussion

In this section we briefly discuss the design of ProbaBot, highlighting some shortcomings and giving a point for improvement. First, the coordination strategy of ProbaBot assumes that all (overall) outcomes of the negotiation are equally likely. This is often not the case as opposing edge agents also have their own preferences given by their edge utility function. An opponent model can be added to the bidding strategy to more accurately define the probability of a certain outcome becoming the outcome of the whole negotiation. Note that this also requires some assumption on edge agents having similar preferences, since the center agent encounters the edge agents in sequence.

Moreover, the assumption on linear additive utility functions, used for the bidding strategy is quite restrictive. Note that already when the center utility function is the maximum of the side utility functions, this assumption does not hold. However, since the bidding strategy of ProbaBot only offers bids above some boulware aspiration level $a(t)$, the impact of this assumption might not be as large.

References

[1] *ANAC 2025 Competition*. Accessed: 22-05-2025. URL: `https://scml.cs.brown.edu/anac`.