ANL2025: Lookahead Utility Estimation for Multi-Deal Negotiation

Garrett Seo, Tri-an Nguyen, Xintong Wang

Rutgers University

garrett.seo@rutgers.edu, tdn39@scarletmail.rutgers.edu, xintong.wang@rutgers.edu

1 Introduction

In ANL 2025 that features multi-deal negotiation, our designed agent, when being the center agent, plans for the future negotiations by doing a look-ahead utility estimation. As future sub-negotiations affect current negotiations, the value of an outcome is dependent on the actions made in later sub-negotiations. We model the negotiations as an extensive-form game tree, where each node of a level represents one negotiation and its children represent the possible outcomes at future sub-negotiations. We assign a probability distribution over the outcomes at each node, depending on its expected utility. We then backpropagate expected utilities all the way to root of the tree.

From here, the negotiation strategy is based on these expected utilities. As there is no time discount, we aim to reach an agreement close to the end of a sub-negotiation to get as much information from the opponent as possible (based on their offers). We learn from their proposals with our utilities on their bid to estimate how they are conceding. We aim to predict the maximum utility we can achieve in the negotiation and propose an offer in this region as we are getting closer to the end of a sub-negotiation.

2 Center Coordination

Our agent deals with sequential multi-deal negotiations by running a lookahead estimation at the beginning of each sub-negotiation. This lookahead is used to calculate the expected value of each outcome within the sub-negotiation, and is implemented by abstracting the negotiation into a game tree. Each level in the tree represents a sub-negotiation while the root of the tree represents the beginning of a full multi-deal negotiation. The lookahead runs a backwards induction over the outcome space. At any given node, it searches through every outcome and calculates the expected utility of achieving that outcome and entering in the next sub-negotiation. Once this is completed, the utility of the root node is equivalent to a sum over all the expected values of it children.

We maintain a probability distribution over all the expected values of the child outcomes where their probabilities are generated via a softmax over their expected values with some temperature (which we performed some hyperparameter search using Bayesian optimization). The utility of the root node is then an expectation based off this probability distribution.

3 Bidding Strategy

Because the utility of the negotiation is not discounted over time, we are able to make use of all the steps within a sub-negotiation. We want to use all these steps in order to gather more information from the opponent through their proposals, which describe which outcomes they prefer and the change in their bids will describe how they are conceding.

Initially, we always bid offers that are higher than some utility. We do this to offer highly preferred outcomes that will either be rejected (we collect more information) or be accepted which is beneficial for us anyways.

We record the proposals that the opponent offers, the utility of that outcome had we accepted, and the time within the sub-negotiation. We then fit a utility curve over time which describes the opponent's concession over time. We fit the curve according to a time-based strategy: $t^e + u_o$, in which t represents the time in the sub-negotiation, e represents the opponent's concessive degree, and u_0 represents the utility of the opponent's first offer.

To provide some intuition, an agent is more Boulwarish if $e > 1$, linear if $e = 1$, and conceding if $e < 1$. Then, we predict a maximum utility we can achieve and we take the minimum between this predicted maximum utility and the value of the time-based strategy. The fitted curve aims to accurately predict the agent's conceding degree e and the maximum utility we can achieve.

In the last timestep to propose, we then bid an outcome near the region of this maximum utility according to the curve. In order to achieve more agreements, we bid an offer whose utility is somewhere in between the highest utility of a proposal the opponent offered and the predicted maximum utility we can achieve. We prioritize offers based on which index-specific issues have been more proposed by the opponent.

4 Acceptance Strategy

The idea is to never accept an offer until the last time step. This is helping us to gather as much information from the opponent as possible and to let the opponent concede as much as possible as well.

We generally reject every proposal until the last timestep. At the last timestep, if we are the edge agent, we only accept the last offer if it's higher than our reservation value. We also accept the last offer if its offer is near the maximum utility we predicted from the curve.

5 Appendix

5.1 LookAhead Details

Typically, game trees show the next state after an agent takes an action. The main concern with the lookahead strategy is that we cannot possibly search the whole tree. To put in perspective, the default parameters for make_multi_deal_scenario results in a tree of $((7^3 + 1)^5)^{100}$ states.[1]

Given the time limit on the negotiation, we cannot evaluate all of these. To deal with this, we used two strategies. First, we represent each level as a separate sub-negotiation, instead of the

[1]Here, 5 is the number of sub-negotiations, 3 is the number of issues, 100 is the (default) number of steps run for a scenario, 7 is the number of possible values per issue, and the additional 1 accounts for no agreement (e.g., None) being reached in that sub-negotiation.

outcome after a turn. This allows us to ignore the turns within a sub-negotiation, which would reduce the above to $(7^3)^5$ states, or simply the number of outcomes. We recursively calculate the expected value of these outcomes, and then use a temperature-adjusted softmax to heuristically evaluate the probabilities of certain outcomes being met in agreement. Second, we cut off our tree search at a certain level, and then use the evaluation of the `side_ufuns` at those outcomes as a heuristic. Note this heuristic is admissible. We then propagate those values up the normal way. To determine this level, we have an empirically estimated limit of 30 million on the number of evaluations we can make, and cut off when we can no explore another level.