LearningAgent

General

We are using the BOA architecture. BOA architecture consists of three distinct components:

1. Bidding strategy
2. Opponent model
3. Acceptance condition

In the follow sections we will elaborate on each component.

Bidding strategy

In general, at every step, we try to offer a bid that meets the following conditions:

- its value is above our threshold
- its value is above the opponent approximate threshold
- its value is pareto efficient

More precisely, we divide the negotiation into two phases:

1. first phase - in this phase we offer the first bid that its utility is above our threshold. We did not deal with the opponent's threshold and utility because we did not learn enough so that we could estimate it
2. second phase, we offered bid that maximize the opponent utility (we try to find the pareto efficient with heuristic approach that test random bids until a bid is found above a threshold. But instead of selecting only 1 random bid, search multiple bids and choose the one with the highest opponent utility) and his utility above our threshold and opponent threshold.

At every turn we check randomly 2k optional bids if no bid is good (in first phase his utility above our threshold. In the second phase his utility above our threshold and opponent threshold) for us, we offer the **optimalBid** (that is the bid with the highest utility for our agent). The **optimalBid** is calculated during the settings phase.

Close to the end of the negotiation (0.95 of the negotiation time has already passed), we check if the best bid offered by the opponent. If it's above our threshold we offer it. Otherwise, we increase the number (10k) of the bids that we check to find a good bid. if we have not found a good bid, we offer the optimal bid.

Opponent model

Opponent model consists of two parts:

- Model of opponent's utility function – its issue weights and values value.
- Model of opponent's threshold (in general, over time).

Model opponent's utility

For modeling this part, we are based on the frequency model. The frequency model assumes that other parties will use preferred issue values more often than less-preferred issue values. Thus, counting how often an opponent uses the issue values gives a way to assess these bids.

The frequency model modeled only the opponent value.

For modeling the issues value, we use the standard deviation of the values frequencies. the higher std of values in issue, the higher std of frequency of values in issue. The other factor affects std of frequency is the issue-weight. The higher weight for the issue, the higher standard deviation of the frequencies of this issue. This is because for issue with weight of 0.01, the opponent doesn't care if it gets 0.0098 (issue-value = 0.98) or 0.0023 (issue-value = 0.23) for this issue, but for issue with weight of 0.8, the opponent does care if it gets 0.72 (0.9) or 0.64 (0.8).

For calculation of the value of value j in issue i, we use the following formula:

$$V_j^{s_i} = \frac{f(V_j^{s_i})}{max_k f(V_k^{s_i})}$$

For calculation of the issue i weight's we use this formula:

$$s_i = \frac{std(V_i)}{\sum_k std(V_k) + \varepsilon}$$

Model opponent's threshold

After calculating the opponent's utility, we need to estimate its threshold.

We divide the time to two phases: In the first phase we learn the opponent frequencies. In the second phase, after we have enough data to calculate the opponent utility (by the formulas above), we estimate the opponent threshold.

We divide the second phase to 40. during the negotiation we have two array one for the opponent offers, for each offer we get, we save it in the corresponds timeslot. The other for our offers that the opponent rejects, for each offer that reject we saved it in the corresponds timeslot. When then negotiation ends, for the offers array we calculate the average of all the offers in a specific timeslot and remember that value (data from some rounds in weighted, so that later round is more important) and for reject array we calculate the maximum utility that the opponent reject in a specific timeslot and remember that value (data from some rounds in weighted, so that later round is more important)

When we meet a known opponent (an opponent with learned data) we smooth the average utility data and the reject data (using moving average with specific window size) and use that data to calculate the opponent threshold. and then for each timeslot we compare the utility of the opponent (as calculated by the extended frequency model) to the learned threshold that corresponds to the current timeslot.

The formula to calculate the threshold is:

$$Threshold(t) = \max(2 * avg\_utility_t - 1, \max\ \_reject_t)$$

The rational of this method is that the opponent always offers only offers whose profit is higher than the threshold. If we take the average of all his offers over time on the assumption that the expenditures are evenly distributed in the range, we can get a number that will testify out of hand. For example, if the threshold is 0.7 the opponent will only offer bids with utility above 0.7 so we will see an average around 0.85.

In addition, we have information about the maximum bid that the opponent rejected. From this information it is possible to deduce what the minimum threshold is.

For example, if we got an average of 0.85, we estimated the threshold at 0.7 and sent the opponent an offer worth 0.8, and he rejected it, we can conclude that his threshold greater than 0.8, so we choose 0.8 to be the threshold.

Acceptance condition

We use adaptive thresholds during the negotiation session.

We accept opponent offers if the utility of the offer is above our threshold. We also offer only bids that their utility is above the threshold.

Our threshold function is:

$$Threshold(t) = 0.95 - \frac{e^{at-1}}{e^{a-1}} * \left(0.95 - \ 0.55 * V_T \ - 0.4 * V_0 \ + 0.5 * std(V_T)\right)$$

The purpose of alpha is to start compromising after the opponent. For calculating alpha, we're looking for the point that the opponent agent offers utility start to decrease this means that he starts to compromise.

References:

1. GeniusWeb Tutorial
2. ANL2021 agents
3. Tim Baarslag & Mark J. C. Hendrikx & Koen V. Hindriks & Catholijn M. Jonker (2015) . Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques.