Moshe Hazoom. ID: 201337904
Guy Dubrovski. ID: 307188342
Slava Bronfman. ID: 305917601

Slava Agent

How to accept and make offers:

We define a variable called GOOD_ENOUGHT_UTILITY that stands for a threshold that if the opponent agent offers us a bid that has a utility higher than it, accept immediately. Otherwise, split into two cases: **Exploration** and **Exploitation**.

- At exploration part, update the maximal bid for us by searching over a large fixed number of iterations (we took the value of 10,000) on which we random different bids. Then, with probability of 0.5, offer it to the opponent and with probability of 0.5, offer to him a bid that is good enough for us randomly, i.e. one that has utility more than UTILITY_THRESHOLD (we took the value 0.95). In order to create a set of those bids (that good enough for us), we random a large number of bids of the domain and took only that has utility more than the threshold UTILITY_THRESHOLD.

- At exploitation part, offer always the best bid for us that we found. **Accept**, if the opponent offers a bid that has utility more than MIN_UTILITY_ACCEPT (we took the value 0.7) and has value higher or equals to all opponent its previous offers. Otherwise, offer the best bid for us.

Exploration defined to be 95% of the time and exploitation defined to be on the other 5% of the time.

*On exceptions in the code, offer the best bid for us.

How did the agent adapt to other agents' strategies:

We said above the way we make offers and we mentioned that with probability of 0.5 we offer some random bid that is good for us. With this kind of randomness, and not a strict policy, we make the opponent agent offers us different bids than before (hopefully) and that way we can maximize the bid with highest utility for us that the opponent agent offer to us. We encountered couple of negotiation sessions on which this randomness made the opponent agent offer a bid with utility that is more than GOOD_ENOUGHT_UTILITY and we accepted it and won.

How did the agent learn about the preference profile in the domain:

Our agent saves at the end of the negotiation (the method endSession) the best bid with highest utility we found from the domain. At the beginning of the

session, we try to load this value from the disk, and if exists, initialize the bid with the highest utility for us to be that value and not 0. That way, we can learn about the specific domain from session to session and improve agent is performance.