Chris Halverson
Jacob Rosenberg

Our implementation of the negotiator class represents a general bargaining strategy. It first offers its preferences in an attempt to get the maximum utility out of negotiations. Clearly, if the other negotiator accepts our preferences, then that is the best outcome for our agent. If the other agent does not accept our offer, real negotiating then begins. To generate our own offers, we enumerated all of the neighbor lists to our previous offer (i.e. offers that are generated by only moving one item one space in either direction) and picked the offer that gave us the highest utility. Throughout the rounds of negotiation, our agent keeps track of the utility of each offer the opposing agent sends. Treating the utility values as y-values and the round numbers as x-values, we calculated the "slope" of the line of best fit. We then compared this slope to the slope of a line that we considered to be representative of a fair bargaining process over the course of all of the rounds. We defined a fair bargain as one that gave roughly half of the maximum utility (or as close as possible) to each agent. We then use this comparison to determine if we believed the opposing agent to be a "fair" opponent. Upon receiving an offer each round, we checked to see if the offer was worth it. This involved calculating the percentage of our maximum utility that accepting the current enemy offer would give us. If this percentage was above a specific threshold, then our agent chose to accept the offer. To represent the idea that we wanted to try to get better deals in the earlier rounds of negotiation and then adjust what we considered 'acceptable' as the negotiation got closer and closer to ending, the threshold was initialized to be 90% of our maximum utility, and each round we decremented it by a specific amount calculated so that we would reach ~50% by the last round of negotiation. It was here that we decided to take advantage of results from past negotiations against the same opponent, by using the average of what percentage of our maximum utility we had actually earned over all previous *successful* negotiations. This number became the target for decrementing the threshold over further scenarios against the same opponent. In the final round of negotiation, we used our belief in whether or not the opposing agent had been 'fair' over the rounds so far to determine behavior. If we thought they had been so, we simply continued our behavior as before, offering one last chance to try to come to an acceptable deal for both parties. If however we determined that the opponent had been stubborn and had not conceded enough of their own utility, we threw out our own preferences as one last offer, thinking that since it was unlikely to come to a fair agreement in one more round when it had not been reached so far, we might as well take advantage of any agents that had been coded to accept any last offer, no matter how bad for them.

We tested against the random negotiator from the framework, one that only ever offered its preference list, and against itself. Against all these negotiators we saw our offers predictably moving towards what we thought of as a fair deal where we were getting about half of our maximum utility. As expected we would often succeed against the random negotiator, as it would randomly decide whether or not to accept. It was almost always the random negotiator accepting our offer, as opposed to the other way around, because for whatever reason all of the random offerings given to us, seemed to be of the same, high utility value for the enemy, and consistently were below our acceptance threshold. Similarly the 'preferences-always' negotiator would never accept a deal from us, and with the sample scenario given, the enemy's preferences never gave us enough utility for us to consider it acceptable. The most interesting behavior we saw was against ourselves, on both sides offers came down towards the middle point, and both 'slope' calculations from each agent showed the other agent being fair. Nevertheless, we never reached a successful deal with ourselves, because the way the utility function seems to work (as discussed on Piazza) it is very unlikely to stumble upon a deal that is positive in utility for both agents, and so even though from each specific agent's point of view, the deals it were offering were 'fair' on the other side, they would always give much lower (0 or negative) utilities for the other party, even though they were only giving the offering agent ~50%. If the utility function is changed in such a way that 'middlingly positive' offers for one party are usually *also* 'middlingly positive' for the other, we believe our agent should consistently come to rather even deals, and hopefully there is enough wiggle room in our per-enemy analytics to gain an edge in overall points over those negotiations.

With extreme uncertainty about a) how the other enemy was behaving and what they wanted and b) what the 'search space' of possible offers was like in terms of benefitting both parties simultaneously, we were forced to adopt our own concept of what 'fair' would be in the framework and attempted to build our agent around that. Our hope was that combining this with a few small tweaks intended to take advantage of weaker opponents and adjust statistically based on past performance would ultimately create a successful negotiator.