Artificial Intelligence Techniques

Negotiation Agent Design - Analysis Report

Group 4

4004868 Tung Phan
4409159 Francesco Corsini
1369326 Dirk Meijer

February 15, 2015

Contents

1 Introduction

To prepare the competing negotiation agents for the upcoming tournament, all compiled agents were publically made available. This enabled us to run tests against these other agents, generating data such as the the utility of the agent per session, the distances of the solutions to Pareto and Nash, social welfare and whether the agents reached an agreement or not. These results are invaluable to tweaking and optimizing our agent, which helps us prepare for the tournament.

Our testing approach will be elaborated in section 2, followed by the test results in section 3. Using these test results, we made impovements to our agent, which will be discussed in section 4, along with our reasons for those changes. Finally, we will do a quick test to verify that the changes we made actually improved the performance of our agent in section 5.

2 Testing approach

In order to produce meaningful test results, we first have to prepare a testing approach. Putting it into context, we have to choose the right negotiation profiles, the number of parties per negotiation and whether we use all the available parties.

We chose to keep using our domain for the testing, because we are the most familiar with our own profiles and scenarios. Our domain contains 3 distinct scenarios, which is elaborated in the previous agent report. We also discovered that using the other domains did not provide enough diversity in choices, resulting in almost the same results across all scenarios.

The amount of parties involved per session is set to 3, as more will only increase the amount of test results without providing actual additional useful data, and it further complicates the negotiation sessions due to the amount of parties needed to agree. We opted to leave the deadline at 180, as we see it as a reasonable deadline. For our own agent, the deadline will have minimal impact, as our behaviour scales along with the deadline. The only difference it will have is that our agent will concede quicker, because the deadline is nearer.

We ommitted group 9 from our testing, because group 9 was the only group producing NullPointerExceptions.

3 Test results

4 Improvements made to the agent

Given how the agent was programmed in first place, no big major code rework were made. This is because we already implemented the agent with a set of parameters, ready to be tweaked during this second iteration of the project. The process was to

see the test results, change a little bit the parameters, run the test again and try to improve the overall performance.

The parameters we mainly worked on were those regarding the acceptance of the bid.
The first one was the parameter that changed the shape of the threshold function for the acceptance of the bid. We started with p=5, that made the acceptance function almost linear, starting from our best utility point and ending in our reservation value(in this case 0). However, after intense testing, we discovered that p=25 is the best, making the function flat in the beginning of the negotiation(early stage, keep position) and steep in the end, making more concessions as the deadline approaches.

5 Final results

We are overall really satisfied with our results. The agent performs quite good for its own utility and the overall welfare.