Delft University of Technology

Artificial Intelligence Techniques
IN4010-12

Group 31: Assignment 1

Authors:
Ravi Autar
Jesse Hagenaars
Arthur Hovanesyan
Prerak Mody
Ombretta Strafforello

September 23, 2018

1 Introduction

Organizing a party together with someone else involves making many choices. Especially when your preferences do not have a lot of overlap with the other person, negotation quickly becomes a mess in which outcomes best for both can be hard to find. To prevent this, Group 31 was tasked with developing a negotiating agent that can do these negotiations automatically. This report documents the choices made by Group 31 in developing this agent.

Section 2 discusses the exploration of the party domain and compares the performance of various agents. After that, Section 3 gives a PEAS description of the environment and explains the BOA framework introduced by [1]. Furthermore, the choices made for the agent's acceptance strategy are explained in this section.

2 Preparation

This section analyzes the party domain in which our agent will be negotiating, and compares the performance of multiple prebuilt agents.

2.1 Party domain analysis

Table 1 lists the issues of the party domain, as well as their discrete values. Obviously, the preferences of the negotiating agents will not be aligned for these issues. Some might prefer a cheaper party with little outsourcing, while others might want to do as little as possible themselves. Because of this, the utility space will contain a *Pareto optimal frontier*, which is defined as the set of bids that is optimal for all agents, meaning that other bids are worse for at least one agent. The pink line in the upper right of Figure 1 depicts the Pareto optimal frontier for two *SimpleAgents* (random bidding strategy) with different preferences.

Table 1: Issues and values of the party domain.

Issue	**Value 1**	**Value 2**	**Value 3**	**Value 4**
Food	Chips and nuts	Finger-food	Handmade food	Catering
Drinks	Non-alcoholic	Beer only	Handmade cocktails	Catering
Location	Party tent	Your dorm	Party room	Ballroom
Invitations	Plain	Photo	Custom, handmade	Custom, printed
Music	MP3	DJ	Band	
Cleanup	Water and soap	Specialized materials	Special equipment	Hired help

2.2 Agent performance analysis

In the case of two *SimpleAgents* with different preferences negotiating against each other (depicted in Figure 1), we can see that the bids are mostly random and do not converge to any particular optimum. The agents stop bidding when both have received a relatively good offer from their opponent compared to previous bids. The Pareto frontier is not approached in this negotiation.

Figure 2 shows a negotiation session where both agents start at the most preferred bids. From there on, the *ConcederNegotiationParty* decreases (concedes) in utility almost every bid, as the green negotiation trace demonstrates. At the same time, the *BoulwareNegotiationParty* barely nudges (shown by the blue trace). An agreement is reached when the Boulware agent proposes a bid with a higher utility

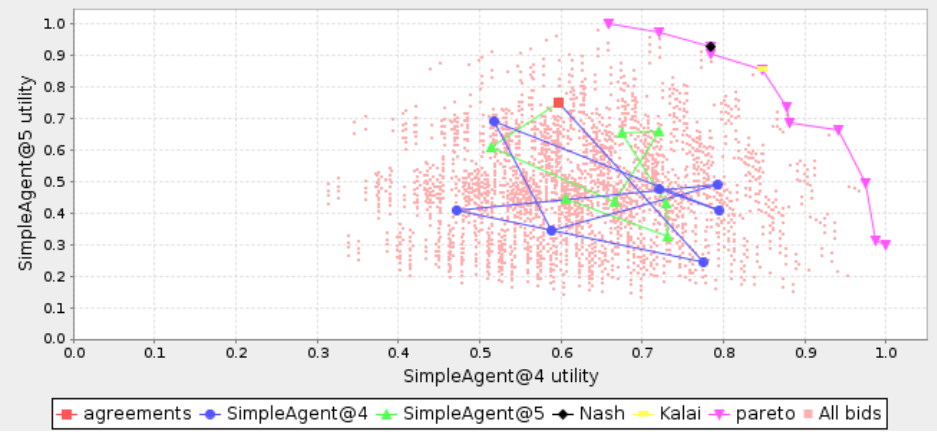

Figure 1: *SimpleAgent A* against *SimpleAgent B*.

than the Conceder agent's current bid. As can be seen in Figure 2, the reached agreement is on the Pareto frontier, which means that the outcome is optimal in some sense (albeit much more optimal for the Boulware agent than for the Conceder agent).

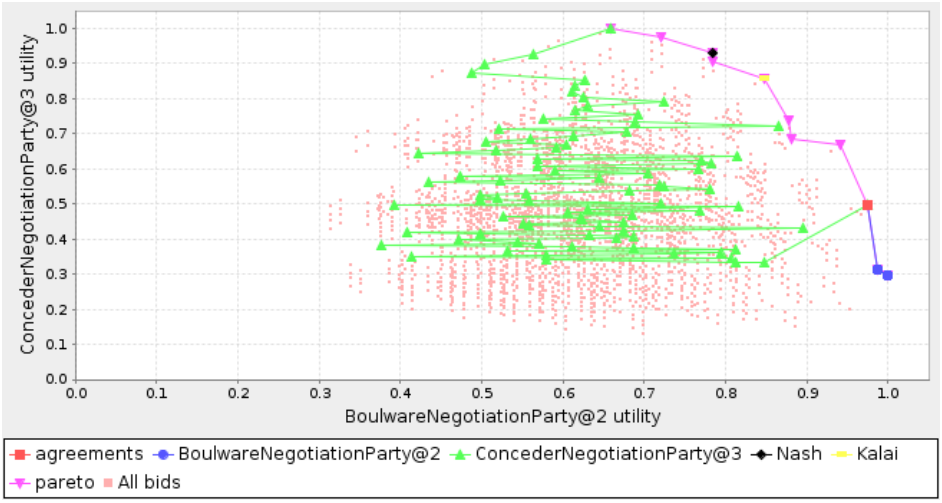

Figure 2: *BoulwareNegotiationParty* versus *ConcederNegotiationParty*.

3 Negotiation strategy

This section first describes the environment according to the PEAS principle, after which the BOA framework for negotiating agents is introduced. Finally, the choices made for the acceptance strategy of the party negotiation agent are explained.

3.1 PEAS description

In designing any agent, it is critical to correctly specify the task environment in which the agent will be operating, in order to prevent problems later in the development process [2]. Table 2 gives the *PEAS description* for the party negotiator.

Table 2: *PEAS description* of the task environment for the automated negotiator.

Agent Type	**Performance measures**	**Environment**	**Actuators**	**Sensors**
Party negotiator	Good food and drinks, nice location, many people attending, good music, all cleaned up after, fair outcome	Casual and relaxed, home-like, with a friend	Display or speakers for showing bidding and preference profile information	Keyboard or microphone for providing user preference profile and opponent bidding

- **Agent:** The agent will be negotiating with a friend who also recently graduated about a mutual graduation party.
- **Performance measures:** The performance of the agent is determined to which degree the agent can maximise the utility functions of the user (you). This means that the agent has to make sure that the selected food, drinks, location, invitations and cleanup styles are in line with the user's preferences as much as possible. Since our negotiation opponent is our friend, we also would like to have a fair outcome of the negotiation. Therefore, the agent should also take the (approximated) utility function of the opponent into consideration.
- **Environment:** Since the negotiation is about a party, it will most likely be really casual and flexible since the stakes are not that high.
- **Actuators:** For the agent to perform correctly it needs a way to place its bids either by showing it on screen or by the use of speakers (text to speech). It also needs a display to show the user (you) its currently selected preference profile for the various issues (as defined in Table 1).
- **Sensors:** The agent also needs a way of knowing what the preference profile of the user (you) is and also what the bids made by the opponent are. These can be provided through a keyboard interface, or by using a microphone for audio input. Note that the agent may also have more sophisticated types of sensors such as thermal infrared cameras, normal cameras etc. in order to understand its opponent better.

3.2 BOA framework

The *BOA framework* was initially developed to assess the performance of each of the components of a negotiating agent, and to allow for reusing these components to develop better negotiation strategies [1]. The components of the BOA framework are [3]:

- **Offering/bidding strategy:** Maps a negotiation trace (history of bids) to a bid or a set of possible bids (in which case it consults with the opponent model in order to select a bid).
- **Opponent model:** A learning algorithm that constructs a model of the opponent's preference profile.
- **Opponent model strategy:** Specifies how the opponent model is used to select a counter bid, and whether the opponent model may be updated in a specific turn.
- **Acceptance strategy:** Determines whether the opponent's bid should be accepted, or that a counter bid should be made. Can also decide to abort the negotiation altogether!

Figure 3 shows the interaction between the various components during negotiation. When a bid is received, the *bidding history* and *opponent model* are updated (in case this is allowed by the *opponent model strategy* for this turn). This ensures the known information about the opponent and the environment is maximized [1]. Next, based on the received bid, the *offering strategy* generates a set of bids that are equally preferred. Using the *opponent model strategy*, a counter bid is selected from this set. Finally, the *acceptance strategy* either accepts or declines the opponent's bid (in which case the counter bid is offered).

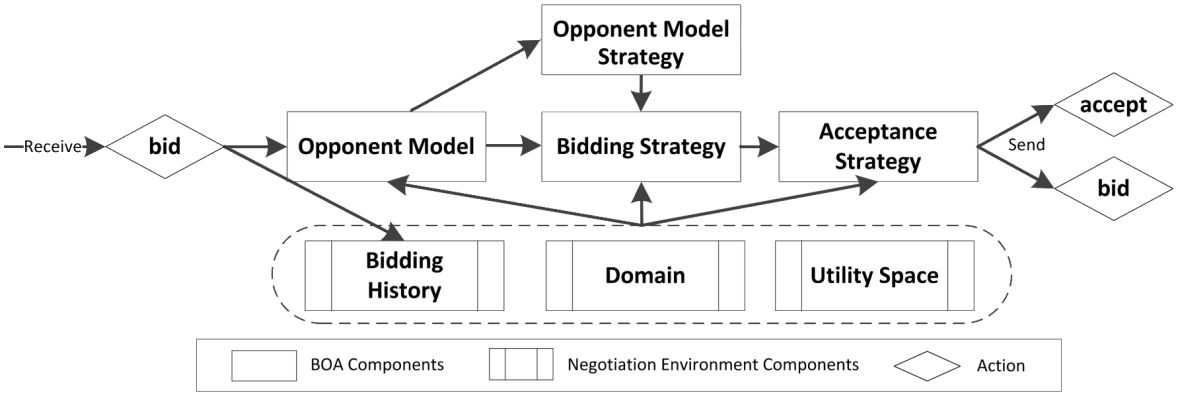

Figure 3: The negotiation flow for a BOA negotiating agent [3].

3.3 Acceptance strategy

The acceptance strategy adopted by the party negotiation agent is inspired by [4]. The strategy involves taking into account previous bids made by the opponent within a fixed time frame. The agent then calculates the discounted utility (which depends on remaining time) of the current bid made by the opponent and accepts if it is greater than the average utility calculated in the look-back window. In order to ensure that the agent accepts using the aforementioned condition only if there are enough bids made by the opponent, the agent will always reject the first k bids made by the opponent, where k is the size of the look-back window.

Table 3: BOA components of our agent.

Component	**Bidding strategy**	**Opponent model**	**Opponent model strategy**	**Acceptance strategy**
Choice	2010 - AgentK	AgentLG model	NTFT	Group31 AS

Table 3 shows the other components that make up our agent (these were just the defaults suggested by Genius). Figure 4 shows a negotiation session in the party domain between our agent (in blue) and a Bayesian agent (in green). From this figure, we can spot some desirable characteristics:

- Our agent stays close to the Pareto frontier, which gives it a relatively high chance of achieving an agreement that is Pareto-optimal.
- Our agent is not easily lured by the opponents *nice* bids, and instead responds with nice bids of its own.

- In the end, a very fair and optimal outcome is reached, which is exactly what we want when negotiating with a friend!

Of course, further investigation is needed to check whether these positive characteristics are a result of the acceptance strategy or of the other components. Also, the robustness of the agent will have to be checked in domains other than the party domain.

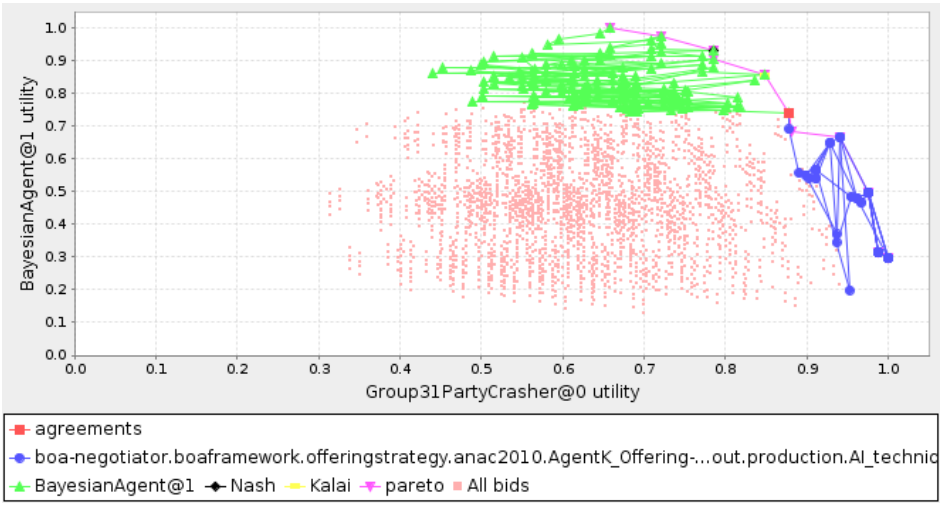

Figure 4: Our agent (blue) against a Bayesian agent (green) in the party domain.

References

[1] T. Baarslag, K. Hindriks, M. Hendrikx, A. Dirkzwager, and C. Jonker, "Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies," in *Novel Insights in Agent-based Complex Automated Negotiation*, pp. 61–83, Springer, 2014.

[2] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Malaysia; Pearson Education Limited,, 2016.

[3] T. Baarslag, W. Pasman, K. Hindriks, and D. Tykhonov, *Using the GENIUS Framework for Running Autonomous Negotiating Parties*, September 2018.

[4] T. Baarslag, "What to Bid and When to Stop," TU Delft, 2014.