Automated Negotiation League (ANL) 2021

Join us on Discord!

1 Challenge

Design a negotiation agent for bilateral negotiation that can learn from every previous encounters while the tournament progresses. The agent that obtained the highest average score (utility) wins.

In previous years of ANL, either no learning across negotiation session was allowed, or learning options where limited. e.g. Learning over identically repeated negotiation sessions. This year we aim to provide as much flexibility as possible while safeguarding for unfair play. At every negotiation, agents obtain a file path where they can save anything they want. An agent gets a chance to process all that has been saved into a persistent state for that agent at multiple occasions during the tournament. This state is stored as a file on disk and the path is passed to the agent every time a negotiation is initiated.

The competition takes place during IJCAI 2021, August 2021, in Montreal, Canada. There will be an estimated $5000 (subject to change) in total available for prize money and for student travel grants[1] which will be made available to participants of ANAC, of which $600 will be given away to ANL participants. This prize money will be distributed as follows:

First place	$300
Second place	$200
Third place	$100

Table 1: Prize money (For evaluation criteria see subsection 2.2)

2 Preliminaries

A negotiation setup consists of a set of rules (protocol) to abide by, an opponent to negotiate against, and a problem (scenario) to negotiate over. We describe all three components in this section.

Negotiation Protocol. The Stacked Alternating Offers Protocol [1] (SOAP) is used as negotiation rules Here, the starting agent has to make an opening

[1] Since IJCAI 2021 will be fully online, the travel grants will not be available this year.

offer after which the agents take turns while performing one of 3 actions:

1. Accept the offer of the other agent
2. Make a counter offer
3. Walk away

To prevent agents from negotiating indefinitely, a deadline of 60 seconds is set.

Opponents. The opponents will be agents that are submitted by other competitors in the ANL.

Scenario. The scenario consist of a discrete outcome space (or domain) $\omega \in \Omega$ and a preference profile per agent. This preference profile is used as a utility function $u(\omega)$ to map the problem space to a utility value for that agent (Equation 1). Here, 0 and 1 are the worst and best utility respectively that can be obtained by the agent. The negotiations are performed under incomplete information, meaning that the utility function of the opponent is unknown.

$$u : \Omega \rightarrow [0, 1] \tag{1}$$

2.1 Platform

Entrants to the competition have to develop and submit an autonomous negotiating party in Java that runs on GeniusWeb [2]. GeniusWeb is a negotiation platform in which you can develop general negotiating parties as well as create negotiation domains and preference profiles. The platform allows you to simulate negotiation sessions and run tournaments. Extensive references are available in section 5 where we provide links to additional information. A basic example agent that can handle the current challenge can be found on GitHub. We aim to provide a quick-start and FAQ on this GitHub page, which we will improve incrementally based on feedback received via e.g. Discord).

2.2 Evaluation

If the agents reach an agreement, then the utility of that outcome is the score for an agent. This utility usually different for both agents. A utility of 0 is obtained if no agreement is reached. This can occur for one of three reasons:

1. One of the agents walks away
2. The deadline is reached
3. One of the agents crashes

At the end of the tournament, the average utility of every agent is calculated. The agent with the highest average utility wins the competition. We will also award a special mention to the competitor that achieves the highest social welfare, which is the joint utility pay-off.

2.3 Rules of encounter

- Agents need to follow the SAOP protocol.
- Opponents are encountered multiple times.
- A single scenario is negotiated over **once** with every opponent.
- In total 50 different scenarios will be played for the competition. These scenarios will all be fully discrete (so no continuous issues).
- Negotiation data and persistent state can only be saved at the path that is provided to the agent.
- Violating the spirit of fair play will result in a disqualification. The ANAC board will be the judge in these matters.
- The competition rules allow multiple entries from a single institution, but require each agent to be developed independently.
- No participant can be a co-author of more than 3 agents.
- The source code of agents must be submitted. This code will be included in the GeniusWeb platform after the competition has finished for future use.

3 Step-by-step learning

We provide a step-by-step description of the tournament from a single agents (and thus the competitors) perspective illustrating the learning behaviour of the agent. The identity of an opponent is disclosed during this competition, so that your can learn the behaviour of every agent in the competition. There are two separate steps that we consider: Negotiation and learning.

Negotiation. See Figure 1. The agent is initiated with session information that contains details on the negotiation setup. It also includes a file path to the persistent state of the agent that serves as the memory of the agent and can be loaded. The agent is also provided with a file path where data of the current negotiation session can be written to.

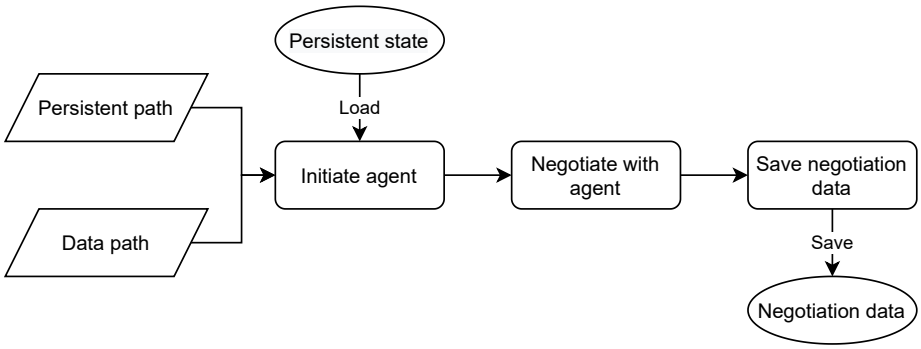

Figure 1: Negotiation step

Learning. See Figure 2. The agent is again initiated with session information that is needed to perform an update to the persistent state. Most notably, a list of file paths is provided that point to data of past negotiations that have not yet been processed in the persistent state. The file path to the persistent state of the agent is also provided. The agent can now load all the unprocessed data and use it to update its persistent state. For this step a deadline of 60 seconds is set after which the agent is killed.

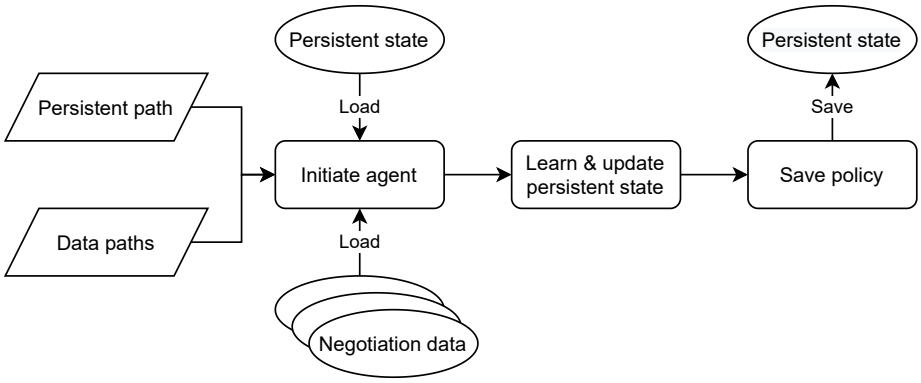

Figure 2: Learning step

Tournament overview. An overview of the total tournament structure is provided in Figure 3.

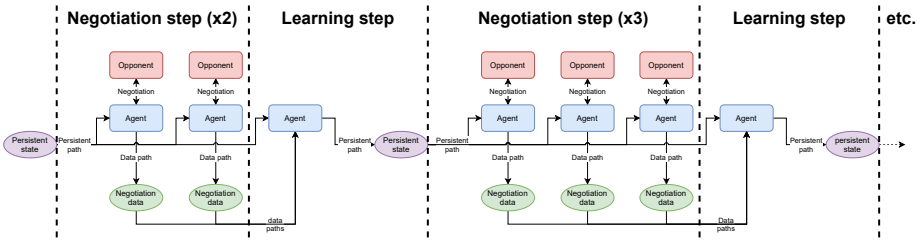

Figure 3: Learning step

4 Submission & questions

Participants submit their agent source code, jar file (with dependencies), and academic report (optionally) in a zipped folder. Please submit your application though the following link:

Submission Form

4.1 Academic report

Each participant has the option to prepare a 2-4 page report describing the design of their agent according to academic standards. The best teams that submit a report will be given the opportunity to give a brief presentation describing their agent at IJCAI 2021.

Furthermore, proceedings of the competition are planned to be published in a special issue. The report will be evaluated by the organisers of this league. For eligibility, the design should provide a contribution to the negotiation community. The report should address the following aspects:

- Bidding Strategy: How the agent generates bid at its each turn.
- Acceptance Strategy: How the agent decides to accept or reject a given bid.
- Opponent Modelling: How the agent models the opponent (e.g. the opponent's strategy, preferences etc.).
- Learning method: What does the agent learn and why? Show the influence of the learning behaviour of the agent.

5 Fact sheet

Important dates

Submission deadline	4th of August
Notification to finalists	18th of August
Event	21st-26th of August 2021

Important links

Making an agent for ANAC 2021
Discord Server
Competition Website
Submission Form
Getting started with GeniusWeb
Technical information GeniusWeb

Contact

Bram Renting	Main contact for questions about the challenge
Wouter Pasman	For questions about implementation and GeniusWeb

Advisers (alphabetic)

Name	Title	Affiliation
Reyhan Aydoğan	Assistant Professor	Ozyegin University
Tim Baarslag	Scientific Staff Member	Centrum Wiskunde & Informatica
Katsuhide Fujita	Associate Professor	Tokyo University of Agriculture and Technology
Holger Hoos	Professor	Leiden University
Catholijn Jonker	Professor	Delft University of Technology

References

[1] R. Aydoğan, D. Festen, K. V. Hindriks, and C. M. Jonker, "Alternating offers protocols for multilateral negotiation," in *Studies in Computational Intelligence*, vol. 674, Springer, 2017, pp. 153–167, ISBN: 978-3-319-51563-2. DOI: 10.1007/978-3-319-51563-2_10.

[2] R. Lin, S. Kraus, T. Baarslag, D. Tykhonov, K. Hindriks, and C. M. Jonker, "Genius: An integrated environment for supporting the design of generic automated negotiators," *Computational Intelligence*, vol. 30, no. 1, pp. 48–70, 2014, ISSN: 08247935. DOI: 10.1111/j.1467-8640.2012.00463.x.