

The Automated Negotiating Agents Competition

Automated Negotiation League

Challenge for ANAC 2020

Representing users in a negotiation: developing an agent that can negotiate while performing preference elicitation

The challenge for 2020 is to design a negotiating agent that can elicit preference information from a user during the negotiation. The idea is that when a negotiating agent represents a user in a negotiation, it does not know exactly what the user wants, and therefore the agent needs to actively improve its user model through a preference elicitation process.

In previous years of ANAC, and in most literature on automated negotiation, the utility function of the agent is presumed known. As was introduced for the first time in 2019, the preferences of the agent will instead be given in the form of a *ranking* of a limited number of possible agreements. Additionally, this year, the agent may elicit further information about the ranking against an elicitation cost in order to improve the user model.

You can register your intention to participate [here](#) and be added to our mailing list.

Website: <http://web.tuat.ac.jp/~katfuji/ANAC2020/genius.html>

COVID-19 update - New submission deadline: 14 July

ANAC 2020 will go ahead, with more than 80 registered participants so far. ANAC will be held in conjunction with IJCAI 2020, which will be held in January 2021. The deadline for submitting agents to each ANAC league is postponed until 14th July, 2020.

The winners of each league will be announced in January 2021 during IJCAI - either live or digitally. Finalists are announced in August 2020 (tentative). A select number of finalists will be invited to publish their agent descriptions in the Journal Knowledge Engineering Review. We aim to provide student support to travel to IJCAI.

Event

The competition takes place during [IJCAI 2020](#), January 2021, in Kyoto, Japan. There is at least \$5000 in total available for prize money and for student travel grants which will be made available to participants.

Entrants

Entrants to the competition have to develop and submit an autonomous negotiating party that runs on [GeniusWeb](#). 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. Example parties and tutorials are available. If you are familiar with the Genius framework, it is easy to develop your agent in this new platform.

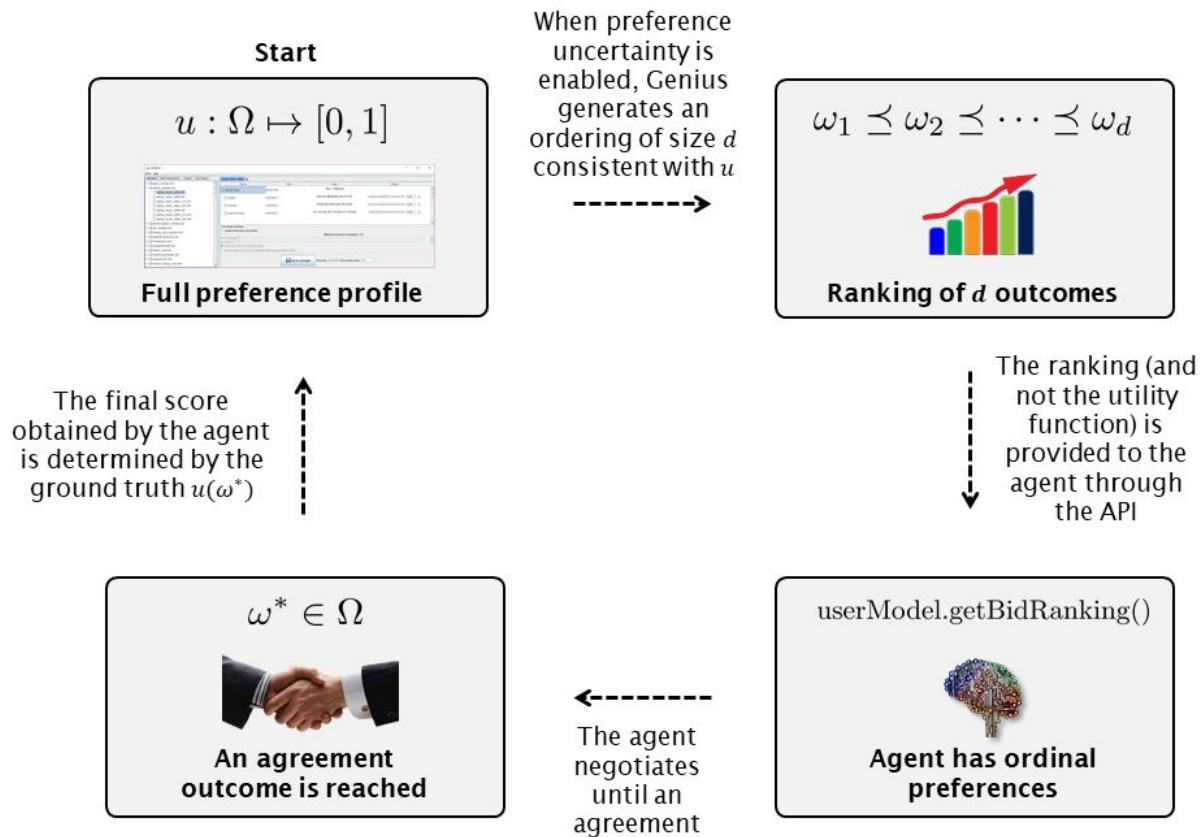
Performance of the parties will be evaluated in a tournament between all participants, where each party is matched with other submitted parties. All submitted parties will negotiate in a number of negotiation scenarios, with varying levels of preference uncertainty and elicitation costs.

We recommend developing a Java based party but you can also use Python 2 using the python adapter available with GeniusWeb.

Negotiating

Rather than receiving a utility function, the party will receive an initial [partial ordered profile](#). This profile will be an ordering of d bids, picked from an existing linear additive utility function U (see Figure). To help the party, the partial profile will contain the best and worst possible bid.

The party negotiates with the partial profile until a certain outcome w^* is reached. The score the party will receive for this outcome will be based on the original utility $U(w^*)$. So in short, the party receives ordinal information only, but it will be evaluated based on the underlying cardinal ground-truth.



The challenge for the parties is to design preference elicitation techniques in order to reach the best negotiation outcome, using techniques from e.g. machine learning, regression techniques, trade-offs, and linear programming. For details, please refer to the Genius manual and the [frequently asked questions](#).

Elicitation

The negotiations will be run using the [SHAOP protocol](#). In this protocol, the party can elicit more information about the true utility space in order to improve its own partial profile. This is done by requesting/eliciting a comparison from the "User". However, every such request comes at an elicitation cost, and all calls add up to a total bother. At the end of the negotiation, the performance of the party is recorded as the User utility of the final agreement, lowered with the total bother.

An elicitation action in the SHAOP protocol is implemented through the ElicitComparison and Compare actions. The ElicitComparison requests a bid to be compared to a set of alternative bids. The Comparison action tells which bids are better and worse than the given bid. The SHAOP and COB party can take these actions at any time, and need not follow the turntaking of the SHAOP protocol.

The SHAOP party knows the cost of each CompareBids request: it is contained in a parameter "elicitationcost" (or if not set, the default is 0.1). Each time CompareBids is called by a party, this cost is added to the total bother.

The SHAOP party will have to incorporate the Comparison result into the partial profile. One way to do this is to keep an ordered list, check the `simpleshaop` example party for details and example code.

Note: As of right now, the total bother is only implemented as constant increments of the elicitation cost. In theory however, it could be a very different function. All such possibilities are what we call bother cost functions. Future developments will explore extensions to support different types of bother cost functions.

Rules of Encounter

Negotiations are bilateral encounters following the SHAOP protocol. Offers are exchanged using a round-based protocol, with a maximum of 200 rounds. Parties do not have any prior knowledge about the preferences and strategy of the opponent. Parties are reset after each encounter; that is, agents may negotiate repeatedly on the same domain or with the same opponents, but they cannot learn from their previous interactions. When no agreement is struck, both parties receive the utility of their reservation bid, which is contained in the received profile. The reservation value can be different for both players. Note that this means a break-off can be preferable to an agreement for one (or both) of the players. This implies that it is risky to wait until the deadline to reach an agreement.

SHAOP specifies that the parties take turns making offers. One party starts the negotiation with an opening bid, after which the other party can take the following actions:

1. Make a counter offer (thus rejecting and overriding the previous offer);
2. Accept the offer;
3. Walk away (i.e. ending the negotiation without any agreement and receiving the reservation value).

This process ends when either a party walks away, a party does an incorrect action, an agreement is reached or the deadline is reached (e.g. 200 rounds).

In order to make running the tournament feasible, parties are expected to generate their offers in a reasonable amount of time. Currently the GeniusWeb runserver imposes a response timeout of 10 seconds during negotiations.

Fair play

Agents will be disqualified for violating the spirit of fair play (e.g. hacking the API, starting threads, attempts to access other party's preference profile). The board of the ANAC 2020 competition will be the judge on these matters (for more information, see <http://ii.tudelft.nl/anac/>). The competition rules allow multiple entries from a single institution, but require each agent to be developed independently.

Evaluation

The winners will be determined by the **average individual utilities** gained by each agent. There will be a separate prize for the **most innovative elicitation strategy**. The teams of the top agents will be notified, and the final results and awards will be announced at IJCAI 2020. It is expected that teams that make it through to the finals will have a representative attending the conference.

Submission (Deadline: 14 July, 2020)

Participants submit their agent source code and class files (in a .zip or .jar package). Please submit your application through the following link:

<https://tinyurl.com/GENIUSANAC2020>

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. 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 strategy design should provide a contribution to the negotiation community. The report is recommended to 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.);
- **Elicitation Method:** how the agent deals with preference uncertainty, how the agent performs preference elicitation, and explaining which heuristics and/or machine learning method it employs for this purpose;
- **Evaluation:** an evaluation of the agent (either against itself or in a small-scale tournament setting).

Important Dates

Submission deadline:	July 14, 2020
Notification to finalists:	August 15, 2020 (tentative)
Event:	January 2021

Questions and Answers

- Website: <http://web.tuat.ac.jp/~katfuji/ANAC2020/genius.html>
- Getting started: <https://ii.tudelft.nl/GeniusWeb/students.html>
- Mailing list registration through the [registration form](#).
- Feel free to consult the [FAQ](#).

- Send your questions to:
 - T.Baarslag@cw.nl (main contact for questions about the challenge)
 - W.Pasman@tudelft.nl (for questions about implementation and GeniusWeb)
 - reyhan.aydogan@ozyegin.edu.tr
 - katfuji@cc.tuat.ac.jp