win.win | a negotiating agent for mutual benefit

JOCHEM BROUWER, SAUL GEBHARDT, WOUTER VAN HARTSKAMP, and OTTO MÄTTAS, Utrecht University, The Netherlands

ACM Reference Format:
Jochem Brouwer, Saul Gebhardt, Wouter van Hartskamp, and Otto Mättas. 2021. win.win | a negotiating agent for mutual benefit. 1, 1 (March 2021), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 PEAS MODEL

1.1 Performance measure

When creating a new agent, the first step is to determine what the performance measure is to which the newly created agent should aspire. [Russell and Norvig 2009, Ch. 2.3.1] The agent that will be created is a negotiation agent, so the most obvious performance measure should be the utility that is received after a negotiation.
One problem with the received utility as a performance measure however, is that the received utility differs from the domain that negotiations were made in and the agent that was negotiated with. Therefore, the agent can be evaluated in a couple of different metrics.

1.1.1 Creating a test environment. The agent will be competing in an ANAC(Automated Negotiating Agents Competition)-like tournament. One of the lessons learned from these tournaments is that there exists no single strategy that outperforms all other strategies over all possible negotiation scenarios. [Baarslag et al. 2015]
The domain that the tournament will be run in, is unknown. Therefore, there does not exist a single domain which can be used to get an indication of the agents' performance in the tournament.
Therefore, it would be best to set up a test environment which would test the created agent against already existing agents on a couple of different domains. These domains would ideally be somewhat varied in size. That way testing the agent in the test environment will give the most diverse and general results.
The agents that would be chosen to be part of the test environment should also be varied in performance. The agent should be capable of performing well against good agents (agents that participated with ANAC), but also should be able to perform well against more simple agents, who usually do not perform well against more advanced agents. Furthermore, the agent should be tested against both cooperative and non-cooperative agents.
A test environment with these properties should be considered a good enough test environment to test an agent on. With this, we can define the performance measures of said agents.

1.1.2 Utility. The first performance measure that should be considered is the utility score. The utility score can be used as a performance measure by taking the average utility that any agent has received by running it in a tournament in the test environment. The previously mentioned issues with taking the utility score as a performance measure, do not apply in this case, since the agent negotiates with a set amount of agents (which are the same in every tournament) and over a variety of different domains (which are also the same in every tournament), there is no longer a problem with using the average utility as a performance measure.

1.1.3 Quickness. In a time-based negotiation, (as opposed to round-based negotiations) both parties can benefit a lot if the agents are able to accept and create bids quickly, since both parties are able to consider and create more bids than whenever both agents take a lot of time to consider bids.
Another way of considering quickness is the time to reach an agreement as a percentage of the total time allowed in a negotiation session.

1.1.4 Percentage of successful negotiations. Another very important performance measure is the percentage of successful negotiations in the test environment. Since the agent will be competing in the final tournament with all kinds of different agents, the agent should be able to reach agreements with all kinds of agents.

1.1.5 Alternative possible performance measures. There are two final possible performance measures that could be considered; the average distance from the Pareto frontier and the average distance from the Nash point. These two performance measures could be used in a way to determine if the agent gets to fair agreements, which is an indication whether the agent is cooperative or non-cooperative.

1.2 Environment

The agent is designed to be able to negotiate in a multi-agent environment. The goal of interaction is to reach a mutually acceptable agreement between two (or more) agents which all have conflicting interest and a desire to cooperate to reach an outcome. For simplicity, only entities bidding against each other and for a mutual deal are considered agents. For generalisation purposes, the notion of objects can still be considered while the agents will not be interacting with such objects or additional entities. Also, the agent has to be able to interact through different domains and scenarios with generalised approach.

1.2.1 Competitiveness. As the tournament environment will never be a zero-sum game, it can be considered a cooperative environment even when certain competitive aspects are present in the form of

Authors' address: Jochem Brouwer, j.brouwer8@students.uu.nl; Saul Gebhardt, s.a.gebhardt@students.uu.nl; Wouter van Hartskamp, w.vanhartskamp@students.uu.nl; Otto Mättas, o.mattas@students.uu.nl, Utrecht University, P.O. Box 80125, Utrecht, Utrecht, The Netherlands, 3508 TC.

XXXX-XXXX/2021/3-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

different individual interest. In considered scenarios, agents' behaviour is best described as maximising a performance measure whose value depends on the other agents' behavior. For distinction, other agent(s) are furthermore deemed opponents.

1.2.2 Observability. Opponent's actions are known through bids which advances the word state for the agent. Communication emerges as rational behaviour. Even randomised action can be considered rational as it avoids the pitfalls of predictability. This adds to the agent's success certainty of achieving its goal. At the same time, this makes the environment only partially observable. Additionally, the negotiation session's length varies and is not predetermined. Overall, this makes the environment non-deterministic.

1.2.3 Adaptiveness. The environment can not change while an agent is deliberating, thus rendering the environment static for the agent. World state change is brought about in steps as a result of interaction between the agent and its opponent(s). There is no need for the agent to keep track of the state outside of the interaction. While every interaction is discrete in its own right, the world itself is continuous as the bids are not predetermined. As explained before, they can be rational even when randomised.

1.2.4 Summary. Some is known about the environment. For example, the fact that agent has an incentive to cooperate with its opponents as this is an inherent feature for all the parties in a negotiation session. On the other hand, much is left unknown. For example the exact way opponents interact during bidding, their strategy for accepting bids and making them. To conclude, we gather all the properties of the environment. We can say it is *partially observable, cooperative, non-deterministic, !sequential/episodic, static, continuous and unknown multi-agent environment.*

1.3 Actuators

In this model, the agents both can offer different strategies to the other agent. The other agent should act upon this strategy: the agent can accept the bid, or it can send another offer.

1.4 Sensors

Where an agent is interacting with it's environment with actuators, the agent perceives it's environment through its sensors. The sensors are the list of input devices the agent has, quite literally the eyes and ears of the agent. With this input, the agent decides on a course of action out of the actuators and handles accordingly. Examples Our example What to do with this input for actuators Mathematic shiezzle

2 BOA COMPONENTS

The BOA framework was chosen due to the fact that the agent is supposed to be a cooperative agent. This means that the bidding strategy should be dynamic, and that the agent attempts to find the best solution for both agents (originating the name win.win). Ideally, the agent also manages to get good results against non-cooperative agents. A high level description of the agent follows.

At the start of the negotiation, the agent creates a bid array of all possible bids, ordered by utility. If the domain is too large, the bid array will consist of a sample of bids in the domain. The first bid that the agent makes is a bid with a utility at 80% of the maximum utility that is possible to achieve. This is a good starting point, since it might fool the other agent in thinking that this is the maximum achievable utility for the win.win agent.

When win.win receives a bid from the opponent, the bid is registered. With the bids that the opponent makes, win.win attempts to create an opponent model. The type of opponent modelling that win.win uses is a modification of a Naive Bayes classifier, called a weighted Naive Bayes classifier. [Frank et al. 2012] The idea is that if a specific value is included in a lot of bids by the agent, this value is considered more important by the other agent than other values.

With this information win.win is able to create a more accurate representation of the opponents' preferences, and is able to use this information to find an agreement with mutual benefit.

The acceptance strategy of win.win is quite simple. In general, win.win accepts a bid if its opponent has just sent a bid with a higher utility for win.win than the utility of the bid win.win is about to send to the other agent.

Win.win also accepts any bid received with a utility higher than some predetermined value. This predetermined value should be a high enough utility such that if an agreement is reached through this acceptance condition, the agreement should be good enough for win.win. Finally, after some time instance t has passed, win.win will accept any bid made by the opponent, since win.win always prefers to reach an agreement.

In the following subsections, the components of the BOA model are discussed in a bit more detail. The current design is based on theoretical research and the aforementioned performance measures. If it turns out that the strategy is not as good, some small changes might be made.

2.1 Bidding Strategy

The first concept to be explained in the BOA model is the bidding strategy. This bidding strategy is defined as: "[...] Bidding strategy is a mapping which maps a negotiation trace to a bid." [Baarslag et al. 2014]. In other words, the bidding strategy is a set of choices in response to a bid by the other agent. This is a relatively new view as earlier works viewed bidding strategy as integral to the acceptance strategy [Hindriks and Tykhonov 2008]. Decoupling the Bidding strategy from the acceptance strategy increases the ease of categorization and automated performance testing of multiple strategies.

2.1.1 Importance. Bidding strategy is seen as one of the most important parts of the BOA model due to the many unanswered questions and concerns the strategy needs to keep into account when bidding [Baarslag 2014]. All things from revealing preferences to estimating the known unknowns of the opponent is to be done in a split second by the AI. This is especially difficult considering the immense number of strategies the opponent can employ. However, there are possible simplifications to be made to help analyse these strategies more effectively.

2.1.2 Types. Baarslag et al. defines at least four main types of baseline bidding strategies: time dependent, resource dependent, behavior dependent, and zero intelligence strategies [Baarslag et al. 2014].

These types are only loosely defined and can overlap significantly. A strategy can be both based on the time of the negotiation and the saved behavior of the opponent.

2.2 Opponent Model

First of all, the opponent's latest bid is collected and recorded. As explained above, it is used for informing the agent's bidding strategy. Furthermore, it is used for creating a representation of the opponent's bidding and acceptance strategies. According to the BOA framework, this representation can be called an opponent model as explained by [Baarslag et al. 2014]. This would help the agent approach opponents dynamically, also respond appropriately and intelligently to many different opponents. As an added benefit, modelling the opponent's strategies in a modular fashion allows the researchers to iterate quickly and asynchronously on different parts of the agent.

2.2.1 Importance. Frequency counting is used to estimate importance of bid values for the opponent. More specifically, a locally weighted Naive Bayes algorithm is leveraged to relax the independence assumption by learning local models at prediction time. As soon as the opponent makes a bid, the agent updates the model.[Frank et al. 2012] The researchers are also keen on comparing the Naive Bayes algorithm with Gaussian Process preference learning framework proposed by [Leahu et al. 2019].

2.2.2 Weighted Naive Bayes. For Naive Bayes, weighting is regularly used to place more emphasis on highly predictive attributes than those that are less predictive. In essence, this might prove risky in a simple negotiation session where the opponent has an aim to confuse the agent with alternating its preferences - by seemingly randomising the bid importance. This warrants the researchers to make an effort to investigate for a solution that selects weights to minimise either the negative conditional log likelihood or the mean squared error objective functions, also contributing to attribute independence as explained by [Zaidi et al. 2013].

2.2.3 Conclusion. To properly express the opponent's bid and its importance, the agent keeps track of the opponent's bid minimum and maximum threshold values in real time. The agent stores and updates mapped values as time progresses. As a result, the agent can determine which bids would be deemed acceptable by the opponent. To conclude, this helps the agent decide whether or not to accept the latest bid received. If not, then which is the appropriate bid to offer to the opponent to continue the negotiation.

2.3 Acceptance Strategy

This section gives a high-level review of the acceptance strategies from [Baarslag et al. 2013]. There is one condition which is changed.

Definitions.

$$x^t_{B\to A}$$

This represents a bid from agent B to agent A at time t. There is an utility function $U_A(x)$ which represents the utility of the bid x for agent A.

Acceptance strategies. When agent B bids at time t and the bid is higher than the counter-offer agent A is about to send out at time t', accept the bid:

$$\mathbf{AC}_A(t', x^{t'}_{A\to B}) \iff U_a(x^t_{B\to A}) \geq U_A(x^{t'}_{A\to B})$$

There is also a generalized version of this acceptance strategy, which scales the bid the agent B sends to A and has a minimum utility gap β which should be fulfilled:

$$\mathbf{AC}_{next}(\alpha, \beta) \iff \alpha \cdot U_A(x^t_{B\to A}) + \beta \geq U_A(x^{t_{n-1}}_{A\to B})$$

For instance, $AC_{next}(1.02, 0.005)$ implies that the bid will get accepted if agent A sends a bid 2% higher than the agent B just sent, plus a minimum gap of 0.005.

The constant acceptance condition accepts if the agent B send a bid to agent A which is higher to some constant α:

$$\mathbf{AC}_{const} \iff U_A(x^t_{B\to A}) \geq \alpha$$

The final acceptance condition depends upon the time: if the current time is later than some time T, then accept the bid:

$$\mathbf{AC}_{time}(T) \iff t' \geq T$$

These acceptance can be combined to the *combined acceptance condition*:

$$\mathbf{AC}_{combined}(T, \alpha) \iff \mathbf{AC}_{next} \lor \mathbf{AC}_{time}(T) \land (U_A(x^t_{B\to A} \geq \alpha)$$

That is, accept the bid if the *next* acceptance condition is satisfied, or we are at some time t after the "deadline" time T and the bid is higher than or equal to α

Extensions. Here the acceptance condition is extended. The negotiation is split up in three phases: the initial phase, the mid-phase and the deadline-phase. In the initial phase, there only is a constant bid which is accepted. The idea here is that if the bid is higher to some (high) value, this bid is immediately accepted. In table 4 of [Baarslag et al. 2013] it can be seen that the highest average utility value of the agent is 0.675. It thus makes sense to immediately accept a bid which is higher than this constant. After the initial phase T_i the condition is switched to the acceptance condition as described above.

Formally:

$$\mathbf{AC}_{combined-init}(T_i, T, \alpha_i, \alpha) \iff (\mathbf{AC}_{const}(\alpha_i) \land \neg\mathbf{AC}_{time}(T_i)) \lor (\mathbf{AC}_{combined}(T, \alpha) \land \mathbf{AC}_{time}(T))$$

REFERENCES

Tim Baarslag. 2014. *What to bid and when to stop.* Ph.D. Dissertation. Delft University of Technology. https://doi.org/10.4233/uuid:3df6e234-a7c1-4dbe-9eb9-baadabc04bca

Tim Baarslag, Reyhan Aydoğan, Koen V Hindriks, Katsuhide Fujita, Takayuki Ito, and Catholijn M Jonker. 2015. The automated negotiating agents competition, 2010–2015. *AI Magazine* 36, 4 (2015), 115–118.

Tim Baarslag, Koen Hindriks, Mark Hendrikx, Alex Dirkzwager, and Catholijn Jonker. 2014. *Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies.* Vol. 535. 61–83. https://doi.org/10.1007/978-4-431-54758-7_4

Tim Baarslag, Koen Hindriks, and Catholijn Jonker. 2013. *Acceptance Conditions in Automated Negotiation.* Springer Berlin Heidelberg, Berlin, Heidelberg, 95–111. https://doi.org/10.1007/978-3-642-30737-9_6

Eibe Frank, Mark Hall, and Bernhard Pfahringer. 2012. Locally Weighted Naive Bayes. (10 2012).

Koen V Hindriks and Dmytro Tykhonov. 2008. Towards a quality assessment method for learning preference profiles in negotiation. In *Agent-mediated electronic commerce and trading agent design and analysis*. Springer, 46–59.

Haralambie Leahu, Michael Kaisers, and Tim Baarslag. 2019. Automated Negotiation with Gaussian Process-based Utility Models. https://doi.org/10.24963/ijcai.2019/60

Stuart Russell and Peter Norvig. 2009. *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall Press, USA.

Nayyar Zaidi, Jesús Cerquides, Mark Carman, and Geoffrey Webb. 2013. Alleviating Naive Bayes Attribute Independence Assumption by Attribute Weighting. *The Journal of Machine Learning Research* 14 (06 2013), 1947–1988.