The Fawkes Agent—the ANAC 2013 Negotiation Contest Winner

Chapter · March 2015

DOI: 10.1007/978-4-431-55525-4_10

Vincent Koeman
Vrije Universiteit Amsterdam

The Fawkes Agent—the ANAC 2013 Negotiation Contest Winner

Vincent J. Koeman, Kees Boon, Joris Z. van den Oever, Madalin Dumitru-Guzu and Laurentiu Catalin Stanculescu

1 Introduction

Within the automated negotiation field, there are many ways to create a negotiation agent. In this paper, we will discuss an agent, Fawkes, that has been created for the Automated Negotiation Agents Contest (ANAC) 2013. ANAC is a competition that uses the GENIUS software framework to allow agents to negotiate over different domains in a tournament set-up as a way to compare the performance of different strategies.

This agent has been designed using the Bidding strategy, Opponent model, and Acceptance strategy (BOA) framework as facilitated by the competition software. This separates the three different components using a standard interface, as shown in Fig. 1. This has been at the core of the Fawkes agent design process, making it possible to design and develop the different components separately and in order.

The main strategy of our agent is to make trend predictions on the bids of the opponent in order to determine if they will have more utility for us in the future, as described by [2]. We will also show how this was improved upon. One of the more prominent changes is that instead of a random bid within a utility range for our agent, we offer the bid in this range that is most likely to have the greatest utility for the opponent, based on what can be modeled from their bids.

The remainder of this paper is organized as follows. First, the underlying research used by our agent will be described in Sect. 2. Next, the way this was implemented using the BOA framework will be discussed in Sect. 3. Section 4 discusses the results of the ANAC 2013 competition and how the agent performed there. In Sect. 5, possible improvements and further avenues of research will be proposed. Finally, in Sect. 6, the performance of the agent will be interpreted.

V.J. Koeman (✉) · K. Boon · J.Z. van den Oever · M. Dumitru-Guzu · L.C. Stanculescu
Delft University of Technology, Delft, Netherlands
e-mail: vj.koeman@quicknet.nl

K. Fujita et al. (eds.), *Next Frontier in Agent-Based Complex Automated Negotiation*, Studies in Computational Intelligence 596,
DOI 10.1007/978-4-431-55525-4_10

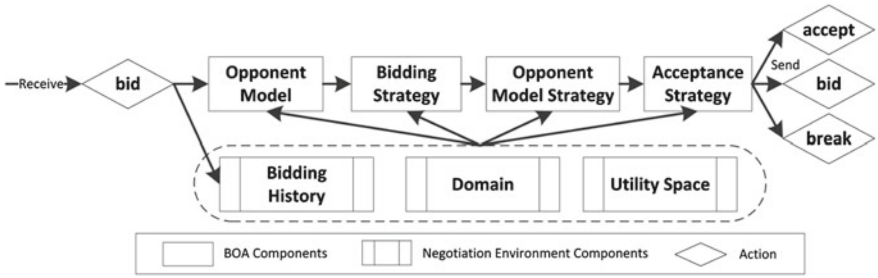

Fig. 1 The structure of a BOA agent

2 Related Work

The field of automated negotiation is a diverse one. However, there are a few papers that have been at the core of the negotiation strategy implemented in the Fawkes agent. In this section, the core work we built on will be discussed first, followed by some concepts that form the basis for the improvements we implemented, including the opponent model and acceptance strategy.

The OMAC (Opponent Modeling and Adaptive Concession) [2] negotiation approach was introduced by Chen et al. for the 2012 ANAC. During that competition, it proved to be an effective strategy. However, it also showed itself to be open for improvements. This negotiation strategy works by modeling opponent bids as wavelets, which can be used to make predictions on the utility of opponent bids to come. The specific method used for this by the OMAC agent is the application of a cubic smoothing spline.

OMAC makes use of a discounting utility expectation function to determine if the prediction is better or worse than the reserved utility for that time. If it is better, it strives towards making bids around that utility. If it is not better, however, it adjusts the last bid towards the reserved utility value. This allows for making concessions while still making the most of the time available for negotiating. Blindly following this prediction might have negative consequences, especially in cases where the available dataset is small or when the opponent is a stubborn agent. For this reason, the reserved utility u', decreasing over time, is used to provide a time based minimum utility.

One thing to keep in mind here is that all utility is considered from the agents own perspective; the opponent might have a very different view on utility. For this reason, the creation of an opponent model for utility can have significant effects (see Sect. 3). This has been leveraged by changing the way counteroffers are made. Instead of random selection of a counteroffer in the given range around u', the offer with highest predicted utility for the opponent within that range is used. The opponent model implemented is based on the frequency model as provided by the example that is a part of GENIUS.

Finally, the basic idea of the acceptance strategy comes from two places: the OMAC agent, and [1]. The previously mentioned reserved utility and predicted utilities from OMAC determine if our counteroffer is going to be better than a offer we had before. This is extended with a combination of acceptance conditions that incorporates AC_{next}, AC_{time}, and AC_{const}. The details of its further development can be found in the next section.

3 Development

GENIUS supports bilateral negotiation tournaments: agents alternate in making bids. A bid is either accepted, countered, or rejected. When a bid is accepted, the negotiation ends, and both agents get the utility that results from that bid. If it is countered, the negotiation continues. If it is rejected, the negotiation ends and both agents receive zero utility. In addition, the negotiation has a time limit, either in milliseconds or in number of bids exchanged. In the ANAC, there was also the added issue of diminishing utility over time. That is, a bid at the start of the negotiation gives higher utility than that same bid near the end. Thus, it becomes potentially advantageous to conclude negotiations swiftly. In ANAC 2013, the agents utility functions were linear and additive. Opponents did not have access to each others utility function.

Our agent extended the ideas behind the OMAC agent, which came third in ANAC 2012. In the next paragraphs, we describe the Fawkes parts using the BOA framework: bidding strategy, opponent modeling, and acceptance strategy. However, as our bidding strategy depends on the opponent modeling, this will be discussed first.

3.1 Opponent Modeling

The purpose of the opponent model is to gain information about the opponent to be able to predict their future moves. In addition, the utility space of the opponent is estimated in order to estimate the opponent's utility for a certain bid. Thus, for each bid of the opponent, several models are updated.

First of all, the values assigned to each issue by the opponent are checked, in order to see if any value differs from the previous bid. As the utility of a bid can be computed as a weighted sum of the utilities associated with the values for each issue, the preferences of an agent are linearly additive functions defined by a set of weights and corresponding evaluation functions for each of the issues. As the range of utility is 0–1, and thus the range of the evaluation functions is 0–1, the weights are to be normalized such that their sum equals 1. When a certain value has not changed, the according issue's weight is increased, until a certain maximum. Thus, options that change more often are seen as having a lesser utility than options that change less often. Using the updated weights and a constant learn value addition, the evaluation functions are updated as well. This model can be classified as a linear frequency model.

Next, the time difference between the opponent's current and previous bid is checked. The maximum time difference between the bids of an opponent is used to estimate how often an opponent makes a bid. To prevent abuse, this value cannot be too high. Moreover, as we estimated that 1 bid per 0.01 s can be made at most, a minimum for this time difference has been put in place as well. This minimum ensures the agent's performance, as it is also the minimum window for which data is saved. Following the OMAC model, a map of all opponent's bids and the accompanying utility for our agent is created. This map is then put through a Daubechie wavelet decomposition using the JWave [5] library, in order to estimate a curve for our received utility. Next, a cubic smoothing spline is run over this decomposition for noise reduction using the SSJ [4] library. Finally, the ratio between the original and smoothed curve is saved as well, indicating the certainty of the value. The smoothed curve and the standard deviation of all ratios are used in the bidding strategy, as discussed in the next part.

3.2 Bidding Strategy

The first bid of the agent is the one that has the most utility for the agent in the domain. As a wavelet decomposition can only be done after having more than one bid of the opponent, our second bid is always the same as the first one as well. For each following bid, a target utility is calculated based on the opponent modeling. This is done by using principles from the OMAC agent again. Starting from the best possible bid for us, a concession rate is dynamically determined. A reserved utility function is used to determine the minimum utility at a given time step. This ensures that we do not concede too quickly to stubborn opponents. This function takes discount, the domain's reservation (failure) value, and other parameters into account, as discussed at the end of this section. Besides this reserved utility, an estimate of the bids received in the near future is made using the smoothed curve from the opponent model, taking the certainty of these values into account. When an estimated utility is above the reserved utility, this is an optimistic scenario. In this scenario, the target utility is the same as the highest estimated received utility. However, to be though, and considering possible errors in the predictions, a linear concession towards this value is done, instead of offering it immediately. This concession takes the current time and discount factor into account.

When all estimated received utilities are below the reserved utility, this is a pessimistic scenario. In this scenario, a stronger concession might be acceptable when it is seen as a local optimum. This mechanism prevents losing the opportunity to reach a globally good agreement, especially in discounted domains. When no acceptable offer exists in this scenario, the target utility is set to the reserved utility.

The calculated target utility is used to determine the lower boundary of the range of utilities in which bids can be made. However, several significant improvements to this value have been made in comparison to the original OMAC agent. First of all, an additional correction regarding the current time is made, lowering the target utility at the end of the negotiation more quickly. Furthermore, a correction for inertia has been added. This correction lowers the target utility when the same offer has been made over and over again by our agent. Moreover, as aforementioned, instead of selecting a random bid from this range, the opponent preference model is used to select the bid which has the highest probable utility for the opponent instead. This last step, the opponent model strategy, also supports the corrections as increasing the range leads to more possible bids, from which a random bid might not be beneficial to either sides. This implementation makes the Fawkes aim at a win-win situation.

3.3 Acceptance Strategy

Compared with the original OMAC agent, the acceptance strategy was changed to a more sophisticated one. Although it is still inspired by [2], it has been improved by taking into account the work done by [1], and consists from a combination of rules. First, the agent will accept when the bid that will be offered to the opponent has a lower utility than the last bid that was offered by the opponent (AC_{next}), as the original OMAC agent does. However, any bid needs to have a utility greater than the average utility of all possible bids in the domain space, from the agent's own perspective, to be considered at all (AC_{const}). In addition, if the current time is close enough to the end of the session, then the last bid offered by the opponent is accepted, as long as it is at least as good as the best recent bid (AC_{time}). This time limit is determined by measuring how long it takes to get an answer from the opponent and keeping the longest time period, under a certain maximum. Using all of these methods ensures a high likelihood of having an agreement, whilst protecting our own interests as well.

3.4 Parameter Search

The agent uses several parameters:

- β: the basic concession rate, e.g. how much the current time affects the concession rate.
- ν: the risk factor, e.g. the used impact of discount.
- ρ: the tolerance threshold, used to check if an opponent's bid is acceptable.
- ζ: the maximum predictive range.

A brute force approach was used to select the best values for these parameters. This approach consisted of a series of 12 s tournaments in which the agent negotiated against the Bayesian, Simple, Boulware, Conceder, Gahboninho V3, HardHeaded, IAMhaggler2011, AgentLG, CUHK, and OMAC_sp2012b agents. The tournaments were executed on four different domains. After initial exploration, β was checked at

0.001, 0.005, 0.01, 0.1 and 0.3. The effect of ρ was evaluated at 0.01, 0.1, 0.3, 0.5 and 0.8. ν was fixed at 0.2, as suggested by the original OMAC research. Similarly, ζ was fixed to a value of 10.

The effectiveness of the combination of parameters was measured by whether the agent reached any agreement, whether it had the higher utility of the two, if its utility was larger than 0.5, what the opponents utility was, and if both utilities were larger than either 0.5, 0.8 or 0.9. Those parameters that fulfilled most of these requirements were used as a first estimate of the best parameters.

The resulting choice of parameters was $\beta = 0.002$ and $\rho = 0.8$, which were found by iterating over values close to those found as best in the earlier search. Divided over five computers, this parameter search took roughly a day to complete.

4 Results

The Fawkes Agent was the winner of the ANAC 2013 held in May 2013. The competition results are discussed in this section.

There were 19 participating agents in the competition. The competition consisted of a preliminary round and a final round. In the preliminary round, negotiations between every combination of the 19 agents with their eighteen opponents were carried out. Participants were allowed to submit a single domain each. The negotiations were performed on eleven domains, which were randomly selected from the submitted domains. Each negotiation was repeated ten times. Since each domain consisted of a pair of preference profiles, both agents negotiating in a domain swapped profiles after those ten repeats. Thus, in total, there were twenty negotiations carried out per domain. The time deadline for a single negotiation session was three minutes. If no agreement was reached at that time, or if either agent opted out of the negotiation, the agents received a predefined conflict utility. The scoring was based primarily on the highest mean utility over all the negotiations, and over the lowest variance second. The highest seven agents were selected to proceed to the final round.

In the final round, the remaining seven agents negotiated on 18 domains (12 from 2013 and 6 from 2012), the other properties of the tournament remaining unchanged.

The results of the preliminary round is given in Table 1. The Fawkes agent performed well, coming in second or third place as the distribution of our score around our mean overlaps with the distribution of the TMF Agent.

The results of the final round are given in Table 2. One must take care with analyzing these results not to arrive at a tautological conclusion: ‘The better agents perform better because they have better results.’ With that in mind, it shows that in the final, utilities were higher. Thus, the agents that performed worse in the preceding round affected the better agents negatively. This suggests the hypothesis that The Fawkes and other high-scoring agents were unable to exploit weaker agents to their benefit, but rather benefit from having a skilled opponent. In other words, finding any agreement seems to be a rewarding strategy as compared to prioritizing your own utility.

Table 1 ANAC2013 preliminary round results

Position	Agent	Rank	Mean	Variance (per run)
1	Agent KF	1	0.562	0.00019
2	The Fawkes	2–3	0.522	0.00132
3	TMF agent	2–4	0.516	0.00163
4	Meta agent	3–4	0.495	0.00252
5	G-Agent	5–8	0.457	0.00241
6	Inox agent	5–8	0.455	0.00235
7	Slava agent	5–11	0.447	0.00018
8	VA stock market agent	5–11	0.446	0.00520
9	RoO agent	7–11	0.432	0.00313
10	Agent talex	7–11	0.431	0.00285
11	Agent MRK2	7–11	0.430	0.00344
12	Elizabeth	12–14	0.387	0.00443
13	ReuthLiron	12–15	0.374	0.00416
14	BOAconstrictorAgent	12–15	0.373	0.00141
15	Pelican	13–18	0.359	0.00434
16	Oriel_Einat_Agent	15–18	0.350	0.00534
17	MasterQiao	15–18	0.345	0.00214
18	Eagent	15–18	0.338	0.00707
19	Clear agent	19	0.315	0.00109

The seven highest scoring agents proceeded to the final round

Indeed, this often seems to lead to high utility contracts. This reflects the well-known prisoners dilemma.

The Fawkes has some provisions that have it prioritize agreement over its own utility. As can be seen in its acceptance strategy. However due to the opponent modeling, it also implicitly considers ways towards reaching an agreement, regardless of whether the opponent is cooperative or not.

Like The Fawkes, Meta Agent sought to exploit the ideas from previous work [3]. It selects an existing strategy based on its performance in similar domains. The classification of the domain is done through machine learning. As can be seen in Table 2, its results are nearly indistinguishable from ours. Since the Meta Agent tries to select 'the best agent for the job' on a given domain, it can be concluded that The Fawkes just barely outperforms every agent from preceding years when they are operating in the environment most suited to them. This is of course contingent on Meta Agents ability to correctly select the right strategy. It would be interesting to see how the Meta Agent performs if it was trained and had access to the 2013 agents, including The Fawkes. The hypothesis would predict that the strategies which Meta Agent selects are socially minded agents that favor reaching an agreement over obtaining high utility.

Table 2 ANAC2013 final round results

Rank	Agent	Mean	Variance
1	The Fawkes	0.606434	0.000011
2	Meta agent	0.600209	0.000083
3	TMF agent	0.583094	0.000012
4–5	Inox agent	0.568215	0.000069
4–5	G-Agent	0.564908	0.000055
6	Agent KF	0.534514	0.000147
7	Slava agent	0.484973	0.000023

The KF Agent, being first in the preliminary round, did not appear in the top three of the final round. It rewards cooperative behavior in its opponents by becoming more cooperative itself. As such, it would be expected that it is less able to find an agreement with uncooperative agents. Comparing this to the iterated prisoners dilemma, it can be concluded that retaliating after a defection is counter-productive.

In the detailed results of the finals, this effect is less pronounced, but still visible. Here we see that the lower scoring agents still score well against the higher ones, while pairings that both ended up low scored mutually mediocre against each other, suggesting an inability to find mutually beneficial agreements with weaker partners. Additionally, the lower scoring agents end up obtaining no utility in a non-negligible amount of the time. In contrast, the higher scoring agents usually do obtain mutually beneficial agreements and it is exceedingly rare for them to end up with zero utility. This suggests that adjusting your behavior based on the perceived strength of the opponent can be worthwhile. As shown by the highest ranking finalists, mutually obtaining high utility is possible, and especially in repeated negotiations it is desirable to do so. Again, the comparison to retaliation in iterated prisoners dilemma, and its effect on overall social welfare is apparent.

5 Future Work

While many improvements upon the work of others have been made, there are still unexplored possibilities for the Fawkes agent. Aside from tuning several variables through testing, there are several structural changes that can be made. For example, the current opponent model assumes a linear combination of preferences. This could be extended to work for non-linear utility functions.

During the parameter testing phase, some of the criteria were about whether the agent was able to achieve over 0.8 or 0.9 utility. However, as shown by the results of the competition, our actually obtained results are nearer to 0.6 on average for the finals, and somewhat lower in the preliminaries. It might be worth investigating whether optimizing for parameters that aim for this empirically more realistic goal or a utility that is a little higher is a good option. This is in line with emphasizing achieving agreement over achieving optimal individual score. In other words: is optimizing towards unrealistic goals counterproductive to your own utility? Additionally, the agent would likely score higher on the most social agent score, which was another scoring measure kept during the ANAC. With the option of storing information on other agents, the previously obtained utility might be a good indicator on the necessity of going for high mutual utility, or to go for any agreement that has more than zero utility.

Further improvement on the opponent model could be to extend it to learn over the course of several runs in a domain, by storing the opponent utility model per domain. A further extension of that would be to have several of the constants be dynamic over the course of several domain runs. A machine learning mechanism such as the one used by the Meta agent could be used to optimize those constants. This would mean that the agent learns not just the best negotiation strategy per domain, but that it is narrowed down even further to operate well on specific opponents or classes of opponents.

6 Conclusion

In this paper, we discussed the design and performance of the Fawkes Agent, the winner of the ANAC 2013 competition.

Meta Agent and The Fawkes both performed well, and were both based on either selecting previous techniques or combining and refining old ones. As such, we can conclude that the automated bilateral negotiation field contains many strategies that are well suited for optimization and streamlining. Both agents do not explicitly care for the opponents disposition, but only about what is the best course of action to get high utility. The results of ANAC 2013 suggest that this mainly involves reaching an agreement, rather than optimizing towards your own utility. In the domains tested, it appears that this approach still leads to high individual utility.

References

1. Baarslag, T., Hindriks, K., Jonker, C.: Acceptance conditions in automated negotiation. Complex Automated Negotiations: Theories, Models, and Software Competitions, pp. 95–111. Springer, New York (2013)
2. Chen, S., Weiss, G.: An efficient and adaptive approach to negotiation in complex environments. In: ECAI, pp. 228–233 (2012)
3. Ilany, L., Gal, Y.: Algorithm selection in bilateral negotiation. In: Workshops at the 27th AAAI Conference on Artificial Intelligence (2013)
4. L'Ecuyer, P., Bague, M., Bonnet, S., Buist, E., Dion, M., Edel, Y., Hong, R.H.S., Keller, A., Marcotte, E., Meliani, L., Panneton, F., Parent-Chartier, J.S., Simard, R., Teule, C., Tremblay, P.A., Vaucher, J.: SSJ—stochastic simulation in java (2010)
5. Scheiblich, C.: JWave—java implementation of wavelet transform algorithms (2010)