Chapter 10
An Adaptive Negotiation Strategy for Real-Time Bilateral Negotiations

Alexander Dirkzwager and Mark Hendrikx

Abstract Each year the Automated Negotiating Agent Competition (ANAC) introduces an increasingly complex negotiation setting to stimulate the development of negotiation strategies. This year, the competition featured a real-time bilateral negotiation setting with private reservation values and time-based discounts. This work introduces the strategy of one of the top three finalists: *The Negotiator Reloaded* (TNR). TNR is the first ANAC agent created using the BOA framework, a framework that allows separately developing and optimizing the components of a negotiation strategy. The agent uses a complex strategy that takes the opponent's behavior and the domain characteristics into account. This work presents the implementation, optimization, and evaluation of the strategy.

Keywords Automated negotiation strategy • Bayesian learning • Domain analysis • Strategy prediction

10.1 Introduction

Last year, the ANAC 2011 competition introduced a negotiation setting in which agents competed in a real-time bilateral negotiation on domains with time-based discounts [1]. This year the setting was extended to feature private reservation values that are discounted over time.

This work introduces the strategy of the third place finalist and the best performing agent on undiscounted domains in the ANAC 2012 competition: *The Negotiator Reloaded* (TNR). TNR is the first agent based on the BOA framework [2], a framework that allows to separately develop the bidding strategy,

A. Dirkzwager (✉) • M. Hendrikx
Delft University of Technology, Interactive Intelligence Group,
Mekelweg 4, Delft, The Netherlands
e-mail: A.S.Y.Dirkzwager@tudelft.nl; M.J.C.Hendrikx@tudelft.nl

I. Marsa-Maestre et al. (eds.), *Novel Insights in Agent-based Complex Automated Negotiation*, Studies in Computational Intelligence 535,
DOI 10.1007/978-4-431-54758-7_10,

opponent model, and acceptance conditions. The flexibility of this framework allows us to optimize the negotiation strategy using the components of agents introduced in previous ANAC competitions.

The following sections discuss the implementation of the agent. Section 10.2 discusses the negotiation strategy, how it is implemented and optimized using the BOA framework. In Sect. 10.3 a toolkit of quality measures is used to quantify the performance of the negotiation strategy. Finally, Sect. 10.4 provides directions for future research.

10.2 Negotiation Strategy

This section discusses the strategy of *The Negotiator Reloaded.* Section 10.2.1 briefly describes the BOA framework used to create TNR (for a detailed discussion, see [2]). Next, Sect. 10.2.2 discusses how the BOA framework is used to implement TNR's components.

10.2.1 Introduction to the BOA Framework

The BOA framework is build upon GENIUS [3] which allows to separately develop the components of a negotiation strategy. The BOA framework makes a distinction between three types of components: a **B**idding strategy which maps a negotiation trace to a bid; an **O**pponent model, which is a learning technique used to model the opponent's preference profile; and finally an **A**cceptance strategy which determines whether the opponent's offer is acceptable. A full negotiation strategy is created by selecting a component for each of the three types. In fact, the full Cartesian product of these components can be evaluated.

There are three main advantages to implementing an agent as a BOA compatible agent: first, each component can be evaluated in isolation; second, a component can be easily switched for an alternative—possibly better—component; and finally, the implementation of separate component simplifies agent creation.

Figure 10.1 provides an overview of how the components interact. When receiving an opponent's bid, the BOA agent first updates the *bidding history* and *opponent model*. Given the opponent's bid, the *bidding strategy* generates a set of similarly preferred counter offers. Next, the *bidding strategy* uses the *opponent model* to select a bid from this set by taking the opponent's utility into account. Finally, the *acceptance strategy* decides whether the opponent's offer should be accepted. If the opponent's bid is not accepted, then the bid generated by the *bidding strategy* is offered instead.

Each component of TNR was implemented separately using the BOA framework. The following section discusses the implementation and optimization of each component in detail.

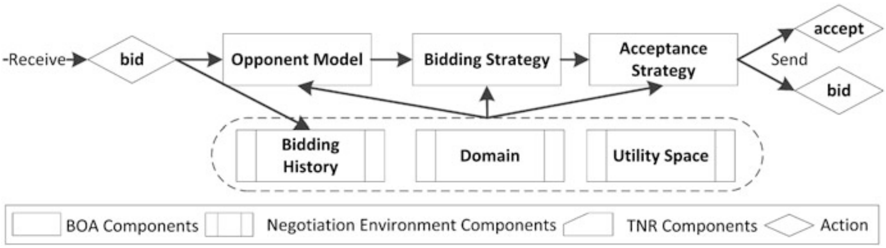

Fig. 10.1 Overview of the BOA framework

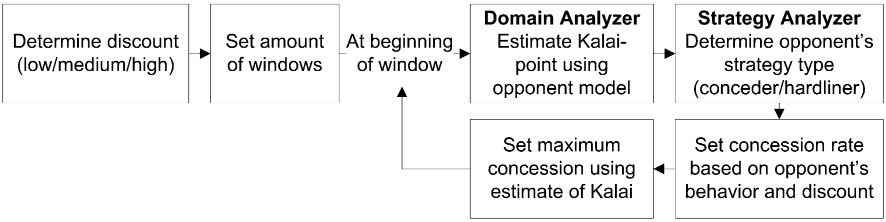

Fig. 10.2 Overview of bidding strategy of TNR

10.2.2 Implementing the BOA Components

This section discusses the three BOA components of TNR in turn: the bidding strategy, the opponent model, and the acceptance strategy. The discussion of each component consists of a description of its implementation, as well as how the component is optimized using quality measures and the BOA framework.

10.2.2.1 Bidding Strategy

TNR is a BOA agent that takes the opponent's strategy and domain characteristics into account to optimize its negotiation strategy. The discussion below follows the diagram of the complete negotiation strategy depicted in Fig. 10.2.

The first step taken by TNR, is that it determines if the discount is low, medium, or high. Next, the time is divided in a set of windows. At the start of each window, the *domain analyzer* is used to estimate the Kalai-Smorodinsky point and the *strategy analyzer* is used to determine the opponent is a conceder or hardliner. Note that preferably these calculations should be done each turn, however, this proved too computationally expensive. The target utility in a specific round is determined using the standard time-dependent decision function [4] depicted in Eq. (10.1). We opted for this decision function as its parameters can be adjusted during the negotiation.

$$P_{min} + (P_{max} - P_{min}) \cdot (1 - F(t)) \;\; where \;\; F(t) = k + (1 - k) \cdot t^{1/e}. \quad (10.1)$$

The value for the concession rate e is selected from a table that maps the discount type (low, medium, high) and opponent's strategy type (conceder or hardliner) to a concession rate. While the discount type does not change, the opponent's behavior is likely to change over time. The maximum concession P_{min} is set to the estimated Kalai-Smorodinsky point calculated by the *domain analyzer*. For domains with a discount, P_{min} is multiplied by the discount to ensure that the agent concedes faster. When the undiscounted reservation value is higher than P_{min}, then P_{min} is set to the reservation value. As a safeguard, P_{min} is not allowed to be lower than a predefined constant. The variable k is always 0, and P_{max} 1. The calculated target utility is used by the bid selector to select a bid with a utility as close as possible to the target utility.

The tactic as discussed above strongly relies on the concession rate table. Since three discount types and two strategy types are distinguished, there are six concession rates to be determined. To do so, we created a variant of the ANAC 2011 competition that excludes the agent *ValueModelAgent* and the domains *Energy* and *NiceOrDie* to decrease computational time. For each discount type we generated a representative set of domains, for example for the type medium discounts we created a set of preference profiles with discounts in the range $(0.4, 0.8]$. Next, we ran the competition multiple times for each discount type to determine the optimum values for the strategy type parameters.

10.2.2.2 Opponent Model

As part of our implementation of the BOA framework, the opponent models of previous ANAC agents were isolated and modified to be compatible with the BOA framework [2]. Since the components now use an identical interface, their quality can be compared using accuracy metrics as discussed by Baarslag et al. [5]. An example of such a measure is the Pearson correlation between the estimated and opponent's real preference profile. In this work we found the *IAMhaggler Bayesian Model* introduced by Williams et al. [6] to be the most accurate in estimating the Kalai-Smorodinsky point. *The Negotiator Reloaded* uses this model as part of its *domain analyzer*.

The computational resources required by the *IAMhaggler Bayesian Model* depend strongly on the domain size. Therefore the opponent model is not used in very large domains, in which case the agent estimates the Kalai-Smorodinsky point to be equal to a predefined constant. While the accuracy of the estimation increases at the beginning of the negotiation, later on it actually decreases over time. We believe that this can be attributed to the assumed decision function that more accurately reflects the real decision function at the beginning of the negotiation for most agents. To avoid this decay in accuracy, *The Negotiator Reloaded* stops updating the opponent model after a predefined amount of time.

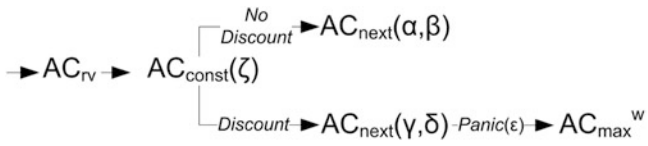

Fig. 10.3 Basic acceptance conditions used by TNR

10.2.2.3 Acceptance Strategy

The acceptance strategy of TNR consists of a set of basic acceptance conditions discussed by Baarslag et al. [7]. The flowchart of the acceptance strategy is depicted in Fig. 10.3. As visualized, there are two paths depending if the discount is negligible or not and six parameters $(\alpha, \beta, \gamma, \delta, \epsilon, \zeta)$.

AC_{rv} is an acceptance condition that decides to accept when the discounted utility of the bid under consideration for offering is lower or equal to the reservation value. AC_{const} is an acceptance condition that accepts when the utility of the opponent's bid is at least equal to a constant ζ. AC_{next} accepts when a linear transformation of the opponent's bid utility is better than the utility of the bid under consideration. Finally, the agent uses AC_{max^w} when there is $1 - \epsilon$ time left and the utilities of the bids of the agents have not crossed. This acceptance condition compares the offered bid with the maximum bid that has been given in a particular window and will accept if is higher than the maximum given in the previous window and if it is higher than 0.5.

The multi-acceptance criteria (MAC) functionality of the BOA framework [2] was used to optimize the acceptance strategy. In short, the MAC can be used to run a large set of acceptance conditions in parallel during the same negotiation thread, assuming that the computational cost of each acceptance condition is minimal. In total 288 acceptance conditions were tested varying in the usage of the panic phase and the four parameters of the two acceptance conditions AC_{next}. The parameters $\alpha = 1.0, \beta = 0.0, \gamma = 1.05, \delta = 0.0, \epsilon = 0.99$ were found to be optimal.

10.3 Empirical Evaluation

The previous sections introduced the BOA framework, and described how it has been applied to optimize our negotiation agent. To demonstrate that using the BOA framework we were able to optimize TNR, and to analyze the behavior of the agent, this section discusses the results of a modified ANAC 2011 competition. Section 10.3.1 details the setup of this tournament and introduces the selected quality measures. Next, Sect. 10.3.2 evaluates the results.

Table 10.1 Overview of quality measures used to quantity performance

Quality measure	Description
Avg. time of agreement	The average time of agreement of all matches which resulted in agreement
Std. time of agreement	Standard deviation of the average time of agreement of each run
Avg. discounted utility	Average discounted utility of all matches
Std. discounted utility	Standard deviation of the average discounted utility of each run
Ratio of agreement	Percentage of matches which resulted in an agreement
Avg. Kalai distance	The average Kalai distance of all matches
Trajectory analysis	The opponent's moves can be classified based on their concession [8]. A unfortunate move for example, is a concession that accidentally results in a lower utility for the agent in comparison to the opponent's previous bid

10.3.1 Experimental Setup

The default alternating offers protocol of GENIUS is used to run a tournament identical to the ANAC 2011 competition, except that *ValueModelAgent* is excluded and TNR is included, and that the agents compete on variants of the ANAC 2011 based on the three discount types, resulting in a total of 24 domains. The complete tournament is ran ten times to increase the statistical significance of the results. In a single tournament eight agents compete against all agents except themselves on 24 domains, playing both possible preference profiles. This results in a total of 13,440 matches that were ran using a distributed version of GENIUS. The overview of quality measures that were implemented to quantify the agents their performance is depicted in Table 10.1.

10.3.2 Experimental Results

This section discusses the results of the tournament visualized in Table 10.2. Note that due to space constraints the standard deviations are not shown as they are negligible, as well as the ratio of agreement, which is higher than 99% for all agents. This high percentage of agreement illustrates that most ANAC 2011 agents prefer agreement over disagreement, and ultimately give in.

TNR achieves the highest discounted utility, and strongly outperforms the runner-up. With regard to the trajectory measures, the agent makes the least concessions, as indicated by its high percentage of silent moves and its low ranking on the percentage of unfortunate moves, fortunate moves, nice moves, and concession moves, which are all types of moves made when the agent tries to make a concession. TNR agent does not make selfish moves that increase its own utility without conceding, which can be attributed to its usage of the time-dependent strategy.

Table 10.2 Overview of the experimental results

Agent	Avg. time of agreement	Avg. discounted utility	Avg. unfortunate moves	Avg. fortunate moves	Avg. nice moves	Avg. selfish moves	Avg. concession moves	Avg. silent moves
The Negotiator Reloaded	0.545	**0.809**	0.033	0.000	0.033	0.000	0.003	**0.930**
Gahboninho	0.528	0.782	0.027	0.001	0.038	0.002	0.004	0.929
HardHeaded	**0.638**	0.778	0.111	0.013	**0.133**	0.052	0.028	0.663
Nice Tit For Tat	0.605	0.767	0.112	0.079	0.066	0.116	0.11	0.512
Agent K2	0.493	0.755	0.154	0.116	0.069	**0.203**	0.174	0.284
The Negotiator	0.591	0.751	0.080	0.036	0.071	0.077	0.051	0.685
IAMhaggler 2011	0.377	0.748	**0.162**	**0.120**	0.074	**0.203**	**0.178**	0.263
BRAMAgent	0.578	0.740	0.115	0.075	0.085	0.148	0.104	0.472

Bold text is used to emphasize the highest value, and underlined the lowest value. All averages are in the range [0, 1]

10.4 Conclusion and Future Work

In this work we discussed the implementation, optimization, and evaluation of a flexible negotiation strategy that outperforms the ANAC 2011 agents on various domains and performs well in the ANAC 2012. *The Negotiator Reloaded* is the first ANAC agent developed using the BOA framework.

The tournament results of our ANAC 2011 variant competition discussed in Sect. 10.3 indicate a strong performance of TNR on various domains against a range of opponents. In the ANAC 2012 competition, TNR finished third overall and achieved the highest utility on undiscounted domains. The agent finished fifth when only focusing on the discounted domains. We believe that this can be attributed to our experimental setup used to optimize the agent: ANAC 2011 agents perform relatively poor on discounted domains.

For future work, it could be interesting to enable TNR to identify behavior-based strategies. In this case the bidding strategy should be further extended to use an effective counter-strategy. Furthermore, the opponent model now used to estimate the Kalai-Smorodinsky point could also be employed to estimate the best bid to offer to the opponent given a set of similarly preferred bids.

Acknowledgements We would like to thank Tim Baarslag, Koen Hindriks, and Catholijn Jonker for introducing us to the field of bilateral negotiation and reviewing our paper. Furthermore, we thank the Universiteitsfonds Delft and the Interactive Intelligence Group of the Delft University of Technology for sponsoring our trip to the AAMAS 2012.

References

1. Baarslag, T., Fujita, K., Gerding, E.H., Hindriks, K., Ito, T., Jennings, N.R., Jonker, C., Kraus, S., Lin, R., Robu, V., Williams, C.R.: Evaluating practical negotiating agents: results and analysis of the 2011 international competition. Artif. Intell. **198**(0), 73–103 (2013)
2. Baarslag, T., Hindriks, K., Hendrikx, M., Dirkzwager, A., Jonker, C.: Decoupling negotiating agents to explore the space of negotiation strategies. In: Proceedings of the Fifth International Workshop on Agent-based Complex Automated Negotiations (ACAN 2012) (2012)
3. Lin, R., Kraus, S., Baarslag. T., Tykhonov, D., Hindriks, K., Jonker, C.: Genius: An integrated environment for supporting the design of generic automated negotiators. Computational Intelligence, Blackwell Publishing Inc. http://mmi.tudelft.nl/sites/default/files/genius.pdf Doi: 10.1111/j.1467-8640.2012.00463.x
4. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. Robot. Auton. Syst. **24**(3–4), 159–182 (1998) Multi-Agent Rationality.
5. Baarslag, T., Hendrikx, M., Hindriks, K., Jonker, C.: Measuring the performance of online opponent models in automated bilateral negotiation. In: Thielscher, M., Zhang, D., (eds.): AI 2012: Advances in Artificial Intelligence. Lecture Notes in Computer Science, vol. 7691, pp.1–14. Springer (2012)
6. Williams, C.R., Robu, V., Gerding, E.H., Jennings, N.R.: Iamhaggler2011: a gaussian process regression based negotiation agent. In: Ito, T., Zhang, M., Robu, V., Matsuo, T., (eds.) Complex Automated Negotiations: Theories, Models, and Software Competitions. Studies in Computational Intelligence, vol.435, pp. 209–212, Springer, Berlin (2013)
7. Baarslag, T., Hindriks, K., Jonker, C.: Acceptance conditions in automated negotiation. In: Ito, T., Zhang, M., Robu, V., Matsuo, T., (eds.) Complex Automated Negotiations: Theories, Models, and Software Competitions. Studies in Computational Intelligence, vol. 435, pp. 95–111. Springer, Berlin (2013)
8. Bosse, T., Jonker, C.M.: Human vs. computer behaviour in multi-issue negotiation. In: Proceedings of the Rational, Robust, and Secure Negotiation Mechanisms in Multi-Agent Systems. RRS ’05, Washington, DC, USA, IEEE Computer Society (2005) 11