DOLOS: A Negotiation Agent Based on Trickery and Deceit

Group 11: Joris Graff (6632300), Marco Mikulin (6600018), Priyanka Singhvi (6481892), Rens van Schouwenburg (6626394), Stefan Gaillard (3962903)

INFOMAS - Group 11, Multi-Agent Systems, 04-04-2019

***Abstract*—An intelligent negotiation agent called DOLOS has been created based on and inspired by components of other intelligent agent designs. For example, parts of the BOA architecture from other agents were used as a starting point, such as HardHeadedFrequencyModel from the HardHeaded agent (Van Springen et al. 2018). DOLOS was subsequently tested in several different multi-agent system environments against different other agents. After multiple rounds of testing and modifications, DOLOS ended up performing reasonably well, although no definite judgement can be made until after the Multi-Agent Systems 2019 tournament. Before DOLOS and automated negotiation in general can be used in real-life negotiations, several iterations of additions have to be made which are also addressed in this paper.**

***Keywords*—Automated negotiation; BOA architecture; Negotiation agent; Multi agent system.**

I. INTRODUCTION

This report describes a new intelligent agent design, named after the ancient Greek god of trickery and deceit, *DOLOS*. This agent is tested in automated bilateral multi-issue negotiations and developed in the GENIUS simulator environment. The agent design is based on the BOA structure proposed by [2] that decouples the negotiation design into four components: the bidding strategy, the opponent model, the opponent model strategy, and the acceptance criteria(although DOLOS does not make use of the opponent model strategy). Advantageous aspects of this structure are amongst other things, having the possibility to reuse and re-combine components from former *ANAC* agents [4] and having the opportunity to upgrade them. An important design choice involves our bidding strategy that is based on deceiving the opponent, especially the opponents opponent model, by simulating concessions, while in reality increasing our own utility with every move. In order to do this, we do not start the negotiation with a utility of 1.0 which unique in the literature. This report is structured as follows. First, the design choices are discussed for every BOA component. Secondly, the performance of the negotiation agent is quantified in its entirety. Next we also investigate the performance of each individual component of the BOA structure, by replacing the component by another component from an older ANAC agent and comparing the results. Thirdly, we reflect on the overall agent design and how it could be used in real-life negotiations. Finally, there is a reflection on the overall collaboration between the team members of the past 9 weeks of the project.

II. NEGOTIATION AGENT DESIGN

A. *Offering Strategy*

Two general decisions for our bidding strategy are important to justify upfront. First, we decided not to use machine learning techniques because of the following reasons. Machine learning is inductive in nature: it will predict on the basis of previously occurring patterns. For this reason, deceiving machine learning-based agents is much easier: using strategies that differ from those of previous agents will likely confuse an agent that bases its strategy on machine learning. A machine-learning based agent needs to play against the same agent time and time again to learn its strategy, so when confronted with a novel type of agent within the narrow confines of a single tournament, it may be completely put off track. This may play to our advantage when playing against machine learning-based agents. Second, we decided not to start off the negotiation with the bid that has maximum utility for us. It is conventionally known that most (naive) agents will start with the bid that has the highest utility for themselves, thus crippling the possibility of negotiating later on. This is where our agent differs from most others. We got inspiration by observing bidding on various e-commerce platforms such as e-bay and marktplaats, where bidders slowly increase the price of the item they wish to buy. Thus, we decided to reverse engineer this strategy, starting slightly above our minimum threshold utility and slowly increasing our utility as the session progresses. We further decided to confuse the opponents modelling of our agent by suddenly changing strategies based on what turn we are in. For the first 0.86 fraction of the turns, we slowly increase our own utility as described, whilst simultaneously attempting to heighten the opponents utility to create the impression that we are conceding. Until 0.96 of the turns, we deploy a tit for tat strategy that concedes based on how much the opponent concedes. This is meant in order to accommodate agents that reward cooperation by achieving mutual rapprochement. Then, from 0.96 to 0.97 of the turns, we shortly move to a third strategy. If we reach this stage, it is likely that the opponent is not very conceding (otherwise the tit for tat strategy would likely have led to a solution in the previous stage). Therefore, it might be more beneficial to draw out such an agent by actually retracting some of our concessions so far and retreating to a utility value that is very high for us. Finally, for the final few turns, we start conceding more rapidly in order to maximise the probability of reaching an agreement. In this stage we try to concede as much as is needed to have the opponent accept our proposal by gradually lowering our concession threshold until the opponent is likely to accept. In doing so, however, we never concede too much, i.e. we will never propose a bid that is estimated to give the opponent more than 1.2 times the utility that we ourselves gain. To optimise the efficiency of the bidding strategy, we followed the HardHeaded agent in creating a list of all bids with their corresponding utilities at the start of the session

(Van Springen et al. 2013). For this, we took over the helper class called BidSelector which can be found on the Genius website.

B. Acceptance Strategy

Our acceptance criterions is rather simple, since it can make use of the rather sophisticated lowering of the concession threshold that is already implemented in the bidding strategy. In any stage of the negotiation, we only accept bids if the bid offered to us is above our set threshold. For most of the negotiation, this threshold is static, since during this phase it is likely that the opponent will still concede and it is therefore not prudent to accept any bids except ones that have a very high utility. For the final stage of the negotiation, we make use of the bidding strategy to determine whether or not to accept a bid. Since the bidding strategy is slowly lowering the boundary to and optimal utility for each stage in the negotiation, we accept the bid if and only if the utility of the bid that we are about to give is lower than the bid that we receive.

C. Opponent Model

We are using a slightly modified version of HardHeadedFrequencyModel from the HardHeaded agent (Van Springen et al. 2013) in order to estimate the opponent utility. The HardHeadedFrequency model defines two variables called goldenValue and Learning Coefficient which are used to lower the weights of issues that change frequently and the evaluations of values that occur frequently. In the original model, the weights and evaluations are only updated from turn 2 onwards. Although this makes sense for the HardHeaded agent, for our agent it is crucial to start offering bids that are as suboptimal as possible for the opponent as quickly as possible. Therefore, we set the evaluations for all issue values contained in the first bid to 10 and return them to 1 after the second bid so that the model can do its work as usual. After a certain while, we decrease the golden value and the learn value addition because the opponent's later bids are probably more exploratory and therefore less revelatory about its preferences. This should counter the problem occurred by the original frequency model that the opponent model actually decreases its reliability over time.

D. Opponent Model Strategy

Our agent does not use the opponent model strategy, since each use that is made of the opponent model is coded directly into the bidding and acceptance strategies.

E. Preference Uncertainty

In order to estimate a utility profile from a limited ranking of bids, we used the approach sketched in Tsimpoukis et al. (2018), which approaches the utility estimation as a linear optimisation problem. That is, as in the article, we defined a linear optimisation problem where the estimator variables were a) a set of phi variables defined for each value, where a phi variable iis the product of a values evaluation and the weight of the issue it belongs to, and b) a set of slack variables for the comparison of each pair of bids. The linear optimisation problem was then defined as in Tsimpoukis et al. (2018), except with the addition of the additional constraints that the phi values of the best and worst bid should sum to the best and worst outcomes respectively, and that the set of the maximally evaluated values for each issue should sum to 1. The problem was formulated and solved using IBMs CPLEX package. We then used the estimated phi values to define a utility space where each issues weight and each values evaluation were normalised. Passing this utility space to the other classes, this allowed us to utilise functions that were originally defined on preference certainty by passing the constructed utility space to them. It should be noted that the preference estimator performed slightly disappointingly, especially for small bid rankings. In the Party domain, when given 10 bids, the agent performed rather randomly. The reason is that the solution of the problem often set multiple phi values to 0, even when in the actual utility space the evaluations of the corresponding values were strictly positive. Although we tried to compensate for this error by always increasing such phi values by a certain amount, this was apparently not enough to compensate for the loss of information caused by the 0 values whenever a small number of bids was given.

III. BENCHMARK SETUP

A. DOLOS vs. ANAC2011 agents

In order to measure the overall performance of the DOLOS agent, it was tested in a bilateral negotiation in the GENIUS simulator against former 2011 ANAC agents. The selected agents were HardHeaded (HH), Gahboninho, and IAMHaggler2011 as opponents for DOLOS and they negotiated in the Party domain. In addition to this we selected three distinct preference profiles from the GENIUS repository: *party1_utility.xml* (p1), *party2_utility.xml* (p2), *party4_utility.xml* (p4). For the Party domain, we setup DOLOS with (p1) vs. HH with (p2), DOLOS with (p2) vs. HH with (p4) and repeat this for the agents IAMHaggler2011 and Gahboninho. This is done to ensure some robustness of the results of the performance. The following performance measures were defined:

- Average distance to Pareto
- Average utility of DOLOS
- Average utility of the opponent
- Total wins (out of six negotiations)
- Number of negotiations ended up in a Pareto state (out of six negotiations)

B. Performance in Other Domains

The negotiation sessions against the *ANAC2011* agents were held within the Party domain, that is a multi-issue scenario. Each issue represents multiple features, e.g. the issue Location has four features Party Tent, Your Dorm, Party Room and Ballroom. Each scenario may differ in the number of issues, the total domain size and competitiveness or let say opposition. Where domain size explains the number of possible agreement outcomes and the opposition describes the scarcity of issues and it gives some indication how agreements are shaped with regards to fairness [1]. Depending on the domain size and competitiveness of a domain, a negotiation agent may perform better in one or the other, hence we exposed the DOLOS agent to other domains NiceOrDie, ItexvsCypress and Smart-grid, which all are considered to be very competitive but the number of issues and the total domain size varies greatly, as seen in *table I.*

Name	No. issues	Domain size	Opposition
Amsterdam	6	3024	Strong
Itex vs Cypress	4	180	Strong

TABLE I: DOMAIN DESCRIPTION IN TERMS OF METRICS [1]

C. Alternate Acceptance Strategy

In order to measure the performance of individual components of the BOA-structured DOLOS agent, we performed the following tests.

- DOLOS[D] with the acceptance strategy of the Yushu agent.
- DOLOS with the acceptance strategy of IAMHaggler2011 agent[IaMH11].
- DOLOS with the acceptance strategy of BRAMAgent agent.
- DOLOS with the acceptance strategy of AgentK 2.

D. Alternate Opponent Model

- DOLOS with the oppenent model IAMHaggler Bayesan model..
- DOLOS with the oppenent model Smith Frequency model v2[Smith].
- DOLOS with the oppenent model Perfect model.
- DOLOS with the oppenent model AgentLG model

IV. RESULTS

A. DOLOS vs ANAC2011 agents

1) Party Domain:

Performance Measure	DOLOS - HH	DOLOS - IAmH11
Avg. Dist. to Pareto	0,02958	0,00000
Avg. Utility DOLOS	0,847	0,893
Avg. Utility opp.	0,901	0,772
Wins (out of 6)	3(3 no aggr.)	6
Pareto reached	1/6	5/6

TABLE II: PARTY DOMAIN: PERFORMANCE METRICS OF DOLOS VS HARDHEADED & DOLOS VS IAMH11

Performance Measure	DOLOS - Gahboninho	DOLOS - DOLOS
Avg. Dist. to Pareto	0,00000	0,00000
Avg. Utility DOLOS	0,822	0,854
Avg. Utility opp.	0,949	0,865
Wins (out of 6)	0(3 no aggr.)	NA
Pareto reached	3/6	6/6

TABLE III: PARTY DOMAIN: PERFORMANCE METRICS OF DOLOS VS GAHBONINHO & DOLOS VS DOLOS

B. Alternate Acceptance Strategy

Performance Measure	D(Yushu-AS)-IAmH11	D (IAmH11-AS) IAmH11
Avg. Dist. to Pareto	0,000	0,00000
Avg. Utility DOLOS	0,878	0,941
Avg. Utility opp.	0,738	0,666
Wins (out of 6)	6	6
Pareto reached	6/6	6/6

Performance Measure	D(BRAM-AS)-IAmH11	D(Agent K) - IAmH11
Avg. Dist. to Pareto	0,090	0,000
Avg. Utility DOLOS	0,800	0,941
Avg. Utility opp.	0,758	0,666
Wins (out of 6)	6	6
Pareto reached	4/6	6/6

C. Alternate Opponent Model

Performance Measure	D(IAMH-OM)-IAmH11	D(Smith) - IAmH11
Avg. Dist. to Pareto	0,000	0,000
Avg. Utility DOLOS	0,933	0,928
Avg. Utility opp.	0,871	0,717
Wins (out of 6)	6	6
Pareto reached	6/6	6/6

Performance Measure	D(P-OM)-IAmH11	D(Nash-OM) - IAmH11
Avg. Dist. to Pareto	0,000	0,000
Avg. Utility DOLOS	0,785	0,8483
Avg. Utility opp.	0,904	,856
Wins (out of 6)	2	4
Pareto reached	6/6	6/6

D. Behaviour Under Preference Uncertainty

Performance Measure	10	20	50
Avg. Dist. to Pareto	0,320	no aggr.	0.123
Avg. Utility DOLOS	0,582	no aggr.	0.849
Avg. Utility opp.	0,569	no aggr.	0.566
Wins (out of 6)	4	no aggr.	6
Pareto reached	0	0	0

E. Cross-Domain Performance

1) Itex vs Cypress:

Performance Measure	DOLOS - HH	DOLOS - IAmHaggler2011
Avg. Dist. to Pareto	NA	0,00000
Avg. Utility DOLOS	NA	0,820
Avg. Utility opp.	NA	0,480
Wins	No Aggr.	1/1
Pareto reached	NA	1/1

TABLE IV: ITEXVSCYPRESS DOMAIN: PERFORMANCE METRICS OF DOLOS VS HARDHEADED & DOLOS VS IAMHAGGLER2011

Performance Measure	DOLOS - Gahboninho	DOLOS - DOLOS
Avg. Dist. to Pareto	NA	0,00000
Avg. Utility DOLOS	NA	0,941
Avg. Utility opp.	NA	0,305
Wins (out of 6)	No Aggr.	NA
Pareto reached	NA	1/1

TABLE V: ITEXVSCYPRESS DOMAIN: PERFORMANCE METRICS OF DOLOS VS GAHBONINHO & DOLOS VS DOLOS

2) Amsterdam domain:

Performance Measure	DOLOS - HH	DOLOS - IAmHaggler2011
Avg. Dist. to Pareto	0,000	0,000
Avg. Utility DOLOS	0,676	0,920
Avg. Utility opp.	0,942	0,743
Wins	0/1	1/1
Pareto reached	1/1	1/1

TABLE VI: ENERGY-GRID DOMAIN: PERFORMANCE METRICS OF DOLOS VS HARDHEADED & DOLOS VS IAMHAGGLER2011

Performance Measure	DOLOS - Gahboninho	DOLOS - DOLOS
Avg. Dist. to Pareto	0,000	0,000
Avg. Utility DOLOS	0,676	0,920
Avg. Utility opp.	0,942	0,743
Wins	0/1	NA
Pareto reached	1/1	1/1

TABLE VII: AMSTERDAM DOMAIN: PERFORMANCE METRICS OF DOLOS VS GAHBONINHO & DOLOS VS DOLOS

V. FUTURE PERSPECTIVES

If our agent is to become capable of supporting or taking over negotiations performed by humans in real-life negotiations, several changes need to be made. Firstly, it needs to be able to communicate with humans in settings where there is a mix between automated and human agents negotiating with each other. To be able to do this, it must be able to justify its actions to other agents, for example by explaining that it cannot offer a certain bit because it is illegal or because there is a shortage concerning a particular product. [6] [2] Optimally, the agent would be able to differentiate between different levels of automated agents and humans, because the justification it needs to use is dependent on the worldview of the entity the agent is interacting with.

In addition, the agent needs to gain to the ability to persuade other agents and to allow avenues through which the human it is supporting can be persuaded. Currently, the preferences of both agents in the bidding strategy are fixed. However, in real-life negotiations, agents sometimes change preferences. [2] If the agent is able to persuade another agent to change its preferences, much more utility can be gained from a negotiation, because the preferences align more. If the preferences come to align perfectly, a utility of 1 can be obtained for both agents. To allow for human agency, an automated agent should in certain cases communicate attempts at persuasion by the other party to the human it is supporting. This ties into the possibility of value change on the part of humans and society. Sociotechnical systems, consisting of the automated agent and the human it is supporting or taking over for, need to account for value change. That means that the agent must be able to take into account the values of the human. For example, an agent should restrict itself to never deceiving another agent if the human it is supporting sees deception as morally wrong. In addition, the agent needs to be able to react to changing values of the human it is supporting and value change in society as a whole. For example, in the past sustainability was not as important a value as it is now. As a consequence, energy technology was not designed to meet sustainability goals and is now very difficult to change because of the static nature of the technology and how it is embedded in our society. [9] Only recently research has started to look into a theory for value change in sociotechnical systems. [7] Taking into account societal values also contributes to maintaining a positive reputation, the importance of which we shall subsequently describe.

The last example of an important aspect of negotiations that is not yet taken into account is reputation. Currently, our negotiator is only prepared for one-shot games, where their reputation is is solely based on the interactions during the negotiation (the opponent model). However, if the agent is used in real-life negotiations, we can imagine a situation where agents communicate with each other about the behaviour of other agents. Agents might then be more willing to start negotiating with for example tit for tat type agents and less willing to start negotiating with hard-headed agents. So our agent needs to take into account how its actions influence its reputation, based on how other agents perceive the desirability of negotiation with our agent (when there is a choice in whom to negotiate with). [8] In some cases, the environment the agent operates in punishes a bad reputation. [5] describes wireless ad hoc networks that use reputation mechanisms to punish nodes that act selfishly by (threatening to) partially or totally disconnecting them from the network. The agent has to be prepared to operate in these kinds of environments.

VI. Final Remarks by Group Coordinator

In general writing, the report and building the negotiating agent went relatively smoothly. The first two assignments went well, with the group generally being able to work on the project together during class and sometimes delegating some tasks to individual team members to work on at home. For instance, for the second assignment, we all read all the articles, but each team member only summarized two to three papers (thereby ensuring we had a summary of each article in the end).

Right from the start of the third assignment, our team strength took diminished somewhat due to the fact that Stefans French grandfather passed away. Stefan had to leave for France (a 10-hour drive) to be able to attend the cremation and help his grandmother with administrative tasks surrounding the death of her husband. This led to him being away for quite a while, and when he was back he had to get back up to speed for many aspects in his life. Therefore, his contribution to the third part of the deliverable was reduced.

Despite the hurdle surrounding team strength, we quickly started with extra meetings to work on the project. We first started with building the agent, which was quite a challenge. Although the design was already finished for the second assignment, we quickly figured out that we would continuously bring changes to the model. It was interesting to experience level-k theory in practice: all of our decisions were based on our changing beliefs about the likelihood of certain choices by other teams. For example, at one point we contemplated implementing an agent that did not try to optimize its own utility, but solely the ratio between the utility it got and that of the other agent. However, we surmised that this would only be viable if at least a third of the other teams also implemented this strategy and that it was too unlikely that this scenario would occur.

In Stefans absence, Marco made sure that the project got started on. He created a new project on Github and a class with empty methods that the team could fill. In general, he and Rens took over the coordinating efforts of Stefan. Joris and Priyanka implemented and optimized the bidding strategy. Marco and Rens worked on the preference uncertainty, with help from Joris and Priyanka when needed. In addition, Stefan and Rens worked out the performance in cross-domains, and Stefan looked at future perspectives, incorporating both the recommended literature and findings from other areas of expertise. At the last moment, Joris cracked the linear optimisation problem we had been struggling with for weeks. For the most part, however, it is quite difficult to assign individual persons to specific aspects of the project. More often than not, the whole team got together to work on the agent or the report. We did not only meet before or after class but also on days where we were all free to come to the university. We worked in cubicles, often connecting one of our laptops to the larger screen that is available in most cubicles. That way, we could, for example, all see how the agent fared against other agents, suggest possible improvements and subsequently discuss the merit of the suggestions. We would then implement some changes, test again and continue the discussion. This would continue over several iterations during an afternoon session.

I (Stefan) recommend that the four other group members receive a significantly higher grade than I, as they also contributed significantly more to the project. I also hope that you take into consideration that group 11 managed to produce a more than capable agent and extensive report despite the hurdles and reduced team strength. It is a testimony to their skill and strength of character that they themselves never contemplated going to the supervisors to complain or ask for an extension.

VII. Final Remarks by the Rest of the Team

The Rest of the team (Joris, Marco, Priyanka and Rens) are more than happy to share the final grade with Stefan!

Appendix

A. Code

https://github.com/Minas94/group11_MAS.

References

[1] Williams, C. R., Robu, V., Gerding, E. H., & Jennings, N. R. (2014). *An overview of the results and insights from the third automated negotiating agents competition (ANAC2012).* In Novel Insights in Agent-based Complex Automated Negotiation (pp. 151-162). Springer, Tokyo.

[2] Baarslag T., Hindriks K., Hendrikx M., Dirkzwager A. & Jonker C. (2014) *Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies.* In: Marsa-Maestre I., Lopez-Carmona M., Ito T., Zhang M., Bai Q., Fujita K. (eds) Novel Insights in Agent-based Complex Automated Negotiation. Studies in Computational Intelligence, vol 535. Springer, Tokyo.

[3] Baarslag,T. *What to Bid and When to Stop.* https://doi.org/10.4233/uuid:3df6e234-a7c1-4dbe-9eb9-baadabc04bca

[4] Fujita K. et al. (2013) *The Second Automated Negotiating Agents Competition (ANAC2011).* In: Ito T., Zhang M., Robu V., Matsuo T. (eds) Complex Automated Negotiations: Theories, Models, and Software Competitions. Studies in Computational Intelligence, vol 435. Springer, Berlin, Heidelberg.

[5] Jaramillo, J. J., & Srikant, R. (2010). *A game theory based reputation mechanism to incentivize cooperation in wireless ad hoc networks.* Ad Hoc Networks, 8(4), 416429. https://doi.org/10.1016/j.adhoc.2009.10.002

[6] Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M. J., & Sierra, C. (2001). *Automated negotiation: prospects, methods and challenges. Group Decision and Negotiation*, 10(2), 199-215.

[7] Van de Poel, I. (n.d.). *Design for Changing Values.* Retrieved April 4, 2019, from https://www.valuechange.eu

[8] Wang, C., Wang, L., Wang, J., Sun, S., & Xia, C. (2017). *Inferring the reputation enhances the cooperation in the public goods game on interdependent lattices.* Applied Mathematics and Computation, 293, 18-29.

[9] Delft Design for Values Institute. (n.d.). *Design for Changing Values: A Theory of Value Change in Sociotechnical Systems.* (ERC Advanced Grant, 2018-2023). Retrieved April 4, 2019, from http://designforvalues.tudelft.nl/projects/design-for-changing-values/