D

ANAC 2011

Eighteen teams from seven different institutes and six different countries submitted negotiating agents to the 2011 tournament. The qualifying round results are the average over all 18 scenarios, which were submitted by the participants. Eight of these teams continued to the finals after undergoing a qualifying round (see Table D.1).

Table D.2 shows the scores of the 8 finalists of the tournament on the 8 scenarios submitted by all finalists.

D.1 Agents

In this section, we provide, in alphabetical order, brief descriptions of the individual strategies of the finalists of ANAC 2011 based on descriptions of the strategies provided by the teams.

Agent K2

This agent is identical to *Agent K* [131], winner of the ANAC 2010 competition. When creating a counter offer *Agent K* calculates a target utility U_t based on the previous offers made by the opponent and the time that is still remaining in the negotiation. *Agent K* then makes random bids above the target utility. If no such bid can be found, the target utility is lowered to allow for more offers. The target utility U_t at time t is calculated using the following formula:

$$U_t = 1 - (1 - E_{\max}(t)) \cdot t^{\alpha}, \tag{D.1}$$

where $E_{\max}(t)$ is the estimated maximum value the opponent will present in the future based on the average and variance of previous bids, and α is a parameter

Rank	Score	Agent	Affiliation
1	0.756	*Gahboninho*	Bar-Ilan University
2	0.708	*HardHeaded*	Delft University of Technology
3	0.706	*ValueModelAgent*	Bar-Ilan University
4	0.702	*Agent K2*	Nagoya Institute of Technology
5	0.701	*IAMhaggler2011*	University of Southampton
6	0.690	*BRAMAgent*	Ben-Gurion University
7	0.686	*Nice Tit for Tat Agent*	Delft University of Technology
8	0.685	*The Negotiator*	Delft University of Technology
9	0.678	*GYRL*	Ben-Gurion University
10	0.671	*WinnerAgent*	Ben-Gurion University
11	0.664	*Chameleon*	University Politehnica of Bucharest
12	0.648	*SimpleAgentNew*	Ben-Gurion University
13	0.640	*LYYAgent*	Bar-Ilan University
14	0.631	*MrFriendly*	Delft University of Technology
15	0.625	*AgentSmith*	Bar-Ilan University
16	0.623	*IAMcrazyHaggler*	University of Southampton
17	0.601	*DNAgent*	Universidad de Alcala
18	0.571	*ShAgent*	Bar-Ilan University

Table D.1: Scores and affiliation of every strategy in the qualifying round of ANAC 2011.

				95% Confidence Interval	
Rank	Agent Strategy	Mean	Standard Deviation	Lower Bound	Upper Bound
1	*HardHeaded*	0.749	0.0096	0.745	0.752
2	*Gahboninho*	0.740	0.0052	0.738	0.742
3	*IAMhaggler2011*	0.686	0.0047	0.685	0.688
4*	*Agent K2*	0.681	0.0047	0.679	0.683
5*	*The Negotiator*	0.680	0.0043	0.679	0.682
6*	*BRAMAgent*	0.680	0.0050	0.678	0.682
7*	*Nice Tit for Tat Agent*	0.678	0.0076	0.675	0.681
8	*ValueModelAgent*	0.617	0.0069	0.614	0.619

Table D.2: Tournament results in the final round of ANAC 2011.

which controls the concession speed.

Agent K uses quite a sophisticated acceptance mechanism, where it will use the average and variations of the previous bid utilities presented by the opponent to determine the best possible bid it can expect in the future. It will either accept or reject the offer based on the probability that the opponent will present a better offer in the future. If it has already received an offer from the opponent with the same utility or higher, it will offer that bid instead.

BRAMAgent

This agent uses opponent modeling in an attempt to propose offers which are likely to be accepted by the opponent. Specifically, its model of the opponent stores the frequency with which each value of each issue is proposed. This information is maintained only over the 10 most recent offers received from the opponent. Therefore, the first 10 offers *BRAMAgent* makes will be its preferred bid (the one which maximizes its utility), while it gathers initial data for its opponent model.

It also uses a time-dependent concession approach, which sets a threshold at a given time. In each turn, *BRAMAgent* tries to create a bid that contains as many of the opponent's preferred values as possible (according to its opponent model), with a utility greater than or equal to the current threshold. If *BRAMAgent* fails to create such a bid, a bid will be selected from a list of bids that was created at the beginning of the session. This list contains all of the possible bids in the scenario (or all the bids it managed to create in 2 seconds), sorted in descending order according to the utility values. *BRAMAgent* chooses randomly a bid that is nearby the previous bid that was made from that list.

BRAMAgent will accept any offer with utility greater than its threshold. The threshold, which affects both acceptance and proposal levels, varies according to time. Specifically, the threshold levels are set as pre-defined, fixed percentages of the maximum utility that can be achieved (0-60 seconds: 93% of the maximum utility, 60-150 seconds: 85%, 150-175 seconds: 70%, 175-180 seconds: 20%).

Gahboninho

This agent uses a meta-learning strategy that first tries to determine whether the opponent is trying to learn from its own concessions, and then exploits this behavior. Thus, during the first few bids, *Gahboninho* steadily concedes to a utility of 0.9 in an attempt to determine whether or not the opponent is trying to profile the agent. At the same time, the agent tries to assert selfishness and evaluate whether or not the opponent is cooperative. The degree of the opponent's selfishness is estimated based on the opponent's proposals. Then, the more the opponent concedes, the more competitive *Gahboninho*'s strategy becomes. The opponent's willingness to concede is estimated based on the size of variance of the opponent's proposals. After this phase, if the opponent is deemed concessive or adaptive, the agent takes a selfish approach, giving up almost no utility. However, if the opponent asserts even more hard-headedness, it adapts itself to minimize losses, otherwise it risks breakdown in the negotiation (which has very low utility for both parties). In generating the bids, the agent calculates its target, U_t at time t as follows:

$$U_t = U_{\max} - (U_{\max} - U_{\min}) \cdot t \tag{D.2}$$

where $U_{\max}$ and $U_{\min}$ are the maximum and minimum utilities (respectively) in the opponent's bidding history. $U_{\max}$ depends on the opponent's selfishness and the discount factor. Unlike many of the other agents, rather than using a model of the opponent to determine the offer to propose at a given utility level, *Gahboninho* uses a random search approach. Specifically, the agent proposes a random offer above the target utility $T(t)$. The benefit of this approach is that it is fast, therefore, given the format of the competition, a very large number of offers can be exchanged, allowing greater search of the outcome space. Moreover, the agent suggests using the opponent's best bid if the time is almost up.

HardHeaded

In each negotiation round, *HardHeaded* considers a set of bids within a pre-defined utility range which is adjusted over time by a pre-specified, monotonically decreasing function. A model of the opponent's utility function is constructed by analyzing the frequency of the values of the issues in every bid received from the opponent. From a set of bids with approximately equal utility for the agent itself, the opponent model is used to suggest bids that are best to the opponent in order to increase chances of reaching an agreement in a shorter period of time.

The concession function specifies an increasing rate of concession (i.e. decreasing utility) for the utility of the agent's bids. The function has non-monotonic curvature with one inflection point, determined by the discount factor of the scenario. This function is determined by tuning the strategy based on the sample scenarios and data made available before the competition. For the scenarios with time discounting, the timeline is split into two phases over which the agent practices different strategies: it starts by using a *Boulware* strategy, and after a certain amount of time has passed (depending on the discount factor), it switches to a *Conceder* strategy [82].

IAMhaggler2011

This agent uses a Gaussian process regression technique to predict the opponent's behavior [260]. It then uses this estimate, along with the uncertainty values provided by the Gaussian process, in order to optimally choose its concession strategy. In so doing, the concession strategy considers both the opponent's behavior and the time constraints.

The concession strategy is then used to determine the target utility at a given time. In the concession strategy, the agent finds the time, t^*, at which the expected discounted utility of the opponent's offer is maximized. In addition, it finds the utility level, u^*, at which the expected discounted utility of our offer is maximized. The agent then concedes towards $[t^*, u^*]$, whilst regularly repeating the Gaussian process and maximizations.

Finally, having chosen a target, the agent proposes an offer which has a utility close to that target. In choosing the bids, *IAMhaggler2011* uses an approach similar to that of *Gahboninho*. Specifically, a random package, with utility close to the target is selected according to the concession strategy. This strategy is a fast process, which allows many offers to be made and encourages the exploration of outcome space.

Nice Tit for Tat Agent

This agent plays a tit-for-tat strategy with respect to its own utility. The agent will initially cooperate, then respond in kind to the opponent's previous action, while aiming for the Nash point in the scenario. If the opponent's bid improves its utility, then the agent concedes accordingly. The agent is nice in the sense that it does not retaliate. Therefore, when the opponent makes an offer which reduces the agent's utility, the *Nice Tit for Tat Agent* assumes the opponent made a mistake and does nothing, waiting for a better bid. This approach is based on [106]. *Nice Tit for Tat Agent* maintains a Bayesian model [112] of its opponent, updated after each move by the opponent. This model is used to try to identify Pareto optimal bids in order to be able to respond to a concession by the opponent with a nice move. The agent will try to mirror the opponent's concession in accordance with its own utility function.

The agent detects very cooperative scenarios to aim for slightly more than Nash utility. Also, if the domain is large, if the discount factor is high, or if time is running out, the agent will make larger concessions towards its bid target. The agent tries to optimize the opponent's utility by making a number of different bids with approximately this bid target utility.

The Negotiator

Unlike the other finalist agents, this agent does not model the opponent. Its behavior depends on the mode it is using, which can be either: DISCOUNT or NODISCOUNT. A negotiation starts with the agent using its NODISCOUNT mode, which results in hardheaded behavior. After a predetermined time period, the agent switches to its DISCOUNT mode, in which its behavior becomes more concessive.

The main difference between the different modes is in the speed of descent of the minimum threshold for acceptance and offering. In the NoDiscount mode, most time is spent on the higher range of utilities and only in the last seconds are the remaining bids visited. The Discount mode treats all bids equally and tries to visit them all. An opponent's offer is accepted if it is above the current minimum threshold. An offer should also satisfy the minimum threshold, however a dynamic upper-bound is used to limit the available bids to offer in a turn. In 30% of the cases this upper-bound is ignored to revisit old bids, which can result in acceptance in later phases of the negotiation.

Finally, *The Negotiator* attempts to estimate the number of remaining moves to ensure that it always accepts before the negotiation deadline.

ValueModelAgent

This agent uses temporal difference reinforcement learning to predict the opponent's utility function. The particular learning technique is focused on finding the amount of utility lost by the opponent for each value. However, as the bid (expected) utilities represent the decrease in all issues, a method is needed to decide which values should change the most. To achieve this, the agent uses estimations of standard deviation and reliability of a value to decide how to make the split. The reliability is also used to decide the learning factor of the individual learning. The agent uses a symmetric lower-bound to approximate the opponent's concession (if the opponent makes 100 different bids, and the 100th bid is worth 0.94, it is assumed the opponent conceded at least 6%). These parameters were determined in advance, based on average performance across a set of scenarios available for testing before the competition.

In more detail, *ValueModelAgent* starts by making bids which lie in the top 2% of the outcome space. It severely limits the concession in the first 80% of the timeline. If there is a large discount, the agent compromises only as much as its prediction of the opponent's compromise. If there is no discount, the agent does not concede as long as the opponent is compromising. If the opponent stops moving, the agent compromises up to two thirds of the opponent's approximated compromise. As the deadline approaches (80%-90% of the time has elapsed), the agent compromises up to 50% of the difference, providing that the opponent is still not compromising. Once 90% of the time has elapsed, the agent sleeps and makes the "final offer", if the opponent returns offers the agent sends the best offer that has been received from the opponent (accepting his last offer only if its close enough). *ValueModelAgent* has a fixed lower limit on its acceptance threshold, of 0.7. Therefore it never accepts an offer with an undiscounted utility lower than this value.