Agent 7: Time-Dependent, Random and Frequency-Based Bidding with Lower Limit of Concession

Julio Cesar Aguilar Jimenez
School of Electronics and Computer Science
MSc. Artificial Intelligence
University of Southampton
Student ID. 29312175
jcaj1n17@soton.ac.uk

1. ABSTRACT

In this paper we present the *design* of **Agent 7**, the *reasons* behind its model and an *analysis* of the results obtained in the *ANAC* internal competition, demonstrating a *great performance* in the different *domains* which ranked it at the first place of the University competition.

2. INTRODUCTION

An *intelligent agent* (**IA**) is an autonomous entity which observes through *sensors* and acts upon an environment. It directs its activity towards achieving goals.

In this paper we will explain how a *time-dependent*, *random and frequency-based bidding* with *lower limit of concession* agent was developed with the purpose of compete in a multiparty negotiation tournament. The **Agent 7** is able to obtain a *high* **utility** while seeking to benefit all parties involved in different domain sizes.

3. STRATEGY

Agent 7 has a negotiation strategy in 4 steps. *First* it calculates the table of **disutility**, then compute the value for the **threshold** function.

As a *third* step, perform a **combined modelling** of the opponent and finally proceed to **generate** the *bid*.

3.1 Disutility Table

Agent 7 derives a relative utility *matrix* according to our agent's utility space as suggested by the Atlas3 paper.[1] This allows us to iterate through *issues* and *values*, and add issues to our *bid* only if they do not take our agent's utility below our disutility **threshold** function $\boldsymbol{T(t)}$.

We can consider a matrix of *issues* $i_1 ... i_n$ with their respective *values* for each issue $v_{i_k,1}$ ... $v_{i_k,n}$.

$$\begin{array}{c} \boldsymbol{i_1} \quad \boldsymbol{i_2} \quad \boldsymbol{i_3} \quad \quad \boldsymbol{i_n} \\ \begin{bmatrix} v_{1,1} & v_{2,1} & v_{3,1} & \dots & v_{n,1} \\ v_{1,2} & v_{2,2} & v_{3,2} & \dots & v_{n,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{1,n} & \dots & \dots & \dots & v_{n,n} \end{bmatrix} \end{array}$$

The *disutility* matrix for **Agent 7** is created at the start of the negotiation using Equation (1), and should not be altered at any point following creation.

$$\boldsymbol{u}_{i,j} = \frac{v_{i,j} - \arg\max\{v_i\}}{\arg\max\{v_i\}} \times w_i \tag{1}$$

Where $\boldsymbol{u}$ is our *disutility* function for the corresponding *value* (i, j), v is the value j for each issue i and w the *weight* for that issue. To facilitate the comprehension, we will consider 3 issues and 4 values as shown below.

Table 1: Disutility function calculation

	$\boldsymbol{i_1, w = 0.5}$	$\boldsymbol{i_2, w = 0.4}$	$\boldsymbol{i_3, w = 0.1}$
v_1	$\frac{10-10}{10} \times 0.5 = \mathbf{0}$	$\frac{2-5}{5} \times 0.3 = \mathbf{-0.18}$	$\frac{1-4}{4} \times 0.2 = \mathbf{-0.15}$
v_2	$\frac{8-10}{10} \times 0.5 = \mathbf{-0.1}$	$\frac{3-5}{5} \times 0.3 = \mathbf{-0.12}$	$\frac{4-4}{4} \times 0.2 = \mathbf{0}$
v_3	$\frac{5-10}{10} \times 0.5 = \mathbf{-0.25}$	$\frac{5-5}{5} \times 0.3 = \mathbf{0}$	$\frac{2-4}{4} \times 0.2 = \mathbf{-0.1}$
v_4	$\frac{2-10}{10} \times 0.5 = \mathbf{-0.4}$	$\frac{5-5}{5} \times 0.3 = \mathbf{0}$	$\frac{3-4}{4} \times 0.2 = \mathbf{-0.05}$

This translation makes our bid generation algorithm runs faster. By means of this method, *computational* complexity is $O(n)$, where n means the number of *issues*.

3.2 Threshold Function T(t)

The *Threshold* function $\boldsymbol{T(t)}$ plays a crucial role for the good *performance* of the agents during the negotiation. Almost every agent in every paper studied uses a *threshold* function.

Agent 7 uses *time-dependent* strategy. In time-dependent strategy[2] a real number $T(t)$, called *target utility*, is computed as Equation (2).

$$T(t) = \alpha \pm t \times \beta \tag{2}$$

In which, t, α and β are *time*, the *utility* and the *value* to control the *concession speed* of the agent, respectively. The approach is similar to that used by agent *Pars* [2], but more complex since it has not a **Boulware** behaviour.

Agent 7 sets α and β based on *heuristics* where the *values* were obtained through *multiple* tests against different *agents* in the **Genius** environment, the function presents a multilinear behaviour. The Expression (3) shows the α, *sign* and β values for each *time* range.

$$T(t) = \begin{cases} a)\ \alpha = 1, & -\beta = 0.4 & \text{if } t \leq 0.25 \\ b)\ \alpha = 0.9, & +\beta = 0.4 & \text{if } 0.25 < t \leq 0.375 \\ c)\ \alpha = 0.95, & -\beta = 0.4 & \text{if } 0.375 < t \leq 0.5 \\ d)\ \alpha = 0.9, & -\beta = 1 & \text{if } 0.5 < t \leq 0.6 \\ e)\ \alpha = 0.8, & +\beta = 2 & \text{if } 0.6 < t \leq 0.7 \\ f)\ \alpha = 1, & -\beta = 3 & \text{if } 0.7 < t \leq 0.8 \\ g)\ \alpha = 0.7, & +\beta = 1 & \text{if } 0.8 < t \leq 0.9 \\ h)\ \alpha = 0.8, & -\beta = 6 & \text{if } 0.9 < t \leq 0.95 \\ i)\ \alpha = 0.5, & +\beta = 4 & \text{if } t > 0.95 \end{cases} \quad (3)$$

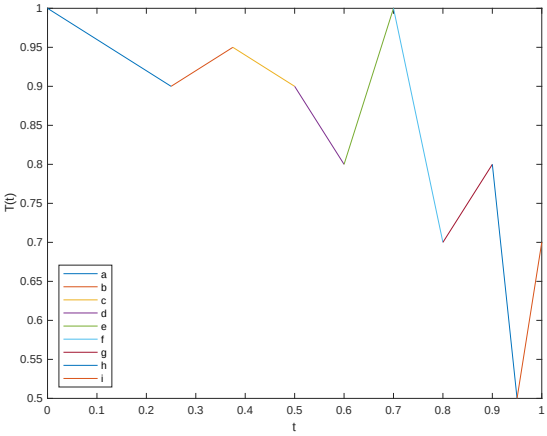

Figure 1: Utility Threshold Values

In the first half **Agent 7** only makes bids above the 0.9 of the utility, then drops to 0.8 and goes up to 1 in 70% of the time.

In 90%, fearing there will not be *agreement*, **Agent 7** drastically lower the threshold to 0.5 in 95% of time. In the final stretch will raise to 0.7, so the most *stubborn* agents tend to accept offers and the resulting *utility* will not be low.

3.3 Combined Opponent Model

Agent 7 is *frequency-based* modelling, each time an *opponent* makes a bid, **Agent 7** update the *frequency table* it is holding for that *opponent*.

To explain this process, consider that after 20 offers the *frequency table* of our opponents looks like Table 2.

Table 2: Opponents Frequency Table

Opp1	i_1	i_2	i_3	Opp2	i_1	i_2	i_3
v_1	6	7	4	v_1	0	16	12
v_2	10	12	9	v_2	0	3	6
v_3	3	1	3	v_3	20	1	1
v_4	1	0	4	v_4	0	0	1

Next, we need to generate a $\boldsymbol{\Omega}$ (*a random number from a list*) value, it would be useful to vary which opponent was being conceded to most as *Random Dancer* [3]. The Equation (4) shows how our models are combined.

$$Combined\,Model = \Omega \times Opp1 + (1 - \Omega) \times Opp2 \quad (4)$$

It focuses our *bids* on a plane connecting the bid-spaces of *opponents*, which should be very close to the *Pareto* frontier.

Table 3 shows how to calculate the *combined model* for the first 3 values in i_1 (assuming $\Omega = 0.5$).

Table 3: Combined Opponent Model Table

	i_1	*Res*
v_1	$0.5 \times 6 + (1 - 0.5) \times 0$	3
v_2	$0.5 \times 10 + (1 - 0.5) \times 0$	5
v_3	$0.5 \times 3 + (1 - 0.5) \times 20$	11.5

3.4 Bid Generation

To demonstrate the process we will use $\boldsymbol{T(0.6)} = 0.8$. First randomly sort our list of *issues* [i_2, i_1, i_3] and initialise the disutility of our bid $\boldsymbol{cdu} = 0$.

As shown in Table 1 the lowest value is $\boldsymbol{-0.18}$ and since $-0.18 \leq (0.8 - 1)$ all values are possible. The next step is generate a *cumulative probability* table using Table 3, to do that we just take the values of the table and divided by the number of *bids* **20** and the result will be added to the next *value*.

Applying this for *issue2* we have [$v_1 = 0.575$, $v_2 = 0.95$, $v_3 = 1$, $v_4 = 1$]. Then we generate a random number, it is **0.62**, and take the first value that is bigger than its, so we take v_2.

For *issue1* the lowest disutility after 0 is 0.1, but since $(-0.12 + -0.1) > (0.8 - 1)$, the only possible value is v_1.

For *issue3* there are 2 possible values ($v_2 = 0$, $v_4 = 0.05$), the *cumulative values* will be [$v_2 = 0.75$, $v_4 = 1$]. The random number is **0.26** and we take v_2.

The bid is [i_1v_1, i_2v_2, i_3v_2] and the disutility will be $[-0.12 + 0 + 0]$. The bid has a utility of **0.88** which is pretty good and it is close to a bid that would be accepted by the *opponents* due to the *combined model*.

4. ANALYSIS

Agent 7 was tested in a competition of 3 different domains with other 35 opponents and 3 agents per negotiation. Tables 4, 5, 6 show the results.

Table 4: *Agreement* in different *Domains*

Domain	% Agree.	No Agree.	Agreements
D_0	99.16	23	2708
D_1	95.54	119	2550
D_2	83.12	478	2354

Table 5: *Pareto* and *Nash* distances

Domain	Av. Pareto	Av. Nash
D_0	0.0200	0.1310
D_1	0.1117	0.2052
D_2	0.2003	0.2518

Table 6: *Utility* in different *Domains*

Domain	% Av. Utility
D_0	86.97
D_1	84.29
D_2	70.32

4.1 Small Domain (*Sportshal*)

For the first domain $\boldsymbol{D}_0$, we can see that the number of **agreements** and **utility** is quite *high* and **Nash** distance is 0.131. These are very good *results* and indicate that most of the negotiations reached the agreement in the first half of the *time* or before reaching 80%. Due to our *combined model* the Nash distance is very short. Figures 2, 3 illustrate the aforementioned.

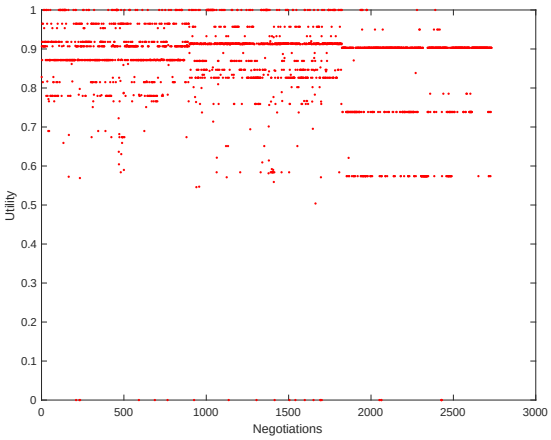

Figure 2: Utility graph for $\boldsymbol{D}_0$

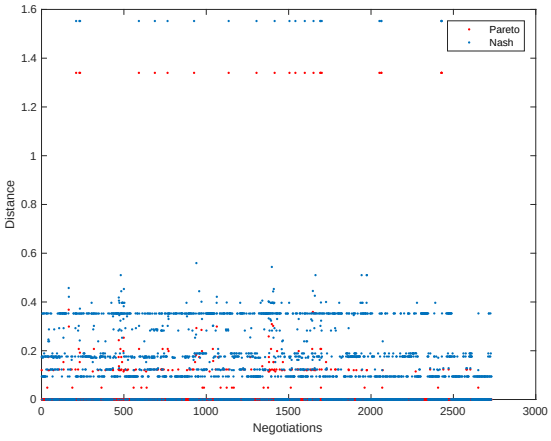

Figure 3: Pareto and Nash distances for $\boldsymbol{D}_0$

4.2 Medium Domain (*WindFarm*)

For $\boldsymbol{D}_1$ the *utility* is above 80%, the **Pareto** and **Nash** distances are small (*1 and 2 tenths*). We can infer looking at Figures 4 and 5 that several of the negotiations in this domain were completed in the *final stretch*, which leads to less *utility*.

There are many negotiations that reach a **utility** of 1, due to the **Threshold** function, it deceives the *opponents* raising the utility in 60% of time. The **Nash** graph is more *scattered*, the agents that have a **random** behaviour for the generation of *bids* as **Agent 7** in *larger* domains lead to this type of behaviour.

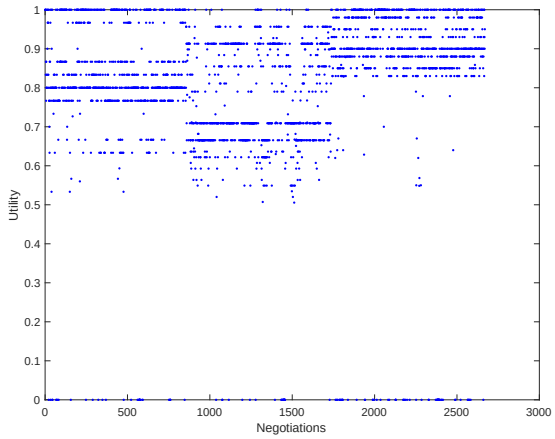

Figure 4: Utility graph for $\boldsymbol{D}_1$

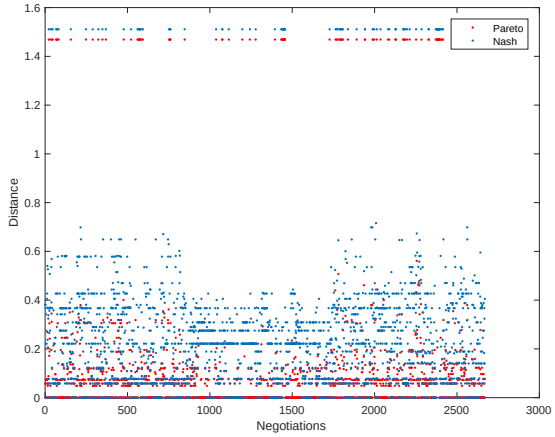

Figure 5: Pareto and Nash distances for $\boldsymbol{D}_1$

4.3 Large Domain (*Politics*)

The domain $\boldsymbol{D}_2$ is where our agent had a *lower performance*, there was a considerable amount of *no agreements* (~ 500). However **Agent 7** got the best Nash distance score among all the agents for this domain.

A critical factor in large *domains* is **time**, if the modelling and strategy of *opponents* are very *complex*, they cause the agent to spend a lot of *time* performing the relevant *calculations*.

Another factor is the agents that tend to seek the greatest utility for themselves and neglect the common wealth. Our **Agent 7** had problems negotiating with agents **2**, **27** and **33**. Although these agents had a *greater* **utility**, their *Nash* distance is much higher than **Agent 7**, which corroborates their *selfish* behaviour.

In Figures 6 and 7 we can see that although the **utility** is more dispersed than the previous ones, the majority is above 70%, the problem is the large number of 0's. On the other hand, the **Nash** distance is very similar to that of $\boldsymbol{D}_1$, which reflects a *coherent behaviour* across the different

domains and looking for a benefit for all the *parties* involved.

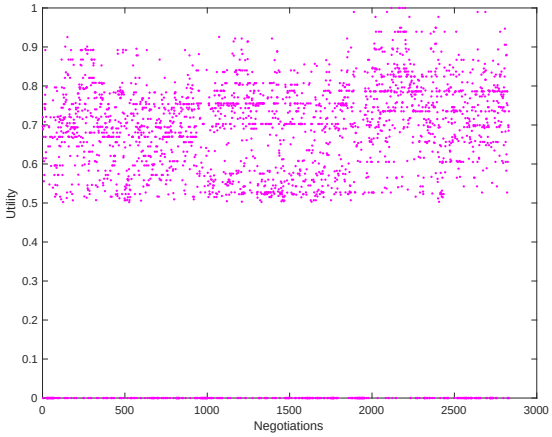

Figure 6: Utility graph for D_2

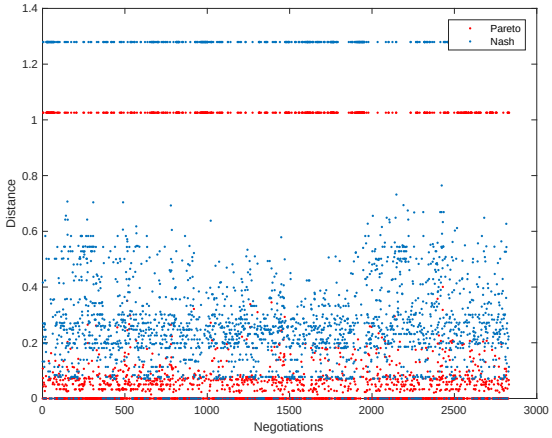

Figure 7: Pareto and Nash distances for D_2

5. FURTHERWORK

One of the possible future *challenges* is to *optimise* the **Threshold** function, make it *intelligent* and change the *minimum* value for that *bid* it has received from *opponents*, one value for each *opponent*.

The **Frequency Table** can also be subject to *optimisation*, we can add a *weight* to those *bids* that are made at the *beginning* or in different *strategic points* that better model the *preferences* of the *opponents*.

Ω can be *improved*, instead of a *list* of *random* values, it can be the product of a *function* that adjusts both *opponents* near the point of *Nash* or *Pareto*, this will give *flexibility* and *dynamism* to our agent.

6. REFERENCES

[1] K. Fujita, Q. Bai, T. Ito, M. Zhang, F. Ren, R. Aydoan, and R. Hadfi. *Modern Approaches to Agent-based Complex Automated Negotiation*, pages 169–173. Springer Publishing Company, Incorporated, 1st edition, 2017.

[2] K. Fujita, Q. Bai, T. Ito, M. Zhang, F. Ren, R. Aydoan, and R. Hadfi. *Modern Approaches to Agent-based Complex Automated Negotiation*, pages 175–183. Springer Publishing Company, Incorporated, 1st edition, 2017.

[3] K. Fujita, Q. Bai, T. Ito, M. Zhang, F. Ren, R. Aydoan, and R. Hadfi. *Modern Approaches to Agent-based Complex Automated Negotiation*, pages 185–189. Springer Publishing Company, Incorporated, 1st edition, 2017.

[4] M. Wooldridge. *An Introduction to MultiAgent Systems*. Wiley Publishing, 2nd edition, 2009.