MultiAgent Systems : AgentPeaK

Pelika Maria 2013030028 Kavroulakis Dimitris 2013030064

February 13, 2019

1 Abstract

In many situations, a form of negotiation can be used to resolve a problem between multiple parties. However, one of the biggest problems is not knowing the intentions and true interests of the opponent. In this report problems like these are trying to be solved in the context of our course "MultiAgent Systems". We studied the obtained utility during several negotiation sessions. Results show that our agent's performance depends not only on the opponent's strategy but also on the domain.

2 Introduction

Automated negotiating agents used to be widely studied in multi-agent systems research.Automated agents can be used side-by-side with a human negotiator embarking on an important negotiation task. There may even be situations in which automated negotiators can replace the human negotiators. Thus, success in developing an automated agent with negotiation capabilities has great advantages and implications.In this paper, we develop a new agent (AgentPeaK) that can negotiate on various negotiation problems.The challenge in developing negotiating agents is to find strategies which react fast at opponent's actions, learn well opponent's preferences, better use the opponent's profile when proposing bids, and use time effectively during negotiation.Like other learning-based reasoning models, at a given negotiation step our agent performs two actions: analyzes the previous action of the opponent and selects and offers the next bid.AgentPeaK reasoning model uses a concession-based strategy which reduces its own utility of the offered bid as the negotiation time passes, in order to avoid the situation that negotiation ends without an agreement.

3 Automated Negotiating Agents Competition(ANAC)

3.1 Purpose of ANAC

The purpose of the competition is to steer research in the area of multilateral negotiations in arbitrary domains.In multilateral negotiation league, entrants will to design and implement an intelligent negotiating agent, which negotiates with two opponents and is able to learn from its previous negotiations. The participants will develop their agents in GENIUS platform.Challenges regarding this league are to design winning strategies for bidding, opponent modeling and bid acceptance strategies when negotiating repeatedly with agents in a multilateral setting.

3.2 Negotiation Platform

Genius allows easy development and integration of existing negotiating agents. It can be used to simulate individual negotiation sessions as well as tournaments between negotiating agents in various scenarios. It allows the specification of negotiation domains and preference profiles by means of a graphical user interface. Therefore, its use is easy and allows developers to concentrate more in implementing a solid agent that fulfills all requirements.

3.3 Issues to overcome

The main issues that had to be solved are two. First, we had to choose a bidding plan affiliated with our agent's characteristics, as well as, an acceptance strategy that gives us the best utility possible. Therefore, research had to be done in order to find the best fitted strategy.

4 Structural Assumptions

Our first assumption is a common one and assumes that the utility of a bid can be computed as a weighted sum of the utilities associated with the values for each issue. Utility functions modelling the preferences of an agent thus are linearly additive functions and are defined by a set of weights wi (or priorities) and corresponding evaluation functions ei(xi) for each of n issues by:

$$u(b_i) = \sum_{i=1}^{n} w_i e_i(x_i \in b_i)$$

where x_i is the value of issue i in bid b_t in the negotiation round t. To ensure that a utility function has a range in [0, 1], the range of the evaluation functions is assumed to be in [0,1] and the weights are assumed to be normalized such that their sum equals 1. In order to learn an opponent's preference profile or utility function U(b) we need to learn both the issue priorities or weights w_i as well as the evaluation functions $e_i(x_i)$. The objective of learning an opponent model thus is to find a model as defined above that is the most plausible candidate or best approximation of the opponent's preference profile.

5 Developing the AgentPeaK

5.1 Searching Strategy

5.1.1 NiceTitforTat

Setting as a point of reference agent NiceTitforTat, searching for an appropriate strategy became an easy task, as AgentPeak's main attribute is to concede whenever is possible. Tit for Tat has been applied and found successful in many other games, including the Iterated Prisoner's Dilemma game [7]. It is considered to be a very robust strategy, mainly because of the following three features:

- it plays nice as long as the opponent plays nice
- it can be provoked into retaliation by a defection of the opponent
- if is forgiving after just one act of retaliation

Moreover, the Nice Tit for Tat 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 of the negotiation scenario. After each move by the opponent, it updates its Bayesian opponent model to make sure it responds with a nice move to a concession by the opponent.

5.1.2 Yushu

Another agent that implies to AgentPeak's policy is agent Yushu.Yushu uses simple heuristics to implement a conservative concession strategy based on a dynamically computed measure of competitiveness and number of negotiations rounds left before the deadline [3]. He does not learn the private information of its opponent given the difficulty of learning in selfish negotiation. Furthermore, he always wants to make an agreement with its opponent, which implies that it makes concessions during negotiation. While making an offer, first computes its reserve utility following the time-dependent strategy. Lastly he uses a simple heuristic to measure a domain's competitiveness: if its opponent's offers always give Yushu a low utility, the domain is more competitive.

5.1.3 Bayesian Opponent Model

reference from bayesian paper In game theory, the class of Bayesian games refers to games in which players do not have complete information about each others' preferences.In this sense, negotiation can be viewed as an instance of a Bayesian game .In such a setting, players can use evidence (or so-called signal functions) to update their beliefs about the other party. During a negotiation at every time t when a new bid b_t is received from the opponent the probability of each hypothesis should be updated using Bayes' rule:

$$P(h_i \mid b_i) = \frac{P(h_j)P(b_i|h_j)}{\sum_{k=1}^{m} P(h_k)P(b_i|h_k)}$$

Here the conditional probability $P(b_t|h_j)$ represents the probability that bid b_t might have been proposed given hypothesis h_j.[2]The normalization factor in the denominator of Bayes' rule ensures that the probability of the entire hypothesis space is 1.The learning approach outlined will increase the probability of a hypothesis about an opponent's preference profile that is most consistent with the bid sequence received so far from that opponent and provides the best match with the utilities of these bids, estimated using the conditional probability distribution associated with tactics. As a result, the more consistent the predicted utility is with a hypothesis, the higher the probability associated with this hypothesis will be.

5.2 Tit for Tat strategy

Tit for tat is a mechanism to a payoff matrix similar to that of a prisoner's dilemma. Tit for tat was introduced by Anatol Rapoport, who developed a strategy in which each participant in an iterated prisoner's dilemma follows a course of action consistent with his opponent's previous turn. For example, if provoked, a player subsequently responds with retaliation; if unprovoked, the player cooperates.

Tit for tat posits that a person is more successful if he cooperates with another person. Implementing a tit for tat strategy occurs when one agent cooperates with

another agent in the very first interaction and then mimics their subsequent moves. This strategy is based on the concepts of retaliation and altruism. When faced with a dilemma, an individual cooperates when another member has an immediate history of cooperating and defaults when the counterparty previously defaulted.[1]

Considering that the strategy can be used in bilateral negotiations, an easy way to bypass that restriction is to deal multilateral negotiations as bilateral. We can do that by processing incoming offers as offers of an individual instead of many agents.

5.3 Implementation

On each step of the implementation, different versions of the agent were created ending up with the final agent. In every approach the agent was tested with different opponents, calculating its final utility. We decided to make simple changes that affect those parameters for a better outcome.

5.3.1 First Approach

The first version of AgentPeak is managing simply the incoming bids, exporting a list of acceptable bids. As first step, the agent saves all incoming bids from each round, picks the higher bid and evaluates it. Calculating its own utility considers that the opponent uses the same utility space he has and depending on the previous higher bid decides whether the opponent is conceding or hitting. Moreover, he selects randomly a bid from the exporting list. A more detailed explanation is shown below.

	JapanTrip	CarDomain	Movie
RandomParty	0.83189	0.49062	1
ConcederNegotiationParty	0.52344	0.61681	1
BoulwareNegotiationParty	0.85526	0.60842	1
AgentPeak	0.76827	0.53838	0.625
distance to pareto	0.09242	0.30893	0
distance to Nash	0.55076	0.42305	0
social welfare	2.97885	2.25422	3.625

Table 1: Four Agents Average Utility on 3 Domains

We tested this implementation in negotiations with 5 different agents on GENIUS in groups of 3,4 and 5. The results were not promising as the agent was not conceding at all. Therefore we needed to find a better solution to make a conceder agent.

5.3.2 Second Approach

The above outcome had raised many questions and we needed to improve not only our offers but also our acceptance strategy. In the second approach AgentPeaK processes the past offers differently. Instead of selecting the highest bid he calculates the average utility of all previous offers and compares it with the incoming. Again the agents decides whether he is going to accept or not, if the upcoming offer's utility is better than the opponent's.

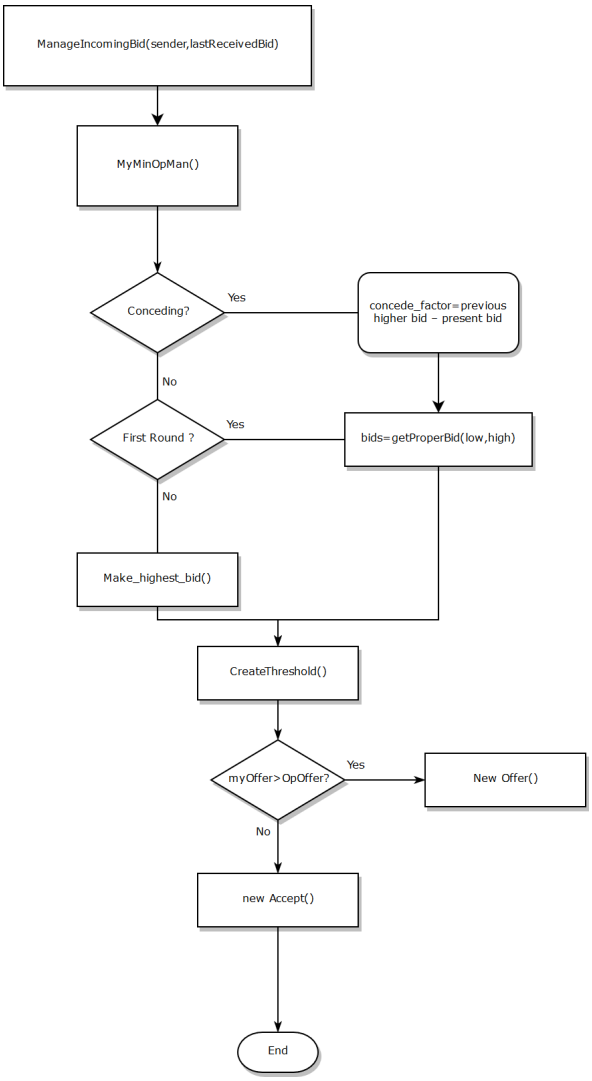


Figure 1: Caption

	JapanTrip	CarDomain	Movie
ConcederNegotiationParty	0.8111	0.80995	0.93258
BoulwareNegotiationParty	0.88375	0.74109	1
AgentPeak	0.62051	0.54147	0.33333
distance to pareto	0	0.09412	0
distance to Nash	0.12044	0.3088	0.53658
social welfare	2.31536	2.09252	2.26591

Table 2: Three Agents Average Utility on 3 Domains

After excessive testing we reached to an unfortunate conclusion as the agent was making unreasonable offers. The bidding strategy seems to be working as we wanted but our acceptance strategy is poor. The above results show that our agent's performance is affected by his position in the negotiation.

5.3.3 Adding Opponent Model

The need of having an opponent model has been created. For that reason we implemented a simple bayesian opponent model as described above. The learning approach has been tested on several domains to demonstrate the effectiveness of the approach. The results moreover showed the effectiveness of using an opponent model in a negotiation strategy to improve the efficiency of the bidding process, however it is not that promising. We tested the agent in several negotiations and concluded that it is not necessary to include this particular opponent model.

	JapanTrip	CarDomain	Movie
ConcederNegotiationParty	0.8111	0.80995	0.93258
BoulwareNegotiationParty	0.88375	0.74109	1
AgentPeak	0.62051	0.54147	0.33333
distance to pareto	0	0.09412	0
distance to Nash	0.12044	0.3088	0.53658
social welfare	2.31536	2.09252	2.26591

Table 3: Opponent Model performance on 3 Domains.

5.3.4 Final Approach

In the final implementation, we have to prove that our agent makes bids wisely, while the urgent need was to find a better acceptance strategy. For that reason, instead of analysing the whole structure of AgentPeaK we are going to point the main differences of the previous implementations.

Bidding Strategy

Opponent's incoming bid is being processed and added to the list "AllIncomingBidUtils", but we first calculate the average utility of all previous incoming bids. Subsequently, we determine the conceding factor.

$concedefactor = \frac{averageUtility - tempUtility}{4}$

FindProperBids(lower bound, upper bound)
This function is responsible to extract an array with all acceptable bids. Before the negotiation starts, during the initialization, we call "findAllmyBidsPossible()" so that we can have an array with all our possible bids on each round. This function is explained later. Lower and upper bounds set the range of the utility of our bids.

GetfinalBid()
In the end of our bidding strategy this function resolves our final Bid. We get to choose from a variable number of bids. For that reason average utility is being calculated and then picks the bid that is closest to this utility.

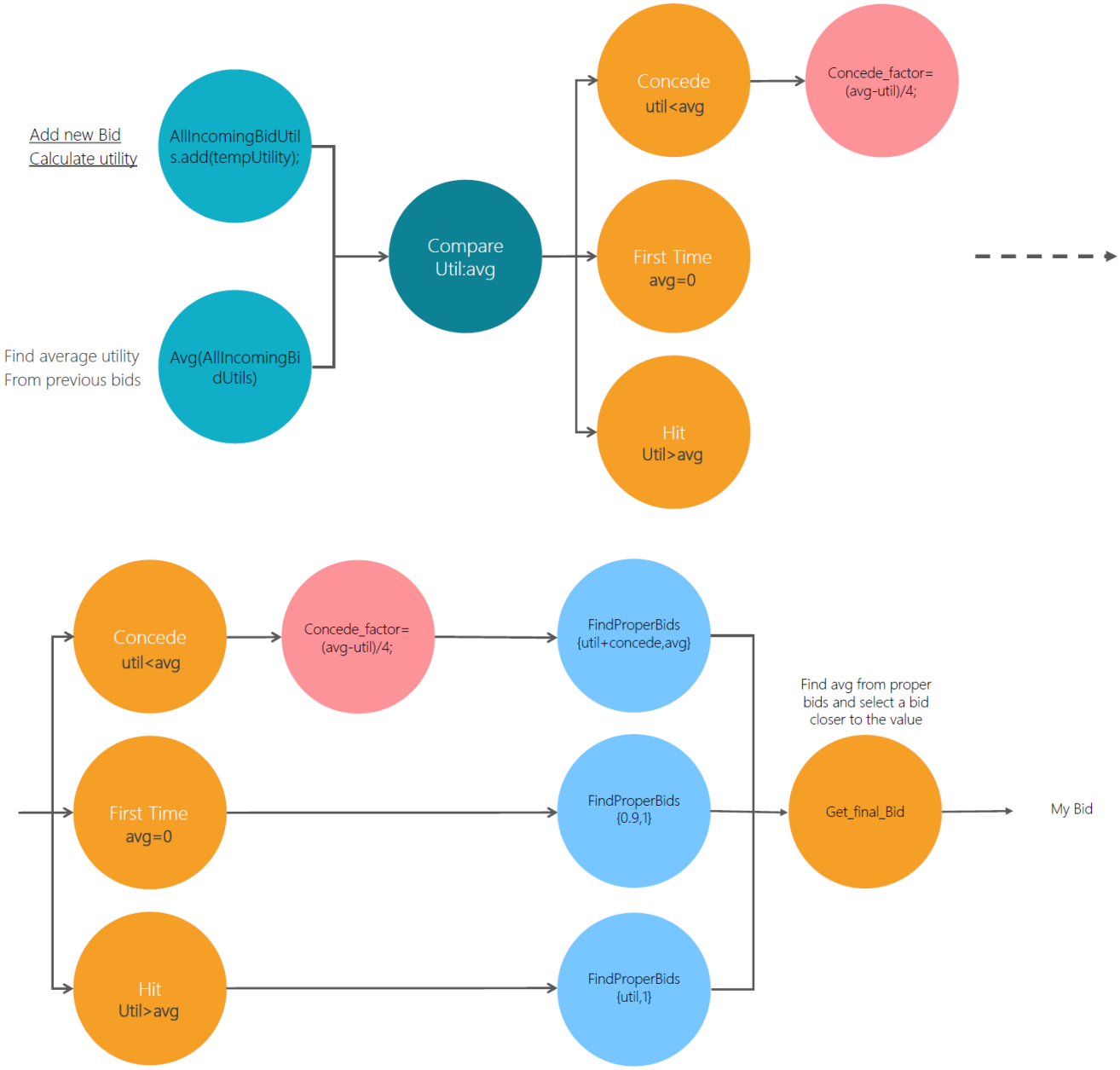


Figure 2: Bidding Strategy

Acceptance Strategy

At this stage the final offer's utility is being recalculated by compromising factor γ. Because in several negotiations there are discount factors, defined in [0, 1], when it is necessary we calculate the discounted utility of outcome from the undiscounted utility U. The following formula gives the requested utility when given the normalized time t.

$$U_D^t(\omega) = U(\omega) \cdot d^t$$

In addition, we consider if our agent has conceded. The main control condition is to compare whether our upcoming utility is higher than the incoming utility. With Boolean "conceded" we assure that our agent will not accept if he has conceded.

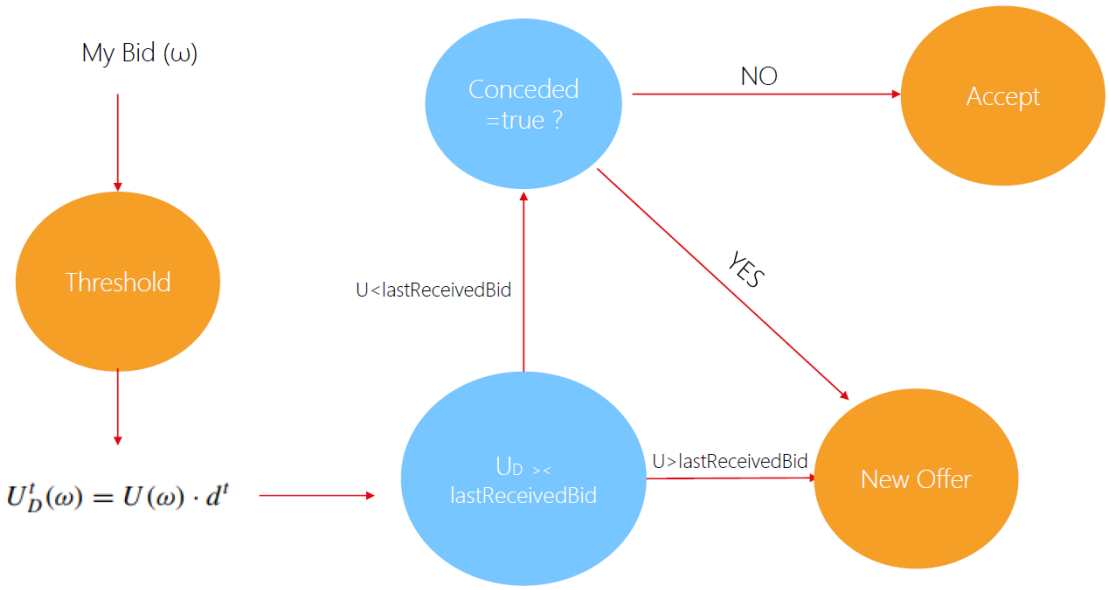


Figure 3: Acceptance strategy

6 Experiments and Results

6.1 Level 1

The agent has been tested in three different domains with 6 different genius' agents. Charts bellow show AgentPeaK's efficiency.

JapanTrip	1o Neg - 3profiles	2o Neg - 3profiles	3o Neg - 3profiles
NumOfAgents	5	4	3
distance to pareto	0.17558	0.13408	0
distance to Nash	0.58102	0.2038	0.70907
social welfare	2.8871	2.66081	2.23492
profiles	(pr1), (pr2), (pr3), (pr1), (pr3)	(pr1), (pr2), (pr3), —, (pr3)	—, (pr2), (pr3), —, (pr1)
RandomParty	0.34954	0.35944	——
ConcederNegotiationParty	0.46303	0.62086	0.51575
BoulwareNegotiationParty	0.8625	0.84026	0.78989
AgentSmith2016	0.34954	——	——
AgentPeak	0.8625	0.84026	0.92928

Table 4: Three different Negotiations on JapanTrip Domain

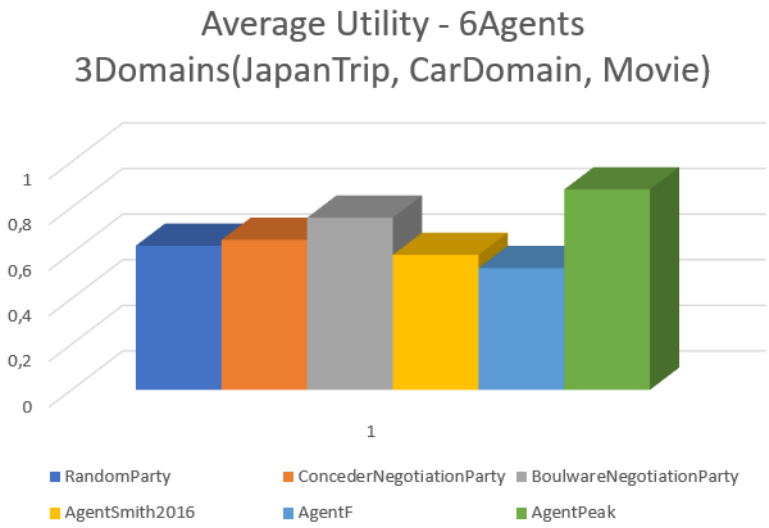
CarDomain	1o Neg - 5profiles	2o Neg - 4profiles	3o Neg - 3profiles
NumOfAgents	5	4	3
distance to pareto	0	0.12575	0
distance to Nash	0.55174	0.51532	0.084
social welfare	3.22788	2.35049	2.37089
profiles	(pr4), (pr2), (pr3), (pr5), (pr1)	(pr3), (pr2), (pr1), —, (pr4)	—, (pr1), (pr3), —, (pr2)
RandomParty	0.67697	0.4895	——
ConcederNegotiationParty	0.44417	0.44417	0.66234
BoulwareNegotiationParty	0.62493	0.63666	0.75826
AgentF	0.52342	——	——
AgentPeak	0.95839	0.78016	0.95029

Table 5: Three different Negotiations on CarDomain

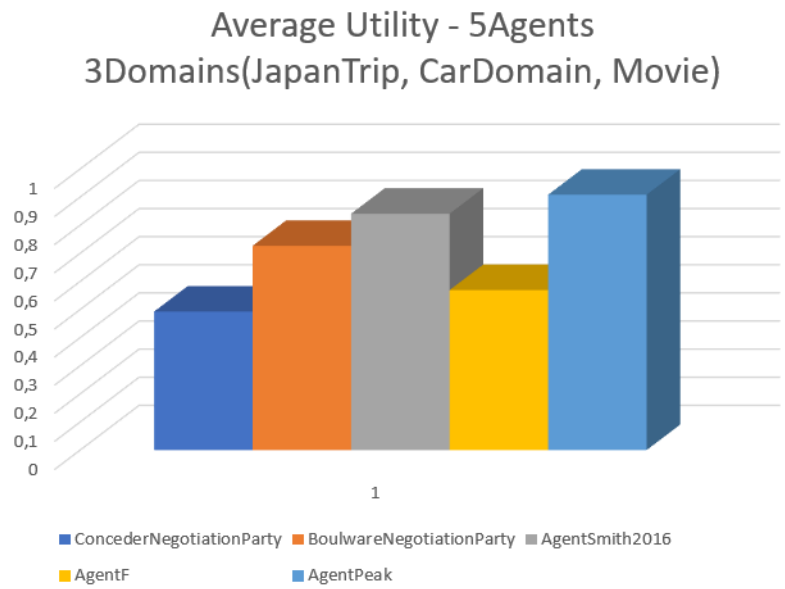
Movie	1o Neg - 3profiles	2o Neg - 3profiles	3o Neg - 3profiles
NumOfAgents	5	4	3
distance to pareto	0	0	0
distance to Nash	0.59563	0.54659	0.53658
social welfare	3.94989	3.20893	2.26591
profiles	(pr5), (pr4), (pr1), (pr3), (pr2)	(pr4), (pr2), (pr1), —, (pr3)	—, (pr3), (pr1), —, (pr2)
RandomParty	0.93398	0.515	——
ConcederNegotiationParty	0.75	0.89337	0.33333
BoulwareNegotiationParty	0.93258	0.80057	0.93258
AgentSmith2016	0.33333	——	——
AgentPeak	1	1	1

Table 6: Three different Negotiations on Movie Domain

Indicatively, tables 4,5,6 present the outcomes in negotiations in different domains. Genius provides a list of different domains. We chose the above 3 to be the main experimental domains using subsequently 3 -5 profiles. Figure 6.1 collects the above results and finds the average utility during negotiations with 5 agents.



Also, negotiations with 2 opponents took place with the average results bellow.



6.2 Level 2

After an extended number of different negotiation in several domains it was time to test AgentPeaK in tournaments. The domains that were used is Smart Grid, Laptop, Japan Trip. After studying and experimenting with agents that genius already had, we got to the conclusion that our final participants in the tournament would be :

- AgentSmith
- Caduceus
- SimpleAgent
- Boulware
- Bayesian Agent
- Conceder

It is understandable that we cannot include the log file from a tournament as the data are not that clear to study in this paper. For that reason we present tables with the number of winnings for each agent.

Smart Grid	Win
AgentPeak	4
AgentSmith	2
Caduceus	3
Grand Total	**9**

Laptop	Win
AgentPeak	2
SimpleAgent	9
NONE	16
Grand Total	**27**

Japan Trip	Win
AgentPeak	21
Boulware	48
Conceder	12
Grand Total	**81**

6.3 Level 3

Creating our own preference profiles

We felt the need to create our own preference profiles and include not only genius' agents but also agents from other teams. For that reason, 2 new preference profiles on Japan Trip domain has been created(Figure 4,5).

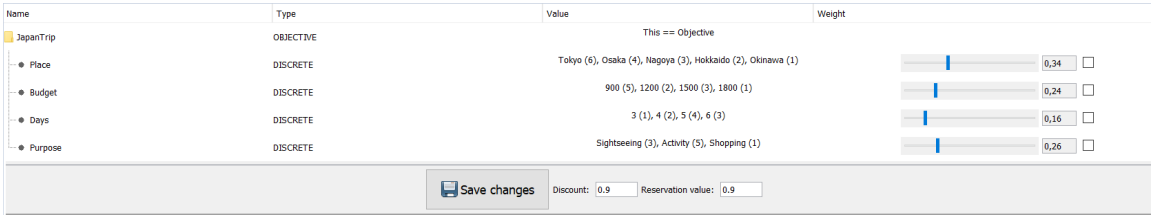


Figure 4: Preference profile 4 on Japan Trip Domain

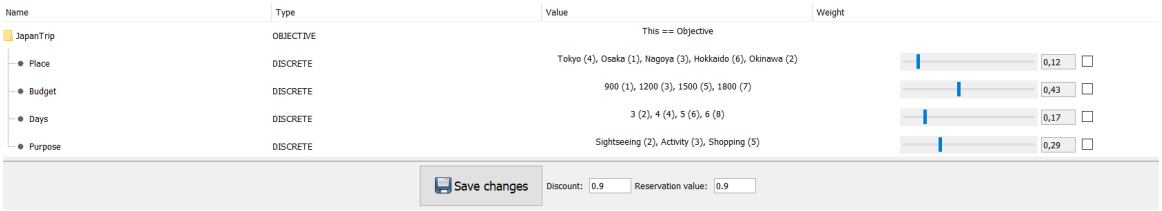


Figure 5: Preference profile 5 on Japan Trip Domain

Lastly, in collaboration with other teams we tested AgentPeaK in our own Pool. In this point, we have to point that in TUC's league (Pool A) we finished last. This position is subjective as we tested our agent many times to prove that he does not finish last every time. The negotiation bellow shows our AgentPeak's performance against MatinAgent and IQAgent. In TUC's league finished 5th in the final pool and IQAgent 8th in pool B.

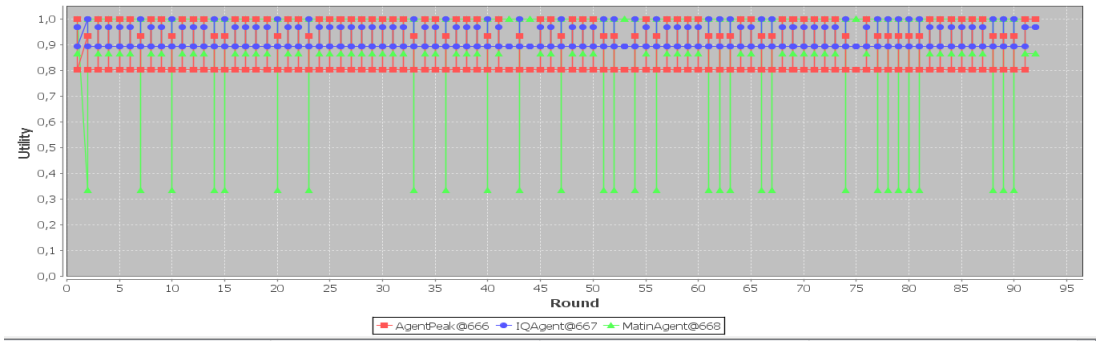


Figure 6: Movie Domain

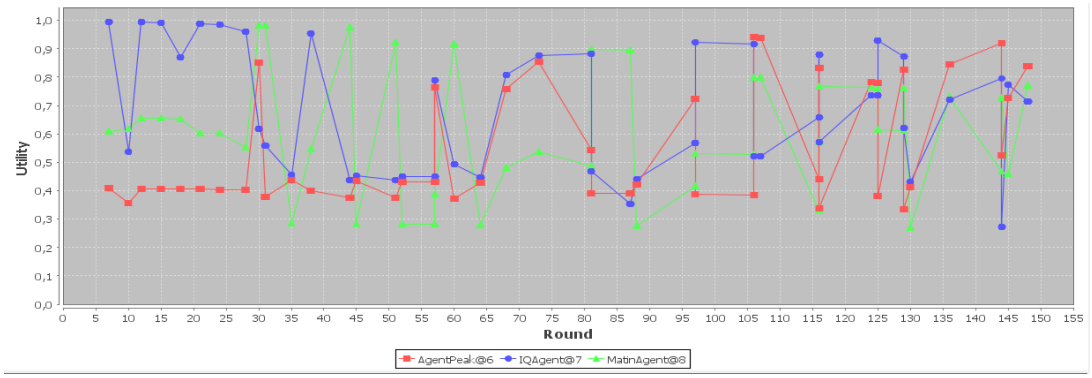


Figure 7: Japan Trip Domain

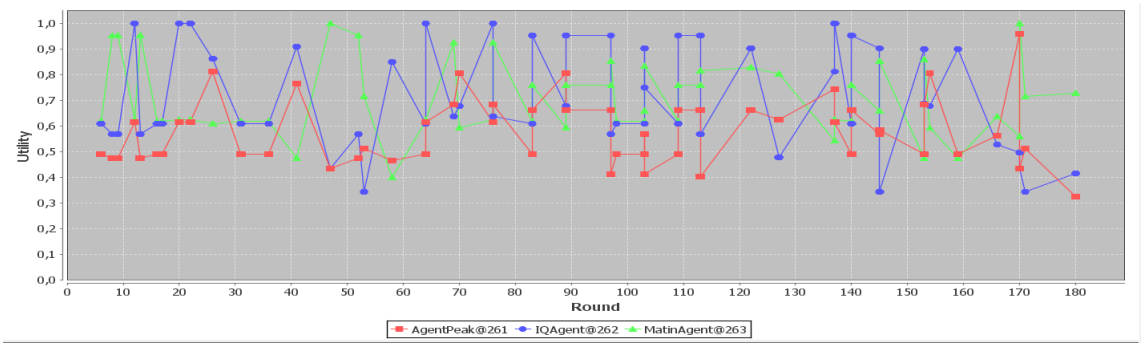


Figure 8: Car Domain

We have to note that in Car Domain we tested all agents in all possible combinations and never reached to an agreement.

After reaching to a conclusion using the existed preference profiles, we run several negotiations with our new preference profiles included. Creating those profiles has widen our research area for our agent. We tested his limits and came up with questions for further improvement.

JapanTrip	4o Neg - 5profiles	5o Neg - 4profiles	7o Neg - 5profiles	8o Neg - 4profiles
NumOfAgents	5	4	5	4
distance to pareto	0	0.1174	0.07024	0
distance to Nash	1.09172	0.28697	0.7283	0.66368
social welfare	3.61257	2.01642	2.79677	2.9536
profiles	(pr5), (pr2), (pr3), (pr4), (pr1)	(pr1), (pr2), (pr3), —, (pr5)	(pr1), (pr2), (pr3), (pr5), (pr4)	—, (pr4), (pr3), (pr2), (pr1)
RandomParty	0.40405	0.30681	0.39539	——
ConcederNegotiationParty	0.71812	0.39382	0.54537	0.71999
BoulwareNegotiationParty	0.72733	0.65026	0.71807	0.78943
AgentSmith2016	0.92067	——	0.43474	0.51545
AgentPeak	0.84241	0.66553	0.7032	0.92874

Table 7: Four different Negotiations with our own preference profiles

7 Conclusions and Future Work

In this paper, we have provided an overview of the strategy of the AgentPeaK.We have designed a new negotiation strategy based on the well-known Tit for Tat principle. AgentPeaK is a simple negotiation agent based on a number of heuristics. The features of AgentPeak include no learning as he does not learn the private information of its opponent given the difficulty of learning in selfish negotiation.He makes sufficiently minimal concessionsin order to make an agreement with its opponent, which implies that it makes concessions during negotiation.

As has been found in post-tournament analysis of the our results , AgentPeaK was the only agent in the tournament to match the behavior of the opponent (which is to be expected). This means it plays tough against hardheaded negotiators, but it also means it plays nice against strategies that concede easily. It is observed in the paper that this approach might not be as successful in negotiation as in some other games, because it does not exploit the conceding strategies enough to reach the top rankings. More research is required to find this delicate balance between cooperative and competitive behavior of a negotiating agent.

References

[1] https://www.investopedia.com/terms/t/tit-for-tat.asp

[2] Opponent Modelling in Automated Multi-Issue Negotiation Using Bayesian Learning, Koen Hindriks, Dmytro Tykhonov.

[3] Yushu A heuristic based agent for automated negotiationg competition ,$https : //www.researchgate.net/publication/225201916_Yushu_AHeuristic - Based_Agent_for_Automated_Negotiating_Competition$

[4] An Introduction to Game Theory, Martin J. Osborne

[5] An Introduction to MultiAgent Systems, Michael Wooldridge

[6] Computational Aspects of Cooperative Game Theory, Georgios Chalkiadakis, Edith Elkind, Michael Wooldridge

[7] A tit for tat negotiation strategy for real-time bilateral negotiations $https : //eprints.soton.ac.uk/373403/$