Intelligent Agents for Automated Cloud Computing Negotiation

Faisal Alsrheed, Abdennour El Rhalibi, Martin Randles, Madjid Merabti

School of Computing and Mathematical Sciences
Liverpool John Moores University
Byrom Street, L3 3AF, Liverpool, UK

Abstract— **Presently, cloud providers offer "off-the-shelf" Service Level Agreements (SLA), on a "take it or leave it" basis. This paper, alternatively, proposes customized SLAs. An automated negotiation is needed to establish customized SLAs between service providers and consumers with no previous knowledge of each other. Traditional negotiations between humans are often fraught with difficulty. Thus, in this work, the use of intelligent agents to represent cloud providers and consumers is advocated. Rubinstein's Alternating Offers Protocol offers a suitable technical solution for this challenging problem. The purpose of this paper is to apply the state-of-the-art in negotiation automated algorithms/agents within a described Cloud Computing SLA framework, and to evaluate the most appropriate negotiation approach based on many criteria.**

Keywords; Automated Negotiation; Cloud computing; Service Level Agreements (SLA) management; intelligent agents.

I. INTRODUCTION

Today, Cloud computing promises a new model of delivering computing resources with a lot of flexibility. The computing technologies can be delivered as Software as a Service (SaaS), Platform as a Service (PaaS) or Infrastructure as a Service (IaaS). Cloud computing is defined as "a large pool of easily usable and accessible virtualized resources. This pool of resources is typically exploited within a pay-per-use model: Guarantees are offered by the Infrastructure provider by means of "customized SLAs" [1]. There are, however, only a relatively small number of cloud providers in the cloud computer market and all of them offer solely "off-the-shelf" Service Level Agreements ("take it or leave it"). Introducing customized SLAs to the cloud market would offer customers and providers, added benefits. Customized SLAs can only be established by negotiation. The negotiation needs to be automated to handle the dynamic and complex environment of cloud computing. In this work rational agents will handle the automated negotiation.

A rational agent can be defined as one that is expected to be self-interested in order to reach an agreement, resulting in a high utility for the agent. The term utility refers to the quality of being useful and it is a numeric value, which measures the satisfaction of the state for an agent (i.e. the negotiation outcome). Utility functions are a way of representing an agent's preferences. The ultimate goal of each agent is to maximize its utility. When two utility-based agents try to maximize their utility in the negotiation process, there often occurs a conflict, and it may be difficult to reach agreement. Game theory may be used at this point to analyze interactions between competing agents and evaluate if cooperation would improve the outcomes for both agents. Game theory is a mathematical theory that studies interactions among self-interested agents [2]. Negotiation can be seen as a game, where two agents try to come to an agreement. Each agent is assumed to have a fixed preference over all possible deals. Both agents face the problem, of trying to maximize their utility function. This has led to a focus on automated negotiation. This interest has increased by the promise offered by intelligent agents being able to negotiate on behalf of human negotiators, or even to outperform them. As Thompson [3] pointed out, there are many problems with negotiation between humans. Firstly, negotiation, between humans, is quite slow, and is further complicated by issues of culture, ego and discrimination. Moreover, people are irrational when they make decisions, because emotions play a big role in the decision making process, evidence of this is provided in Prospect Theory [4].

In this work, a framework is proposed for achieving automated SLA negotiation between providers and consumers for cloud computing resources. In this work, the main original technical elements of the proposal are inside the negotiation stage. Also In this work, we proposed our agent Wise-H-T. The rest of this paper is organized as follows: Section 2 presents the related work; in Section 3 the Framework is introduced. Section 4 looks at the negotiation algorithms. Section 5 presents negotiation scenarios. Section 6 discusses negotiation experiments. Section 7 introduces a discussion about the experiments and the agents' evaluation, and in Section 8 the main contributions and possible future development of this work are discussed as a conclusion.

II. RELATED WORK

There are some existing negotiation frameworks and negotiation support systems already developed: OPELIX [5] is an European project that allows a customer and provider to complete fully automated bilateral negotiations. OPELIX implements all the important phases of a business operation, including product offers and discovery, a negotiation process, payment activities, and the delivery of the product to the customer. Projects Inspire [6] and Aspire [7] are associated developments. Inspire [6] helps human operators in managing bilateral negotiations by organizing offers and counter-offers. Aspire [7] improves upon Inspire by giving negotiation support using intelligent agents to make suggestions to negotiators. Agents in Aspire do not completely run the negotiation process, but offer help in taking decisions. Though, they are

fully aware of the status of the negotiation sessions. Kasbah [8] allows a customer and a provider to generate their own agents, assign them some strategic directions, and launch them at a centralized marketplace for negotiations. CAAT [9] is a framework that can be used to design multi-agent systems for bilateral negotiations. The negotiation protocol allows valid series of interactions using messages. Gheorghe et al. presented a work in [10] suggesting to using automated and intelligent negotiation solutions for reaching an SLA for an open competitive computational grid. However, SLA negotiations in grids are completely different from cloud computing negotiations. The SLA negotiations in cloud are more complex: In grid computing, negotiations will be between users that would like to use the same resource. On the other hand, in cloud computing, there are many providers that compete for a customer whilst the customer is looking for the best deal by negotiating with many providers. Also, the offer and demand in the cloud market play a major role in choosing a negotiations strategy.

The Negotiation frameworks and Negotiation Support Systems (NSS) presented above certainly make interesting advances towards automated negotiation; however they are not generic and flexible enough to easily customize negotiations for individual application domains. In addition, there are a number of issues that the above-mentioned works largely ignore which will be discussed and addressed herein. These include: The dynamic nature and heterogeneity of cloud computing. Each participant (provider and customer) has different preferences. The above works assume that price is the only or most important issue for the customer. Each participant (provider and customer) can build their agent or select one of the agents provided. Each agent behaves differently according to strategy. Also, the following features are not supported by above works; Supporting multi-issue negotiation with a large domain (hundreds of thousands possible outcomes), The possibility of negotiating with multi-providers, To be open for new agents, The possibility of re-negotiating and Monitoring the SLA after the negotiation has completed. In this paper we propose a solution which satisfies these requirements more fully.

III. FRAMEWORK

The framework is made up of 5 stages; the output of each stage is an essential input for the next stage:

Stage 1: Gathering

In this stage all the inputs for the framework will be gathered together. The inputs will be a customer's request and a provider's offer, the policy of the negotiation's strategy, the negotiation's preferences, the price policy, the monitoring rules and policies, the real-time monitoring results and the monitoring alerts. All the inputs will be saved in an accessible database.

Stage 2: Filtering

In this stage, the customer requests that have been sent in the gathering stage will be used to filter all the providers in order to recommend the best matched candidates. The customer's request can include the detailed criteria of the demanded cloud computing service. The output of this stage will be the candidate providers, with whom the customer will be negotiating separately.

Stage 3: Negotiation

In this stage the customer will negotiate separately with each candidate provider. Then, the outcomes of each session of the negotiation will be compared. The output of this stage is that the best outcome from the customer's perspective will be picked up, which will be the agreed value for each parameter.

Stage 4: SLA Agreement.

In this stage the provider and the customer will be informed about the Agreement, which will be specified in measurable terms. The output of the SLA Agreement stage will be a list of metrics that can be monitored in the following stage.

Stage 5: Monitoring.

This stage will use a monitoring client to gather the real-time data to ensure the SLA is enforced. Based on the monitoring rules and policies, actions will be taken to correct the cloud computing provision if the provided fails to respect the SLA.

In this paper we present how stage 3 (Negotiation) works, and the benefit of our approach; the next section will introduce the negotiation protocol and the algorithms.

IV. NEGOTIATION PROTOCOL AND ALGORITHMS

The negotiation protocol is needed to determine the overall order of actions during a negotiation. In this work a Protocol known as Rubinstein's Alternating Offers Protocol, or Rubinstein bargaining model, will be used as formalized in [11].The strong motivation behind chosen this protocol are; First, the simplicity of this protocol. Second, it is been widely studied and used in the literature [12]. Third, In Rubinstein's Alternating Offers Protocol, there is no delay in the transaction. This protocol is a one-to-one protocol (Agent-to-Agent): Agents negotiate over a series of rounds. At the first round, an agent makes an offer then the other agent either accepts or rejects it. If the offer is accepted, the deal is implemented (Agreement). If the offer is not accepted, then the negotiation keeps going until one agent accepts the other offer, or the negotiation times out without agreement.

Now, after determining the protocol of negotiation, it is necessary to discuss the negotiators. The provider and customer will negotiate over a set of issues, and every issue has an associated range of alternatives or values. A negotiation outcome consists of a mapping of every issue to a value, and the set of all possible outcomes is called the negotiation domain. Both parties have privately-known preferences described by their utility functions. Both utility functions, map every possible outcome $\omega \in \Omega$ to a real-valued number in the range $[0, 1]$, where ω is the outcome and Ω is the domain. The overall utility consists of a weighted sum of the utility for each individual issue.

$$U(_{v1,\ldots\ldots vn}) = \sum_{i=1}^{N} w_i \frac{eval(v_i)}{Max(eval(v_i))} \qquad (1)$$

A bid is a set of chosen values $^{v1,\ldots\ldots vn}$ for each of the n issues. Each of these values has been assigned an evaluation value

$eval(v_i)$ in the utility space. The utility is the weighted sum of the normalized evaluation values. While the domain (i.e. the set of outcomes) is common knowledge, the utility function of each player is private information. This means that the players do not have access to the utility function of the opponent. However, the player can attempt to learn it during the negotiation. Agents represent the negotiators (provider and customer). Each agent has a different strategy of negotiating. The ideal agent needs to be able to learn the opponent behaviour from its moves and predict the opponent's next moves. In this way the agent can then decide when to make a cooperative offer or a selfish offer; when to accept the opponent's offer; when to end the negotiation without agreement; keep track of the remaining time in the negotiation session; or estimate the Nash-Equilibrium point [13].
Each agent follows a completely different approach to perform each of these actions. In this work, we will analyze and compare how seven different agents will follow different kinds of strategies to perform each of above actions, and their outcomes in negotiation agreement. The agents are informally termed HardHeaded, Tit-for-Tat, Hardliner, IAMhaggler, Gahboninho , AgentFSEGA and WiseH-T.

A. HardHeaded

The HardHeaded agent [14] starts each negotiation session by computing the utility for all possible bids. Then it stores them in a search tree (i.e. binary tree data structure) for fast recovery. This agent uses a learning module. The target of the learning module is to learn the utility value and the weights of the opposing agent [14]. To study the opposing agent, this agent makes two assumptions about the opponent; it first assumes that the opponent restricts the bids with a limited utility range. The second assumption is that the opponent does not prefer to be offered the same bid over and over again. HardHeaded's learning function is "a greedy reinforcement learning function" [14]. This learning function keeps updating the issue weights and value utilities of the preference profile immediately after each bid. At the same time this learning function will always try to identify the most valuable bid, and the least valuable bid for the opponent, so it can offer a bid which is most likely to be accepted when the negotiating session is about to end [14].

B. Tit-for-Tat

This agent's strategy is based on the principle of Tit-for-Tat (tft) [15]. In Tit-for-Tat strategy, the first move is always a cooperative move and then keeps mirroring whatever the other player did in the previous round [15]. This agent plays a tit-for-tat strategy with respect to its own utility. In the beginning, this agent will cooperate, and then respond to the opponent's previous action, while aiming for the Nash point of the negotiation scenario [16]. After every opponent's move, this agent will update its Bayesian opponent model to make sure it reacts with a beneficial move to a concession by the opponent [16]. This opponent model will help the agent to measure the opponent's concession in terms of the agent's own utility function; mirror this bid as described in the tft strategy above, giving up the same amount as is offered by the opponent or make the offer as attractive as possible for the opponent using the Bayesian opponent model [16]. In addition, the opponent model is used by this agent to make an estimate of the location of the Nash point of the negotiation scenario, and then aims for this outcome [16].

C. Hardliner

Hardliner [16] is a very selfish and stubborn agent that keeps repeating the same offer, which is only good for itself, expecting the opponent to give up and accept the offer at the end. Its approach to negotiation is known as "take-it-or-leave-it" strategy, which is similar to current "off-the-shelf" SLAs. This strategy makes a bid of maximum utility for itself and never concedes. This is the most competitive strategy that can be used. This agent is deterministic. It will give the opponent the full negotiation time to make concessions and accept its offer. This agent is analyzed, as it represents the present cloud providers approach to negotiation e.g. Amazon ES2, Google Compute Engine and Microsoft Azure, since they only propose take-it-or-leave-it cloud packages for the market.

D. IAMhaggler

This agent involves three parts; the first part predicts the concession of the opponent by using a Gaussian process regression technique [17]. The second part sets the concession rate in such way that it optimizes the expected utility given that prediction. The third part generates a multi-issue offer according to the concession rate [18]. This agent first has to predict how the opponent will concede during the negotiation, only by using the information that can be observed (the opponent's offers and the utility of these offers according to the agent's own utility function). This agent uses the strategy of the opponent's future concession prediction, to set its concession rate by optimizing the expected utility given to that prediction. After selecting a target utility, this strategy needs to make an offer that has a utility close to that target [17].

E. Gahboninho

This agent starts the negotiation with a no-compromising strategy in an attempt to put pressure on the opponent. During the negotiation, the agent observes the opponent's behavior. This agent strategy is that when the opponent attempts to propose a realistic and compromising offer, there is no need to compromise, as it is not going to make the opponent bid a better offer. At the same time, this agent may also model the opponent's preferences. In contrast, if the opponent will not compromise whatsoever, this agent will avoid deadlock and give up its utility in accordance with the pressure. As the pressure increases the agent may give up its utility quicker, but would never go lower than the utility of the best offer that is suggested by the opponent. Once the agent has gathered a sufficient amount of information about the opponent's preferences, it will attempt to filter the domain (i.e. all possible outcomes) of most of the inefficient outcomes, in order to make sure that the critical bids, which are usually the last rounds, are effective [19]. This agent may seem to be a greedy agent but it can avoid break-offs when facing uncompromising opponents. However, this agent's success also relies on the cooperation of opponent [19].

F. AgentFSEGA

This agent is a Bayesian learning agent [20]; the Bayesian learning negotiation strategy is adapted to cope with time constraints. Also, the Bayesian learning during the negotiation, will try to infer the utility function of the other player by executing two actions; Firstly, analyzing the incoming opponent's proposal and updating the opponent's profile. Secondly, selecting and proposing the next bid. Bayesian learning [20] maps to a probability each possible hypothesis about the opponent profile. The hypothesis can be the rankings of the preferences for the issues of the opponent. It is time-constrained agents, which means that based on the remaining time of the negotiation session, this agent will act and be more flexible.

G. Wise H-T (W H-T)

The Bidding Opponent Acceptance (BOA) framework [21] is used to form our agent Wise H-T. The BOA negotiation agent architecture allows researchers to re-use existing components from other BOA agents. The BOA agent can be made of four different modules: one module that decides whether the opponent's bid is acceptable (*acceptance strategy*); one that decides which set of bids could be proposed next (*bidding strategy*); one that tries to guess the opponent's preferences (*opponent model*), and finally a component which specifies how the opponent model is used to select a bid for the opponent (*opponent model strategy*). The overall negotiation strategy is a result of the interaction between these components [21]. The advantages of separating the negotiation strategy into four components are: first, it allows study of the performance of individual components; second, it allows a systematic exploration of the space of possible negotiation strategies; third, the re-use of existing components simplifies the creation of new negotiation strategies [21]. Wise H-T is made from the bidding strategy from HardHeaded and the acceptance strategy from Tit-for-Tat. The opponent model is the opposite model. The input of the opponent model is a set of possible bids and negotiation trace. The output is the estimated opponent utility of a set of bids

V. SCENARIO REPRESENTATION

Each of the following agents; AgentFSEGA, Gahboninho, HardHeaded, Tit-for-Tat Agent, IAMhaggler, and Wise H-T will negotiate against a baseline that is composed of the following agents (Hardliner, Gahboninho, HardHeaded, Tit-for-Tat Agent and IAMhaggler).

There are two criteria that will be taken into account, in order to evaluate each agent. First, is the performance and the second one is the fairness. Performance is the sum of the all utilities the agent has while negotiating with the other agents. The agent with the highest number means that the agent has the best performance:

$$Performance = 0 \leq \frac{\sum_{k=1}^{n_a} u_a}{n_a} \leq 1 \qquad (2)$$

Where n_a is the number of the agents and u_a is the utility of the agent. The Performance is measured between 0 and 1.

Fairness is formally defined as:

$$Fairness = 0 \leq \frac{\sum_{k=1}^{n_a} u_a + u_{op}}{n_a * 2} \leq 1 \qquad (3)$$

Where n_a is the number of the agents and u_a is the utility of the agent and u_{op} is the utility of the opponent. The Fairness is measured between 0 and 1. The agent with the highest number means that the agent has the best fairness. The Fairness is measured between 0 and 1.

TABLE I. TABLE 1 STORAGE AS A SERVICE CRITERIA

Issue	Value	Customer Evaluation	Weight	provider Evaluation	Weight
Availability Zone (location)	US-East US-West Europe Asia (Tokyo)	1.00 0.75 0.50 0.25	0.17	1.00 0.50 0.25 0.75	0.19
Term (months)	[1-6] [7-12] [12-24] >24	0.25 1.00 0.75 0.50	0.22	1.00 0.50 0.25 0.75	0.11
Backup	Every 12 hours 1 days 1 week 1 month	0.75 0.25 0.50 1.00	0	0.75 1.00 0.50 0.25	0.18
Data In	Light Medium Heavy Live stream	0.25 0.75 1.00 0.50	0.07	1.00 0.25 0.75 0.50	0.22
Data Out	Light Medium Heavy Live stream	1.00 0.75 0.50 0.25	0.54	1.00 0.50 0.75 0.25	0.30

In this scenario we assume that a customer is looking for a provider who is capable of providing Storage As a Service, with the criteria as shown in table 1. After finding providers who are willing to provide offers matching the above criteria, the customer will negotiate with them. However, each side (provider and customer) have different preferences. For example, the provider would like a customer requiring long-term facilities in one location with less Utilization. Afterwards they need to negotiate. The negotiation will be closed in the sense that there is uncertainty about the opponent's preferences. In this scenario, a customer and a provider, negotiate over the specifications of Storage as a Service. There are 5 issues: Availability Zone, Term, Back up, Data In, Data out. In this scenario, each issue has 4 options, so there are 1027 possible outcomes for this negotiation.

It is essential to set up a deadline for the negotiation, as without a deadline the negotiation might go on forever. The effect of switching between time-based deadline and round-based deadline will be investigated. Also, the effect of the increase and the decrease of the deadline to the negotiation outcome will be investigated. To investigate the effect of a deadline to the negotiation outcomes, the same scenario was ran with the same agents for three different times, and then the outcomes were compared: first a very short time of 10 seconds; the second for 100 seconds and the last one for 1000

seconds. The same has been done with the round-based protocol: first a very short time of 10 rounds; the second for 100 rounds, the third for 1000 rounds and the last one for 10000 rounds.

VI. NEGOTIATION EXPERIMENTS OUTCOME:

In this section, the results will be discussed. The following graph (Fig. 1) shows the results of performance (p) and fairness (f) for agentfsega (FSEGA) , gahboninho (GAB) , hardheaded (HH), imhaggler (IMH), tit for tat (TFT), wise h-t (W H-T) when they will negotiate against each of the following agents; Hardliner , Gahboninho, HardHeaded, Tit for Tat Agent and IAMhaggler.

A. Time-Based Deadline:

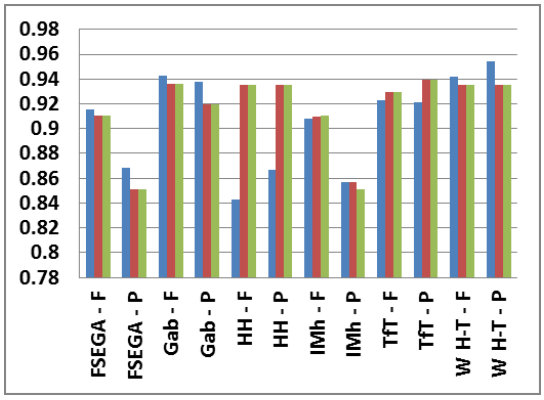

Fig. 1. Agents' Performance & Fairness for 10, 100 & 1000 Seconds

The blue bars show the 10 seconds negotiations outcomes and red bars show the 100 seconds negotiations outcomes and green bars show the 1000 seconds negotiations outcomes. When the deadline is 10 seconds, Wise H-T agent was first for performance. Gahboninho was first for the fairness. Wise H-T agent is in second place for fairness. When the deadline is changed to 100 and 1000 seconds, Tit for Tat Agent was in first place for performance. Gahboninho was first for the fairness.

B. Round-Based Deadline

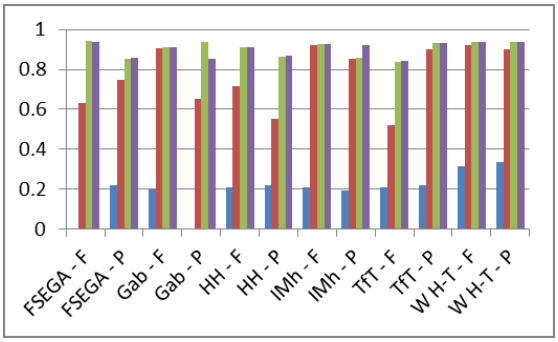

Fig.2. Agents' Performance & Fairness for 10, 100, 1000 & 10 000 Rounds

The blue bars show the 10 rounds negotiations outcomes and red bars show the 100 rounds negotiations outcomes and green bars show the 1000 rounds negotiations outcomes. The purple bars show the 10000 rounds negotiations outcomes

Figure 2 shows Wise H-T's ability to end a negotiation with an agreement in Pareto efficient frontier [22] around Nash Point [23]; even though negotiation is with Hardheaded with the short deadline of 10 seconds.

Figure 3 illustrates an example of negotiation between Wise-H-T agent (A) and HardHeaded (B), highlighting among other results all the possible bids, agent A bids, agent B bids and the agreement.

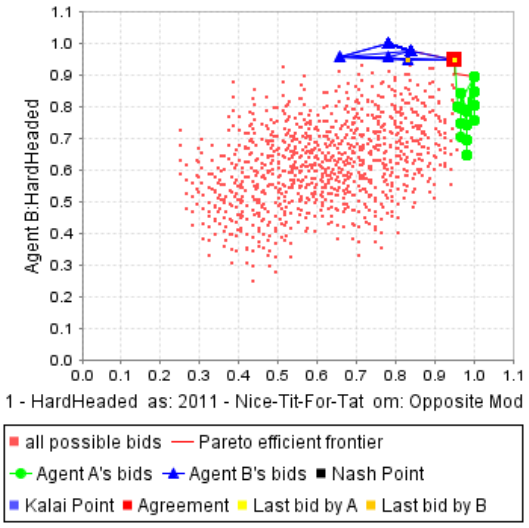

Fig.3. Wise H-T vs Hardheaded

VII. DISCUSSION

A. Analysis

a) Time-Based Deadline:

The results show that Wise H-T is capable of doing well even if the deadline is short. HardHeaded performed as well as the Wise H-T for a deadline of either 100 or 1000 seconds. Hardheaded was the most effective agent when the time changed from 10 seconds to 100 or 1000 seconds. There is only a slight difference in the performance of a IMhaggler agent when the deadline was 100 seconds and 1000 seconds, where the rest of the agents have achieved exactly the same performance and fairness when the deadline was 100 or 1000 seconds.

b) Round-Based Deadline:

Overall, all the agents did better when deadlines were increased. When the rounds were increased up to 100, Tit-for-Tat and Wise H-T had the best performance and HardHeaded had the worst. Gahboninho had the best performance when the deadline was 1000 rounds; second place was for Wise H-T. For 10000 rounds, HardHeaded, Tit-for-Tat, AgentFSEGA and Wise H-T got exactly the same results as for when the deadline was 1000 rounds, but Wise H-T got the best results.

For fairness, AgentFSEGA's result was 0 and the best result was for the Wise H-T agent. When the deadline was 100 rounds, IAMhaggler2011 and Wise H-T did the best and shared the first place, second placed agent was Gahboninho and the final one was Tit-for-Tat. All agents did exactly the same as they did when the deadline was 1000 or 10000. AgentFSEGA did slightly better at a 10000 rounds than the

Wise H-T agent. After these investigations, the following recommendations can be made.

B. Recommendations

a) Round-Based Deadline

There is a high cost in setting the deadline to a low number of rounds such as 10 rounds or less, as most of the negotiation sessions will end up with no agreement. The agents will do slightly better if the deadline be increased to 100 rounds. However, if the goal is to increase the overall fairness and performance then it is recommended to increase the deadline to 1000 rounds but not more than 1000, as it will not make any difference to the outcomes of the negotiations.

b) Time-Based Deadline:

There is a correlation between the deadline round-based protocol and the time-based protocol; it was found that when the deadline was set up to 10 seconds the number of rounds that each negotiation session took was between 2 and 2000. When the deadline was set up to 100 seconds, each negotiation session took between 4000 and 14000 rounds. However, when the deadline was increased to 1000 seconds, the number of rounds increased to be between 19000 and 23000. Consequently, investigation shows that by increasing the deadline to more than 10000 rounds or 100 seconds will not improve the negotiation outcomes. So, it is not recommended to increase the deadline to 100 seconds or more, as this will not make any difference to the outcomes of the negotiations.

VIII. CONCLUSION

This work focuses on how Rubinstein's Alternating Offers Protocol can be used for automated negotiation for Cloud Computing. A framework is proposed for achieving automated negotiation between providers and consumers in cloud environments. The main novelty of the work is that the framework is made especially for cloud computing by using state-of-the-art automated negotiation algorithms/agents. Also, at the same time it is flexible and open to new algorithms/agents that might be developed in the future. The related work has been classified into the following categories: Negotiation Evolution, Negotiation frameworks and Negotiation Support Systems (NSS), Negotiation algorithms are discussed and the Negotiation experiments outcome is analyzed. The effect of switching between time-based deadline and round-based deadline was investigated. Also, the effect of the increase and the decrease of the deadline to the negotiation outcome investigated.

ACKNOWLEDGMENT

We are grateful to the Interactive Intelligence research Group at Delft University of Technology for making Genius [24] available which we used for the evaluations.

REFERENCES

[1] Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner(2008)"A break in the clouds: towards a cloud definition". SIGCOMM Comput. Commun. Rev. 39, 1.

[2] Binmore K (1992) Fun & Games: a text on Game Theory, D.C. Heath and Company, Lexington, MA

[3] Thompson, L (2012) "The Mind and Heart of the Negotiator," Edition 5, Pearson Education.

[4] Kahneman, D and Tversky, A (1979) "Prospect Theory: An Analysis of Decision under Risk" Econometrica Vol. 47, No. 2. pp. 263-292.

[5] Hauswirth, M, Jazayeri, M, Miklos, Z, Podnar, I, Di Nitto, E &Wombacher, A (2001) An Architecture for Information Commerce Systems. In: Proceedings of the Sixth International Conference on Telecommunications (ConTEL).

[6] Kersten, G.E & Noronha, S.J (1999)-based Negotiation Support: Design, Implementation and Use. Journal of Decision Support Systems 25(2).

[7] Kersten, G.E & Lo, G (2003) An Integrated Negotiation Support System and Software Agents for E-Business Negotiation. International Journal of Internet and Enterprise Management 1(3)

[8] Chavez, A, Dreilinger, D, Guttman, R, &Maes, P (1997) A Real-Life Experiment in Creating an Agent Marketplace, the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97)

[9] Ncho, A &Aimeur, E (2004) Building a Multi-Agent System for Automatic Negotiation in Web Service Applications. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-agent Systems.

[10] Gheorghe CS,Liviu D Şerban, Cristian M L (2012), A time- constrained SLA negotiation strategy in competitive computational grids, Future Generation Computer Systems, Volume 28, Issue 8

[11] Rubinstein, A (1982) Perfect Equilibrium in a Bargaining Model. Econometrica: Journal of theEconometric Society, 50:97–109

[12] Filzmoser, M (2010) Automated vs. Human Negotiation, International Journal of Artificial Intelligence 4(10), pp. 64-77.

[13] Nash, Jr. J.F (1950) The Bargaining Problem. Journal of the Econometric Society 18(2)

[14] Thijs van Krimpen, Daphne Looije, Siamak Hajizadeh (2013),HardHeaded, Complex Automated Negotiations: Theories, Models, and Software Competitions Studies in Computational Intelligence Volume 435 , pp 223-227

[15] Axelrod, R(1984)The Evolution of Cooperation. Basic Books

[16] Baarslag, Tim, Fujita, Katsuhide, Gerding, Enrico H., Hindriks, Koen, Ito, Takayuki, Jennings, Nicholas R., Jonker, Catholijn, Kraus, Sarit, Lin, Raz, Robu, Valentin and Williams, Colin R. (2013) Evaluating Practical Negotiating Agents: Results and Analysis of the 2011 International Competition. Artificial Intelligent.

[17] Rasmussen, C.E., Williams, C.K.I (2006) Gaussian Processes for Machine Learning. The MIT Press

[18] Colin R. Williams, Valentin Robu, Enrico H. Gerding, Nicholas R. Jennings (2013) IAMhaggler2011: A Gaussian Process Regression Based Negotiation Agent, Complex Automated Negotiations: Theories, Models, and Software Competitions Studies in Computational Intelligence Volume 435, pp 209-212

[19] Mai Ben A, Nadav Sofy, Avshalom Elimelech (2013) Gahboninho: Strategy for Balancing Pressure and Compromise in Automated Negotiation.Complex Automated Negotiations: Theories, Models, and Software Competitions, Studies in Computational Intelligence Volume 435, pp 205-208.

[20] Gheorghe Cosmin Silaghi, Liviu Dan Ş, Cristian Marius L (2012), A time-constrained SLA negotiation strategy in competitive computational grids, Future Generation Computer Systems, 28 (8), P1303-1315.

[21] Baarslag, T., Hindriks, K., Hendrikx, M., Dirkzwager, A., Jonker, C.: Decoupling negotiating agents to explore the space of negotiation strategies. In: Proceedings of the 5th International Workshop on Agent-based Complex Automated Negotiations, ACAN 2012 (2012).

[22] Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. Structural and multidisciplinary optimization, 26(6), 369-395.

[23] Jr. J.F (1950) The Bargaining Problem. Journal of the Econometric Society 18(2).

[24] Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K. and Jonker, C. M. (2012), Genius: An Integrated Environment For Supporting The Design Of Generic Automated Negotiators. Computational Intelligence.