Asian Journal of Information Technology 15 (18): 3587-3597, 2016
ISSN: 1682-3915

Efficient Negotiation of SLA Parameters in Cloud Computing Environment

S. Usha Kiruthika and S. Thamarai Selvi
Department of Computer Technology, Anna University, MIT Campus, Chennai, Chromepet, India

Abstract: Cloud computing has enabled a pay-per-use model where anything is available anywhere as a service. While service provisioning requires no manual intervention the automated establishment of Service Level Agreement (SLA) which determines the Quality-of-Service (QoS) is still a challenge. This is mainly because of the complex decision making involved in automated negotiation of QoS parameters. Automated negotiation involves two or more self-interested agents negotiating with each other to agree upon a set of QoS parameters. The success of an automated negotiation system lies in the negotiation strategy that an agent adopts to generate counter-offers. In this research, we propose the architecture of an SLA management system. We analyze the properties of linear utility functions and propose an algorithm for generation of counter-offers that are more acceptable to the opponent.

Key words: Service level agreement, automated negotiation, concession, trade-off, opponent

INTRODUCTION

With growing number of small scale businesses and ever rising costs in the development of infrastructure, cloud computing has come as a boon for many consumers getting services through the Internet. The services may be in the form of Software (SaaS), Platform (PaaS), Infrastructure (IaaS) or anything (XaaS). But one common challenge for all types of providers is the establishment of a Service Level Agreement (SLA). Most providers today provide a verbal SLA which includes ambiguities and is flexible for the provider in many ways. The consumer has no role in the creation of these SLAs and has to accept them mandatorily in order to use the services of the provider. To solve this problem, specifications for formal SLAs such as WSLA (Keller and Ludwig, 2003), WS-agreement SLAng have been proposed in XML format. These specifications let the provider and the consumer to negotiate before agreeing on an SLA. With increasing number of consumers moving on to the cloud, it is becoming very time-consuming to conduct negotiation manually. Manual negotiation requires the negotiating parties to be present at opposite ends at the same time for faster results. This is not always possible as the parties may be separated geographically across time zones. Moreover, the human negotiators must be good decision makers. Automated negotiation solves these problems since the actual negotiation is done by software with only the requirements specified by the user. It is much faster and negotiation could be done with finer granularity when required.

Though negotiation is a universal problem applicable to broad areas, this study discusses the problem in the context of negotiation of SLAs between a consumer and a provider in cloud environment. The problem of counter-offer generation in a bilateral SLA negotiation with multiple parameters has been focused in this research work. We assume linear utilities and ranking of opponent's parameters is known to the agents.In order to generate a counter offer a negotiating agent may decide to give a concession or do a trade-off. Trade-off is the preferred alternative as it maximizes the joint outcome of the negotiating agents (Zheng *et al*., 2012). Concession is done only when it is not possible to do a trade-off. We propose a trade-off algorithm that aims to make the offer generated by the trade-off more acceptable to the opponent. A good trade-off is possible only when the preferences of the negotiators are opposite. Preference denotes the amount of importance a participant of negotiation attaches to each parameter being negotiated. When negotiators have opposite preferences, one negotiator attaches more importance to a parameter that the other negotiator attaches less importance. To summarize, the main contributions of this work are:

- Development of an SLA management framework
- Analysis of properties of linear utility functions related to single-issue negotiation and their applicability to multi-issue negotiation
- Development of trade-off algorithm to generate offers more acceptable to the opponent

Corresponding Author: S. Usha Kiruthika, Department of Computer Technology, Anna University, MIT Campus, Chennai, Chromepet, India

Literature review: SLA negotiation framework has been discussed by several works (Chhetri *et al*., 2006; Ludwig *et al*., 2005; Alhamad *et al*., 2010; Al-Aaidroos *et al*., 2011; Wu *et al*., 2013). In particular, Yan *et al*. (2007) propose an SLA negotiation framework and discuss the process of negotiation including the protocols involved. A negotiation architecture is proposed by Di Nitto *et al*. (2007). SLA negotiation is done using a search based approach. A fitness function based on the utilities and their distances is presented. For optimization, experiments have been done using genetic algorithms, hill-climbing and simulated annealing. It has been found that simulated annealing gives the best performance. The approach assumes varying preferences over the course of negotiation. In these works the frameworks and negotiation strategies or protocols are focused.

Automated negotiation research community has always aimed for reaching an optimal agreement with known or unknown opponent information using various approaches. Faratin *et al*. (2002) propose a trade-off algorithm that uses hill-climbing technique to search for offers similar to the opponent's offer. The search starts at the opponent's offered contract and proceeds by generating a set of contracts that lie closer to the iso-curve (representing the agent's aspiration level). Another research (Cheng *et al*., 2006) proposing search based approach uses fuzzy inference for providing trade-offs. A suitable counter-offer for an offer is selected by searching a multi-dimensional space formed by negotiable issues. The desired values and weights are revealed to the opponents but the utility functions are kept private. The most similar offers are selected and counter-offers are generated using heuristics represented by a set of fuzzy rules. The similarity between two offers is found using the nearest-neighbor matching algorithm. Li *et al*. (2013) describe a negotiation framework that changes based on the environment. The behavior of a negotiating agent changes depending on the competition and demand for a particular service. The BATNA (Best Alternative to a Negotiated Agreement) and strategies for negotiation are dynamic and the agents are proactive and responsive by searching for options which are outside of the negotiation and which may improve their outcomes. Ros and Sierra (2006) propose a meta-strategy that combines concession and trade-off. Trade-off is done by the agents until deadlock is detected. A deadlock occurs when the proposal of an offer of an agent decreases from the previous offer. When a deadlock is detected, concession is done. The trade-off algorithm is based on trade-off algorithm by Faratin *et al*. (2002). Zheng *et al*. (2014) proposes a negotiation approach mixing the concession and trade-off approaches to yield the benefits of both. The negotiation is for an internet of things environment and is based on 'game of chicken'. Vetschera *et al*. (2014) present a concession based negotiation approach in which the concession in each round is variable and is determined by the user allowing more user control. The bargaining is done in utility space and is mapped to the offers in issue space using several variants of optimization models.

MATERIALS AND METHODS

Architecture of the SLA management system: We first introduce the architecture of an SLA management system (Fig. 1) which is the environment in which automated negotiation is done. The cloud architecture consists of three stake holders. They are providers, consumers and brokers. A consumer approaches a broker to identify and avail services from providers. The broker matches the requirements of the consumer with a provider with a matchmaking service. Requirements are a combination of QoS parameters, service parameters and cost. The broker then gets the initial proposal from the matched set of providers. The initial proposal of a provider contains the initial values of the negotiable parameters promised by the provider and is available in the SLA template pool.

The broker selects the SLA that is most suited for the consumer according to the requirements given by the consumer. The broker may choose more than one SLA from one or more providers. With the selected providers, the broker negotiates according to the requirements of the consumer using the negotiation service. The negotiator service on provider's side and broker's side employ negotiating agents (Since agents are the actual participants in a negotiation, the term agents and participants are used interchangeably in this study). Once an agreement is reached with one of the providers, an SLA is established. The SLA is instantiated by the SLA instantiator service of the broker and it is shared with the provider and consumer. The monitoring system of the broker monitors the service provisioning of the provider for any violation of the SLA. The provider will be penalized in case of a violation. This is the overview of the SLA management system. The focus of this paper is on the negotiator service that is responsible for automated negotiation between the broker and the provider.

Negotiator service: Each negotiator agent is an autonomous software program that negotiates on behalf of either the consumer or the provider. An instance of the negotiation agent is created by the negotiation broker

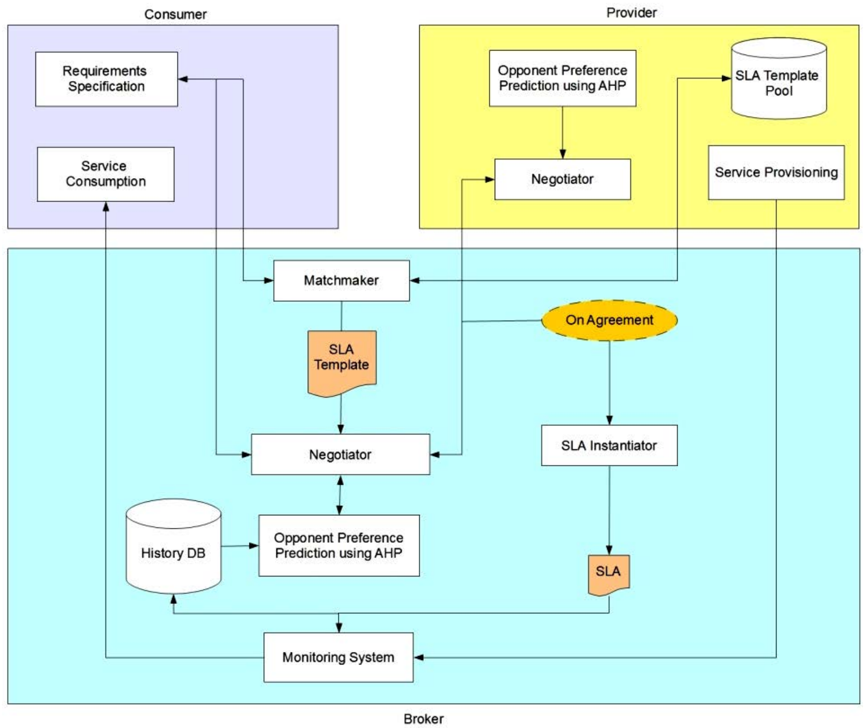

Fig. 1: Architecture of SLA management framework

once the list of providers with whom negotiation has to take place is finalized. Each provider employs a negotiation agent on its behalf. The agents on both the broker's side and the provider's side are capable of negotiating with multiple opponents. This is to allow negotiation of a provider agent with multiple broker agents and negotiation of a broker agent with multiple provider agents. The functions of a negotiation agent on broker side and provider side are exactly the same.

A negotiation agent receives proposals from opponents and accepts, rejects or generates counter proposals by giving a concession or trade-off. It also chooses the best offer from the negotiations with various opponents. All the counter-proposals received by a negotiator agent from one opponent are hidden from other opponents. Therefore, the negotiation system of each agent is multi-threaded with individual negotiations independent of one another. A negotiator agent ranks their opponents based on the final outcome of each negotiation. This system is particularly beneficial to a consumer who looks for the best service available among a set of providers.

Linear utility: The formula used for calculation of utility of a parameter value varies depending on whether a participant aspires for a higher value of a parameter or a lower value of a parameter. Let p_{min} be the minimum and p_{max} the maximum fixed by a participant. Let p ($p_{min} \leq p \leq p_{max}$) be the parameter value for which utility is being calculated.

When the participant aspires for higher value of the parameter (p_{min} is reserved value and p_{max} is preferred value), utility of p is:

$$u = \frac{p - p_{min}}{p_{max} - p_{min}} \tag{1}$$

When the participant aspires for lower value of the parameter (p_{min} is preferred value and p_{max} is reserved value), utility of p is:

$$u = \frac{p_{max} - p}{p_{max} - p_{min}} \tag{2}$$

Utility is modeled as a linear increasing or decreasing function. When the participant aspires for higher value, utility is an increasing function (Eq. 1) that is higher the value higher the utility. Otherwise, utility is a decreasing function (Eq. 2). The total utility (Eq. 3) is calculated by assigning weights to each parameter. A higher weight denotes the parameter is more important.

$$u_{tot} = w_1u_1 + w_2u_2 + w_3u_3 + ... + w_nu_n \quad (3)$$

Reaching pareto-optimality: Though the negotiating agents individually aim for higher utilities, a good negotiation ends in an agreement that yields a high joint outcome. That is on agreement, the sum of utilities of the negotiators needs to be as high as possible and the difference of utilities needs to be as low as possible. An ideal point would be (1.0, 1.0) in the utility space which yields the maximum sum of 2 and zero difference. But this is utopian point and is not feasible in most negotiations. Feasible points are marked by a feasibility region which is a line for single parameter negotiations (Fig. 2a) and an area for multiple parameter negotiations (Fig. 2b and c).

The point that lies in the upper-rightmost corner in the feasibility region is the point that results in best joint outcome. In single parameter negotiations, the feasibility line is the Pareto-frontier. In multi-parameter negotiations, the North-West edge of the feasibility region marks the Pareto-frontier. All the points that lie in the Pareto-frontier are Pareto-optimal points. A Pareto-optimal point is a point in which a better utility cannot be achieved for an agent without lowering the utility of the other agent. The Pareto-frontier joins (0, 1) and (1, 0) points when the maximum and minimum values of each parameter are the same for both negotiators. Otherwise it joins (0, m) and (n, 0) where m and n depend on maximum and minimum values fixed by the negotiators.

We now derive the optimal point of agreement in terms of minimum and maximum values of participants for a single parameter negotiation. The optimal point of agreement is the intersection of the utility functions. Let a_B, b_B be the minimum and maximum values for a parameter p of agent A. Let x be the minimum and maximum values of p for agent B. Let x be the agreement value and y the utility of x. Utility function of Agent A:

$$y = \frac{x - a_A}{b_A - a_A} \quad (4)$$

Utility function of Agent B:

$$y = \frac{a_B + x}{b_B - a_B} \quad (5)$$

In an optimal agreement, the utilities of both the negotiators are equal. So, equating (Eq. 4 and 5):

$$\frac{x - a_A}{b_A - a_A} = \frac{a_B - x}{b_B - a_B}$$

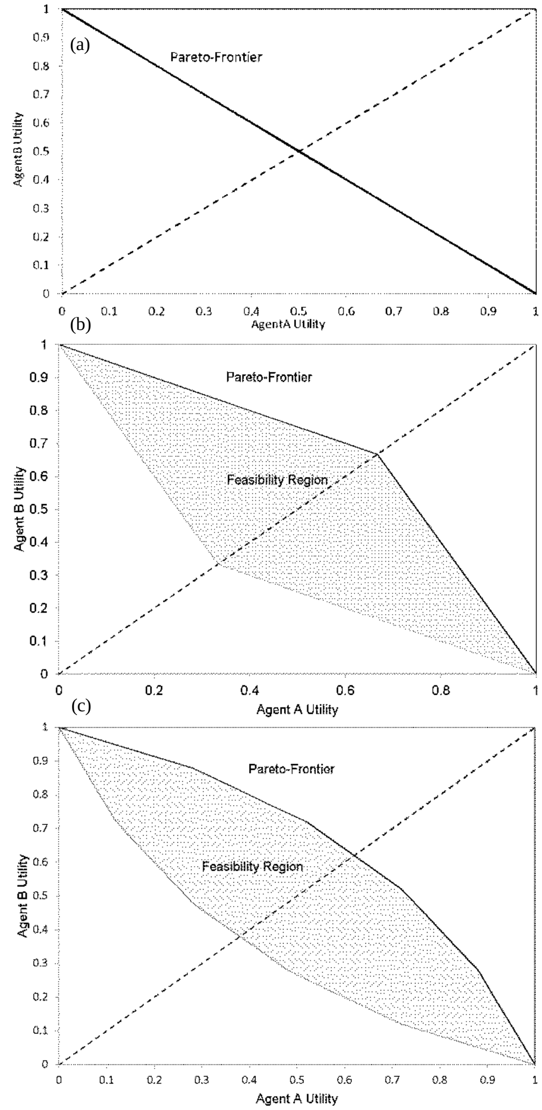

Fig. 2: a) Pareto-Frontier for single parameter negotiation; b) Pareto-Frontier and feasibility region for 2 parameter Negotiation and c) Pareto-Frontier and feasibility region for 5 Parameter Negotiation

Solving for x:

$$x = \frac{b_Ab_B - a_Aa_B}{(b_A - a_A) + (b_B - a_B)} \quad (6)$$

Substituting x in Eq. 4:

$$y = \frac{b_B - a_A}{(b_A - a_A) + (b_B - a_B)} \quad (7)$$

Similarly, when the utility function of agent B is increasing and agent A is decreasing x remains the same as in Eq. 6 and y is:

$$y = \frac{b_A - a_B}{(b_A - a_A) + (b_B - a_B)} \tag{8}$$

From Eq. 6 and 7 (a or b), (x, y) is the optimal point of agreement between the two agents for a single parameter negotiation.

Observation 1: When the midpoints of preferences of both the parties are the same, the utility of the Pareto-optimal agreement is 0.5 for both the parties (Fig. 3). The Pareto-frontier is shown in Fig. 2a.

Observation 2: When the midpoint of the decreasing utility function lies to the right of the midpoint of the increasing utility function, the Pareto-optimal utility lies above 0.5 (Fig. 4a and b). Otherwise the Pareto-optimal utility lies below 0.5 (Fig. 4c and d).

Observation 3: If the participants were to reverse their utility functions, the Pareto-optimal agreement remains the same as for their original utility functions but the utilities of the agreement changes to new utility = (1-old utility) for the participants.

Reaching the Pareto-optimal of individual parameters is possible by each agent stating offers alternatively along their utility function line starting from the maximum utility towards their minimum utility (Fig. 5). But this should be done with small decrements in each iteration for reaching the exact optimal.

When there are two or more parameters, reaching the optimum is not this simple. The parameters are attached with importance weights which complicate the process. If Pareto-optimal is reached for individual parameter utilities, it cannot be guaranteed that the total utility for the agents is Pareto-optimal (Lemma 1). Trade-offs between them could result in a better agreement than the aggregation of optima of individual parameters.

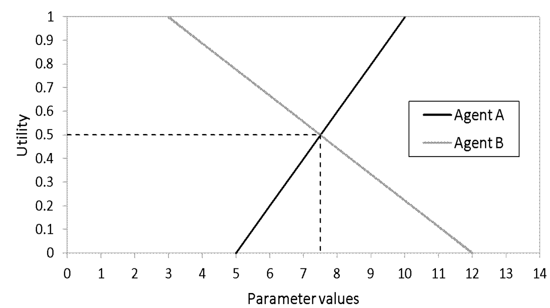

Fig. 3: Pareto-optimum at 0.5 [P: (min = 5; max = 10) C: (min = 3; max = 12)]

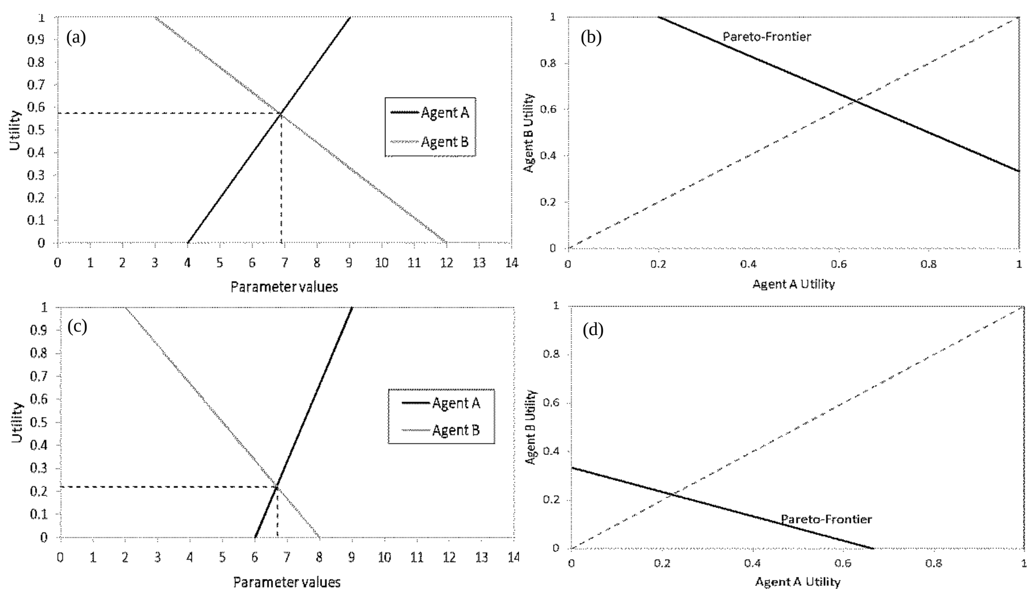

Fig. 4: a) Pareto-optimum higher than 0.5[A: (min = 4; max = 9) B: (min = 5; max = 11)]; b) Pareto-optimal frontier lying above 0.5 [A: (min = 4; max = 9) B: (min = 5; max = 11)]; c) Pareto-optimum lower than 0.5[A:(min = 6; max = 9) B: (min = 2; max = 8)] and d) Pareto-optimal frontier lying below 0.5 [A: (min = 6; max = 9) B: (min = 2; max = 8)]

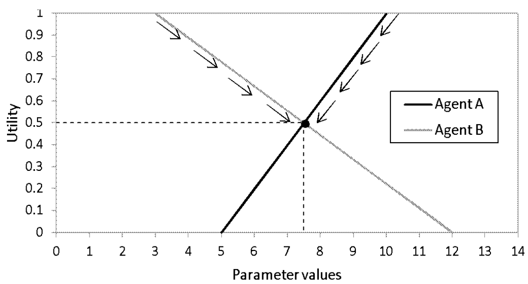

Fig. 5: Reaching Pareto-optimum

Lemma 1: When agreement for individual parameters is Pareto-optimal, the total utilities between the agents are not Pareto-optimal.

Proof: Let us consider negotiation between agents A and B over two parameters (1 and 2). When the individual utilities are Pareto-optimal for both the agents, the utilities of each parameter is the same as the opponent.

Let u_1^P and u_2^P be the Pareto-optimal point between A and B for parameter 1 and 2 individually. Therefore, from Eq. 3, the total utilities of A and B are:

$$u_{tot}^A = w_1^A u_1^P + w_2^A u_2^P \tag{8}$$

and:

$$u_{tot}^B = w_1^B u_1^P + w_2^B u_2^P \tag{9}$$

Let $w_1^A > w_2^A$ if A gives a trade-off by increasing u_1^P to u_1^A and decreasing u_2^P to u_2^A; $u_{tot}^A = w_1^A u_1^A + w_2^A u_2^A$ where u_{tot}^A remains the same as in Eq. 8. On agent B's side, u_1^P decreases to u_1^B and u_2^B increases to u_2^B. If $w_1^B < w_2^B$, u_{tot}^B in Eq. 9 increases. This is because the decrease in first term in Eq. 9 is less than the increase in the second term.

Thus, total utility of agent B has increased without a decrease in total utility of agent A. Therefore, after attaining Pareto-optimal of individual utilities it is possible to better the total utility of one agent without lowering the total utility of another agent. This implies that the total utilities are not Pareto-optimal.

Condition for positive trade-off: We define the following terminology to represent the impact of trade-off for an opponent.

A positive trade-off is a trade-off given by an agent during a bilateral negotiation that results in an increase of total utility for the opponent compared to previous proposal of the agent.

A negative trade-off is a trade-off given by an agent that results in a decrease of total utility for the opponent compared to previous proposal of the agent.

A zero trade-off is a trade-off given by an agent that makes no impact on the total utility for the opponent compared to previous proposal of the agent.

Lemma 2: The condition for positive trade-off between two parameters given by agent A is:

$$\frac{w_1^A}{w_2^A}\frac{(a_1^B - b_1^B)}{(a_1^A - b_1^A)} > \frac{w_1^B}{w_2^B}\frac{(a_2^B - b_2^B)}{(a_2^A - b_2^A)} \tag{10}$$

Where:

w_1 and w_2 = Weights of parameters 1 and 2
a_1 and a_2 = Maximum limits for parameters 1 and 2
b_1 and b_2 = Minimum limits for parameters 1 and 2

Proof: On Agent A's side (from Eq. 3):

$$u_{tot}^A = w_1^A u_1^A + w_2^A u_2^A$$
$$w_1^A < w_1^A$$

Where:

u_1^A = Increasing function
u_2^A = Decreasing function

On Agent B's side (from Eq. 3):

$$u_{tot}^B = w_1^B u_1^B + w_2^B u_2^B$$

$$w_1^B < w_2^B$$

Where:

w_1^B = Decreasing function
w_2^B = Increasing function

Let x_1 and x_2 be the values of parameters 1 and 2. Let us assume that the agent A gives a trade-off by increasing u_1^A and decreasing u_2^A such that the total utility on A's side remains constant. We have to find the condition for which A's trade-off results in an increase in total utility on agent B's side. Equating the increase and the decrease during trade-off on A's side:

$$w_1^A({}_{new}u_1^A - {}_{old}w_1^A) = w_2^A({}_{old}u_2^A - {}_{new}w_2^A)$$

Substituting for u in terms of a and b and solving we get:

$$\frac{({}_{new}x_2 - {}_{old}x_2)}{({}_{new}x_1 - {}_{old}x_1)} = \frac{w_1^A}{w_2^A}\frac{(a_1^A - b_1^A)}{(a_2^A - b_2^A)} \tag{11}$$

On B's side, u_1^B decreases and u_2^B increases. Total utility on B's side increases when:

$$w_1^B({}_{old}u_1^B - {}_{new}u_1 w_1^B) < w_2^B({}_{new}u_2^B - {}_{old}u_2^B) \Rightarrow$$
$$\frac{({}_{new}x_2 - {}_{old}x_2)}{({}_{new}x_1 - {}_{old}x_1)} > \frac{w_1^B}{w_2^B}\frac{(a_1^B - b_1^B)}{(a_2^B - b_2^B)} \tag{12}$$

From Eq. 11 and 12, we derive (Eq. 10):

$$\frac{w_1^A}{w_2^A}\frac{(a_1^B - b_1^B)}{(a_1^A - b_1^A)} > \frac{w_1^B}{w_2^B}\frac{(a_2^B - b_2^B)}{(a_2^A - b_2^A)} \tag{13}$$

When the condition in Eq. 10 holds true, trade-off by agent A will increase total utility for the agent B. A similar condition could be derived for a decreasing utility function of agent A and an increasing one for agent B.

If it is assumed that the ranges are same for both agents for both the parameters, weights assigned to parameters on each agent's side decides positive trade-off. But it is difficult to ensure or enforce same ranges of values to the agents as range selection is a matter of personal choice. An agent may intentionally choose a particular range to gain better from a trade-off. However, it is rational to assume that the ranges do not vary much between the agents and hence make only a minimal impact in a trade-off. On the other hand, weights may vary widely between the agents. Preferences of two negotiators may be exactly the same or totally the opposite. Therefore, in accordance with Eq. 10, even if the exact ratio of parameter weights is unknown, knowledge of the opponent's preference of one issue over another issue can result in a positive trade-off.

Trade-off algorithm: In this study, we first discuss a motivating example for the trade-off algorithm. Then we present and explain the trade-off algorithm. The concept of trade-off algorithm is based on the results established in the preceding sections.

Motivating example for trade-off algorithm: Let us assume a provider offering storage service to a consumer. The parties negotiate 4 parameters: storage capacity (in GB), availability (in %), response time (in ms) and price (in Rs.). Weights assigned to the parameters by the provider are 0.1-0.4, respectively. The consumer assigns 0.4, 0.3, 0.2 and 0.1 as weights of parameters.

The weights denote the importance that each party gives to each parameter. The values are assigned by a human user. A higher value of weight denotes more importance and vice versa. The weights are normalized such that the sum of all weights for each party is 1. In the example, the order of importance given to the parameters by the parties is completely the opposite. The provider values price the most while the consumer values storage capacity the most. Hence the ranking for the parameters [C, A, R, P] would be [4, 3, 2, 1] for the provider and [1, 2, 3, 4] for the consumer. Lower value of rank denotes higher importance.

For now let us assume that the ranking of importance is shared between the parties before negotiation. The actual weights are not shared. Since, the order of importance for consumer in this example is totally opposite to that of the provider, a tradeoff would yield good results. This is because the provider would be stringent on the price parameter while he would relax more on storage capacity. The consumer can utilize it and he can bargain more storage capacity while not bothering too much about paying higher price (because weight for price is just 0.1). The end result would be better total utility value for both the agents. Also, their convergence would be faster.

Opposite ranking order is an ideal case. There may be any combination of permutation of ranks for a pair of negotiators. When the rankings of both agents are almost similar, trading off a lower ranked parameter's utility for a better ranked utility will make the same changes for the other agent also. The more dissimilar the rankings are the better the trade-off would be. A rank correlation coefficient provides a good way to compare the order of importance between the negotiators. Pearson's coefficient, Kendall's Tau and Spearman's Rho are commonly used rank correlation coefficients. Experiments show that though the rank correlation coefficients do not exactly correspond to better trade-offs there is an overall trend that the lower the coefficient, the better the trade-off.

In Fig. 6 the graph shows the utilities of agent B after trade-off is given by agent A. The initial utility of B is 0.308 (marked by dashed line in Fig. 6). Kendall's tau of two different combinations of rankings may be the same (e.g., Tau for both (1243,1234) and (1324,1234) is 0.6667). Different combinations with same Kendall's tau show

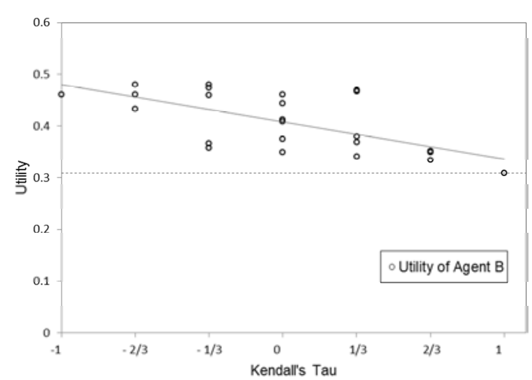

Fig. 6: Utility vs. Kendall's Tau

varied impact on the opponent side. Hence there are many plots for same Tau value. As the correlation gets more similar (as τ increases), trade-off by agent A generally yields lower utilities for the consumer as indicated by the trendline in Fig. 6. But the coefficient itself is not an exact indicator for the estimation of benefit of the opponent for a trade-off given by an agent. A graph similar to Fig. 6 is obtained when Pearson's correlation coefficient is used instead of Kendall's tau.

Trade-off algorithm: The proposed trade-off approach is implemented in Algorithm 1. The algorithm is O(n) and takes proposal (P) for which trade-off needs to be given, the corresponding weights (W), parameter ranking of the agent who gives the trade-off (R_{self}), parameter ranking of the opponent (R_{opp}) and the trade-off factor as inputs. The output of the algorithm is the counter-proposal calculated after trade-off.

In algorithm 1, first the utilities of parameters are calculated (lines 1-3). Then the utilities are categorized depending on the corresponding parameter's ranking (lines 5-12). A lower number of rank means a parameter is higher ranked. Hence if self rank number is lower than that of the opponent for a particular parameter, the parameter is ranked higher than the opponent and added to the array 'High'. Similarly utilities of lower ranked parameters and equally ranked parameters are added to the arrays 'Low' and 'Equal' respectively. The utilities in array High are increased by a factor x (lines 16-18). Then the utilities of array Low are decreased by a factor y such that the total utility (Total) remains the same (lines 22-24). 'y' is calculated as follows:

$$
\begin{aligned}
LowTotal_{new} &= sum(Low)-y(sum(Low)) \\
&= (1-y)\ LowTotal_{old} \\
\Rightarrow y &= 1-(LowTot_{new}/LowTotal_{old})
\end{aligned}
\tag{14}
$$

Finally, all the utilities are added to $utility_{new}$ array (line 25) and the corresponding parameter values for the counter proposal are calculated (lines 26-28).

Algorithm 1: Tradeoff (P, W, R_{self}, R_{opp}, x):

```
Input: P (p_1, p_2, ..., p_n), W (w_1, w_2, ..., w_n), R_self (r_1, r_2, ..., r_n), x
Output: CP (p_1, p_2, ..., p_n)
    for i = 1 to n
            Utility_old[i] calculateUtility(p_i)
    end for
    Total sum(Utility_old)
    for i = 1 to n
            ifR_self[i]<R_opp[i]then
                    add u_i×w_i to High
            else if R_self[i]>R_opp[i]then
                    add u_i×w_i to Low
            else
                    add u_i×w_i to Equal
            end if
    end for
    LowTotal_old sum(Low)
    EqualTotal sum(Equal)
    for each element∈High
            increase element by factor x
    end for
    HighTotal_new sum(High)
    LowTotal_new Total – HighTotal_new-EqualTotal
    y 1-(LowTotal_new/LowTotal_old)
    for each element Low
            decrease element by factor y
    end for
    add High, Low, Equal to Utility_new
    for i = 1 to n
            CP[i] calculateValue(Utility_new)
    end for
```

RESULTS AND DISCUSSION

The experiments focus on testing the efficiency of the trade-off algorithm in the proposed framework. The efficiency of the trade-off algorithm is tested in terms of number of rounds of negotiation required to reach an agreement and final utilities achieved. We test the effectiveness of the proposed approach using the following hypotheses.

Hypothesis 1: The efficiency of trade-off algorithm is not significantly affected if Kendall's tau between predicted and actual ranking is positive.

This hypothesis is to test the resilience of the trade-off algorithm. The hypothesis was tested by running negotiations with known opponent preferences on agent A's side and intentionally wrong opponent preferences on agent B's side. The actual ranking of parameters of agent A is (1-4). Figure 7a shows negotiation with correct opponent preferences on both sides. Figure 7b shows negotiation with agent B's prediction of agent A's preferences as (1-4). Kendall's tau between (1-4) and (1-4) is 0.667. Similarly the other graphs in Fig. 7 show negotiations between agents A and B with agent B's prediction wrong by the corresponding Tau value. The graphs show progressive diversion of consumer's proposals from the Pareto-Frontier as Tau decreases. The hypothesis is confirmed as the diversion from Pareto-line is minimal for predictions with positive tau values.

Hypothesis 2: The average number of rounds of negotiation required to reach an agreement using the proposed approach is less compared to other random approaches

The scalability of the proposed negotiation approach is also tested in this hypothesis. Negotiations were conducted by varying the parameters from 2-10 with same trade-off and concession factors. The average number of

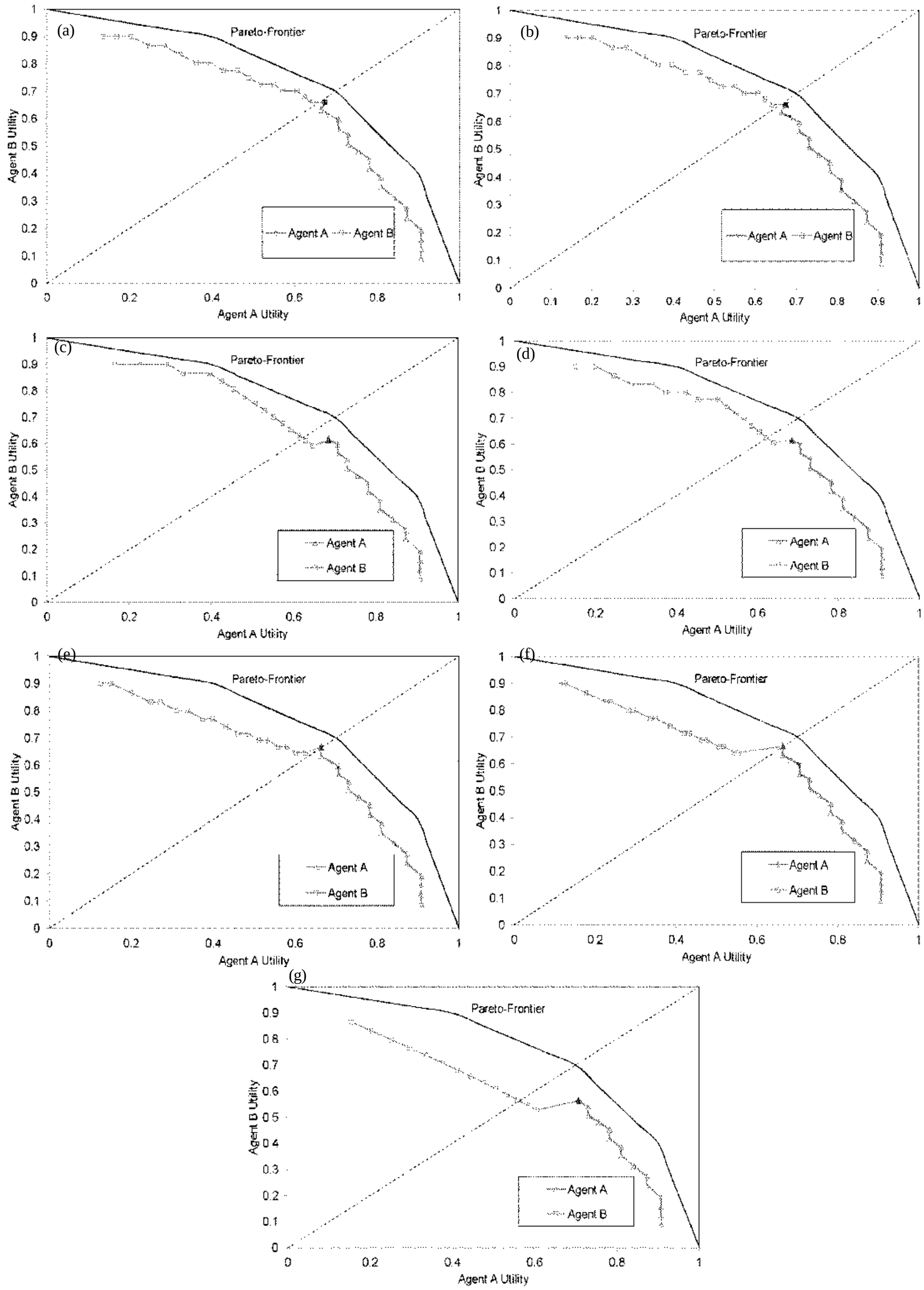

Fig. 7: Negotiation between agent A (known opponent preferences) and agent B (unknown opponent preferences): a) Tau = 1 between actual preference of A and B's prediction; b) Tau = 0.667between actual preference of A and B's prediction; c) Tau = 0.334between actual preference of A and B's prediction; d) Tau = 0between actual preference of A and B's prediction; e) Tau = -0.334between actual preference of A and B's prediction; Tau = -0.667between actual preference of A and B's prediction and) Tau = -1 between actual preference of A and B's prediction

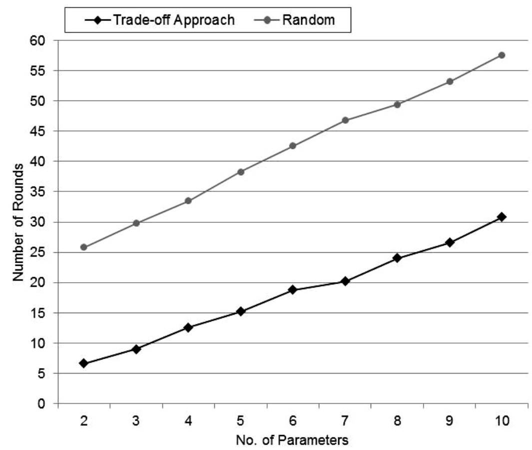

Fig. 8: Number of rounds of Negotiation varying number of parameters for the proposed trade-off approach and random approach

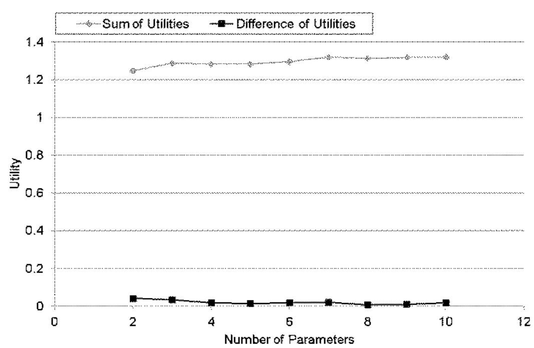

Fig. 9: Sum and difference of utilities for different number of parameters

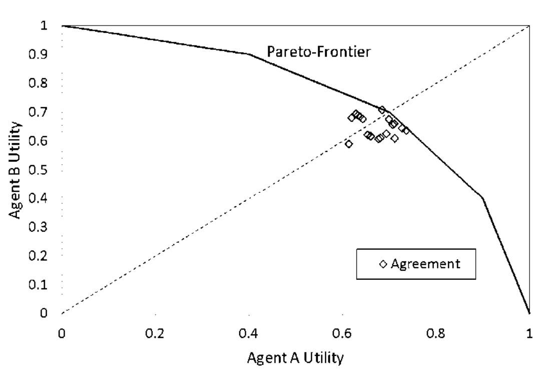

Fig. 10: Agreement points varying trade-off and concession factors

additional rounds required for each additional parameter is 3. The results are compared to random approach in Fig. 8.

Hypothesis 3: The proposed trade-off algorithm yields high social outcome. The sum of total utilities of the agents after agreement is achieved averages to 1.297 and the difference averages to 0.02. The high total and low difference indicates high joint outcome of the proposed approach. The total and difference for different number of parameters are shown in Fig. 9. It can be seen that the total and difference are almost constant independent of the number of parameters negotiated.

Hypothesis 4: The lower the trade-off and concession factors, the nearer the agreement is to the Pareto-frontier. Figure 10 shows the point of agreements over different trade-off and concession factors using the proposed trade-off algorithm. Higher values of trade-off and concession factor results in faster convergence but the points of agreement are relatively farther from the pareto-optimal. In Fig. 10 the points that lie closer to the Pareto-optimal line are the points-of-agreement of negotiations with lower trade-off and concession factors. This can be explained by the fact that when the factors are lower, more of the utility space is explored and hence better utilities.

CONCLUSION

In this study, we proposed a framework for SLA management and studied the properties of linear utility functions. Based on the inferences we proposed a trade-off algorithm that generates proposals that are more acceptable to the negotiation opponent. Generating such offers ensures faster negotiations and reduces the number of negotiations that do not reach an agreement thus decreasing the cost spent on unsuccessful negotiations. We plan to extend this paper for negotiations with incomplete information rather than partial information. We would also like to incorporate an opponent model to the proposed approach.

REFERENCES

Al-Aaidroos, M., N. Jailani and M. Mukhtar, 2011. Agent-based negotiation framework for web service's SLA. Proceedings of the IEEE 7th International Conference on Information Technology in Asia, July 12-13, 2011, Malaysia, pp: 1-7.

Alhamad, M, T. Dillon and E. Chang, 2010. Conceptual SLA framework for cloud computing. Proceedings of the 4th IEEE International Conference on Digital Ecosystems and Technologies, April 13-16, 2010, Dubai, pp: 606-610.

Cheng, C.B., C.C.H. Chan and K.C. Lin, 2006. Intelligent agents for e-marketplace: Negotiation with issue trade-offs by fuzzy inference systems. Decis. Support Syst., 42: 626-638.

Chhetri, M.B., J. Lin, S. Goh, J. Yan and J.Y. Zhang *et al*., 2006. A coordinated architecture for the agent-based service level agreement negotiation of web service composition. Proceedings of the IEEE Conference on Australian Software Engineering (ASWEC'06), April 18-21, 2006, IEEE, Australia, ISBN: 0-7695-2551-2, pp: 1-10.

DiNitto, E., M.D. Penta, A. Gambi, G. Ripa and M.L. Villani, 2007. Negotiation of Service Level Agreements: An Architecture and A Search-Based Approach. In: Service Oriented Computing. Bernd, J.K., J.L. Kwei and N. Priya (Eds.). Springer Berlin Heidelberg, Berlin, Germany, ISBN: 978-3-540-74973-8, pp: 295-306.

Faratin, P., C. Sierra and N.R. Jennings, 2002. Using similarity criteria to make issue trade-offs in automated negotiations. Artif. Intell., 142: 205-237.

Keller, A. and H. Ludwig, 2003. The WSLA framework: Specifying and monitoring service level agreements for web services. J. Network Syst. Manage, 11: 57-81.

Li, M., Q.B. Vo, R. Kowalczyk, S. Ossowski and G. Kersten, 2013. Automated negotiation in open and distributed environments. Expert Syst. Appl., 40: 6195-6212.

Ludwig, A., P. Braun, R. Kowalczyk and B. Franczyk, 2005. A Framework for Automated Negotiation of Service Level Agreements in Services Grids. In: Business Process Management. Christoph, J.B. and A. Haller (Eds.). Springer Berlin Heidelberg, Berlin, Germany, ISBN: 978-3-540-32595-6, pp: 89-101.

Ros, R. and C. Sierra, 2006. A negotiation meta strategy combining trade-off and concession moves. Auton. Agents Multi Agent Syst., 12: 163-181.

Vetschera, R., M. Filzmoser and R. Mitterhofer, 2014. An analytical approach to offer generation in concession-based negotiation processes. Group Decis.Negotiation, 23: 71-9910.1007/s10726-012-9329-z.

Wu, L., S.K. Garg, R. Buyya, C. Chen and S. Versteeg, 2013. Automated SLA negotiation framework for cloud computing. Proceedings of the 2013 13th IEEE/ACM International Symposium on Cluster Cloud and Grid Computing (CCGrid), May 13-16, 2013, IEEE, Delft, Netherlands, ISBN: 978-1-4673-6465-2, pp: 235-244.

Yan, J., R. Kowalczyk, J. Lin and M.B. Chhetri, S.K. Goh and J. Zhang, 2007. Autonomous service level agreement negotiation for service composition provision. Future Gener. Comput. Syst., 23: 748-759.

Zheng, X., P. Martin and K. Brohman, 2012. Cloud service negotiation: Concession vs. tradeoff approaches. Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster Cloud and Grid Computing (CCGrid2012), May 13-16, 2012, IEEE Computer Society, Washington, USA., ISBN: 978-0-7695-4691-9, pp: 515-522.

Zheng, X., P. Martin, K. Brohman and L. DaXu, 2014. Cloud service negotiation in internet of things environment: A mixed approach. IEEE. Trans. Ind. Inf., 10: 1506-1515.