Decreasing Impact of SLA Violations: A Proactive Resource Allocation Approach for Cloud Computing Environments

Hossein Morshedlou and Mohammad Reza Meybodi

Abstract—User satisfaction as a significant antecedent to user loyalty has been highlighted by many researchers in market based literatures. SLA violation as an important factor can decrease users' satisfaction level. The amount of this decrease depends on user's characteristics. Some of these characteristics are related to QoS requirements and announced to service provider through SLAs. But some of them are unknown for service provider and selfish users are not interested to reveal them truly. Most the works in literature ignore considering such characteristics and treat users just based on SLA parameters. So, two users with different characteristics but similar SLAs have equal importance for the service provider. In this paper, we use two user's hidden characteristics, named willingness to pay for service and willingness to pay for certainty, to present a new proactive resource allocation approach with aim of decreasing impact of SLA violations. New methods based on learning automaton for estimation of these characteristics are provided as well. To validate our approach we conducted some numerical simulations in critical situations. The results confirm that our approach has ability to improve users' satisfaction level that cause to gain in profitability.

Index Terms—Users satisfaction level, cloud service, resource allocation, willingness to pay, learning automaton

1 Introduction

User satisfaction as a significant antecedent to user loyalty in market based environments has been highlighted by many researchers [9]. Since in commercial environments such as cloud [8], users pay for service usages so their loyalty has direct effect on profitability. Gaining satisfaction from users depends on different parameters. Some parameters are server side, e.g., QoS parameters, but some other depends on user's characteristics and differs from one user to another, e.g., risk aversion. SLA violation as an important factor can make users malcontent and decrease their satisfaction level. Amount of the decrease depends on the mentioned characteristics. Some of these characteristics are unknown for service providers and selfish users are not interested to reveal them truly. Most the works in literature [9], [22], [32] ignore considering such characteristics and treat users just based on their SLA parameters. So, two users with different characteristics but similar SLAs have equal importance for a service provider.

In this paper, we present a new approach to reduce impact of SLA violations on users' satisfaction level, but not by decreasing number of SLA violations as must the works in literature do. Instead we try to use characteristics of users to decrease impact of SLA violation on users' satisfaction level (USL) (Throughout this paper SLA violation means discarding user's request or not serving it before its deadline). We investigate the mentioned approach in a resource allocation scenario. We use two characteristics, called willingness to pay for service (WTP) and willingness to pay for certainty, to present a new proactive resource allocation approach. Values of these characteristics are unknown for service provider.

To illustrate applicability of our approach, we provided some numerical simulations in critical situations. Critical situation means a condition in which a service provider has faced with unforeseen lack of resources and SLA-violation for some users is inevitable. For example, step 5 in Fig. 1 has failed and there are no adequate resources to serve all the accepted queued requests. Service provider has two options: discarding the queued requests or forcing some existing VMs to release their resources. In latter case, service provider can allocate the released resources to the significant queued requests. Both cases would cause SLA violations, but the users who suffer from violations are different. Since users have different characteristics so users' satisfaction level under the two mentioned options will be different. Our approach presents a method to decide which VMs should be enforced to release their resources.

The remaining of this paper is organized as follows: some related work is reviewed in Section 2. Section 3 introduces some preliminary concepts. In Section 4, user discrimination measures are introduced. Section 5 contains system description and representation of resource allocation problem in form of pairing process. In the following, mapping of pairing process to pairing game is provided and outcome of pairing process is analyzed using game theoretic concepts. Numerical results are provided in Section 6 and we conclude paper in Section 7.

- *H. Morshedlou is with the Department of Computer Engineering and Information Science, AmirKabir University of Technology, Tehran, Iran. E-mail: morshedlou@ aut.ac.ir.*
- *M.R. Meybodi is with the Department of Computer Engineering and Information Technology, AmirKabir University of Technology, Tehran, Iran. E-mail: mmeybodi@aut.ac.ir.*

Manuscript received 14 Sept. 2013; revised 11 Jan. 2014; accepted 21 Jan. 2014. Date of publication 6 Feb. 2014; date of current version 30 July 2014.
Recommended for acceptance by I. Bojanova, R.C.H. Hua, O. Rana, and M. Parashar.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCC.2014.2305151

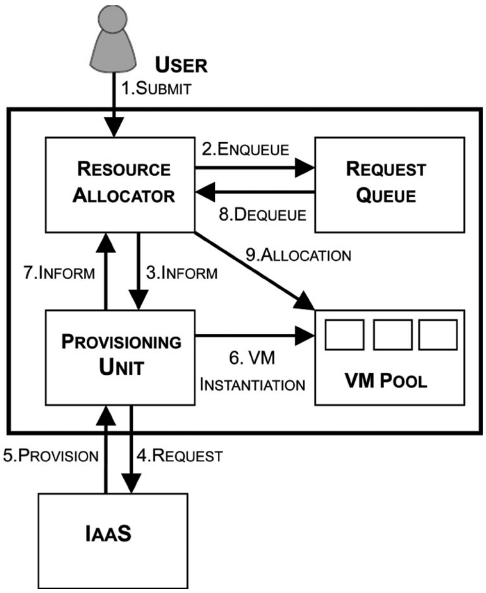


Fig. 1. Cloud service provider procedure.

2 Related Work

Many attempts [8], [10], [11], [16], [22], [29], [31], [32] have been done in looking for approaches and policies for resource allocations on dynamic resources such as clouds. Most the works in literature employ SLA and QoS constraints for resource allocation and ignore considering user's characteristics. Profitability as objective function plays an important role in decision making while user satisfaction as a measure, with indirect effect on profitability, has not received enough attention. In following we provide a fast review on some related works in SLA-based resource provisioning and allocation and user satisfaction subjects.

Villegas et al. [31] focus on a system model where all resources are provided from IaaS clouds previously. Although our approach has ability to be employed by systems with dynamic resource provisioning, but its efficiency will be more evident when facing with lack of resource in previously provided resources. Mao et al. [23] present an approach for autonomic provisioning to users with deadlines. With having selfish agents in autonomic approaches, we expect these agents try to improve satisfaction level of their associated users. However this approach doesn't include user's hidden characteristics in decision making of autonomous agents. Song et al. [30] and [33] presents multilevel scheduling methods to handle resources based on SLAs. Both these works act just based on SLA parameters. Wu et al. [32] suggests resource allocation algorithms for cloud providers who want to reduce cost and SLA violations. Design of the proposed algorithms in [32] makes cloud providers capable to handle the dynamic change of users, mapping requests to resource level parameters and managing diversity of VMs. It also considers the users' QoS parameters and resource level parameters such as service start time. In service oriented environments there are several works for addressing the dynamic and negotiable SLAs [12]. Offering negotiable SLA during offering a service can be useful especially when provider faces with critical situation. Through negotiation for lower QoS, provider can decrease effect of resource leakage. However, due to complexity of negotiable SLAs, none of the commercial IaaS providers in real world is offering this type of SLAs. What real world providers usually offer is an SLA contract that specifies simple parameters like as network availability or deadlines. So handling critical situations through negotiation is somewhat complicated. Our proposed approach tries to decrease impact of SLA violation through a user discrimination approach and avoid entering to complex procedure of negotiation.

From marketing viewpoint in service industries, there are some works [7], [19], [20], [24] in literatures that consider relation among service quality, customer satisfaction, customer loyalty and repurchasing intentions. For example, [24] present a conceptual model for relating customer satisfaction and repurchase behavior. The model is based on the premise that satisfaction ratings observed in a typical survey are error-prone measures of the customer's true satisfaction and they may vary on the basis of customer characteristics. Bolton [7] develops a dynamic model of the duration of provider-customer relation that focuses on the role of customer satisfaction. This model helps provider to identify specific actions that can increase retention and profitability in the long run. Keaveney [19] conducted an experimental study on customer switching behavior in service industries. It shows that customer switching behavior damages market share and profitability of service firms. Lam et al. [20] discusses about relationships among customer satisfaction, loyalty and customer switching. Reported results in marketing literatures state that customer satisfaction directly influence on profitability of a firm in long term.

3 Preliminary Concepts

In this section we offer a brief introduction for users' satisfaction level and willingness to pay for service. For concepts of risk aversion and willingness to pay for certainty we refer reader to [36].

3.1 Learning Automaton

A learning automaton is an adaptive decision-making unit [25], [26] located in a stochastic environment that learns the optimal action through repeated interactions with environment. The action selection is based on a specific probability distribution which is updated according to environment response the automaton receives by employing a particular action. Learning automata and its extensions [4], [5], [6] have wide range of applications in various domains such as particle swarm optimization (PSO) [18], [34], wireless sensor networks [13], [14], cellular mobile networks [3], stochastic graphs [1] and, etc.

3.1.1 Finite Action-Set Learning Automaton (FALA)

FALA is type of variable LA that its action-set is always considered to be finite. This type of LA has been studied extensively in many applications. Let $\alpha_i(k) \in \alpha(k)$ denotes the selected action by learning automaton based on the probability distribution $p(k)$ defined over the action set at instant k. Let a and b be learning rates which are associated with reward and penalty parameters. r denotes the number of actions that can be taken. If the selected action $\alpha_i(k)$ receive reward then the probability vector $p(k)$ is updated using (1). In case of receiving penalty, (2) is used instead of (1):

$$p_i(n+1) = \begin{cases} p_i(n) + a[1 - p_i(n)] & i = j \\ (1-a)p_i(n) & \forall i, i \neq j \end{cases} \tag{1}$$

$$p_i(n+1) = \begin{cases} (1-b)p_i(n) & i = j \\ \left(\frac{b}{r-1}\right) + (1-b)p_i(n) & \forall i, i \neq j. \end{cases} \tag{2}$$

The learning algorithm will be called linear reward-penalty or L_{R-P} if $(a = b)$. In case of $a \gg b$, we call it the learning reward-ε penalty or $L_{R-\varepsilon P}$ and finally if $(b = 0)$ they are called linear reward-Inaction algorithms (L_{R-I}).

3.2 Users' Satisfaction Level

We define users' satisfaction level based on expected value of user's utility as in (3):

$$USL(t) = \frac{\sum_{i \in Users} u_i(w_i(t))}{number\_of\_users}. \tag{3}$$

A utility function, u(w), measures the value or utility that an user attaches to the monetary amount w. note that the service provider is not aware of user's utility function. Throughout this paper we assume that all users have almost same preliminary amount of wealth w but different utility functions. $w_i(t)$ in (3) shows the user i's profit by using the service at iteration t. $w_i(t)$ depends on different parameters such as price of service, whether the user has faced SLA violation or not and etc. Equation (4) shows a possible way for computation of $w_i(t)$. $request \in \{0, 1\}$ presents the existence of request from user i in iteration t. $violation \in \{0, 1\}$ shows that user i has faced with SLA violation (violation $= 1$) in iteration t or not (violation $= 0$):

$$\begin{aligned} w_i(t) = request \times (WTP_i \times (1 - violation) \\ -price(t) + Penalty \times violation) + w_0. \end{aligned} \tag{4}$$

We also assume that each utility functions satisfy the conditions $u\prime(w) > 0$ and $u''(w) < 0$. The former condition states that a user prefers more money or wealth to less. This assumption seems perfectly rational. The latter states that as the user's wealth increases, he/she places less value on a fixed increase in wealth.

3.3 Willingness to Pay for Service

Willingness to pay for service illustrate the value of service for a user. In other words, WTP is maximum monetary amount that user is actually willing to pay for service. Higher WTP means higher importance of service and it can be used as a decision measure among users. Since a user with higher WTP has potential to pay more for services, so his/her satisfaction would cause higher profitability for service provider. For two users A, B with $WTP_A > WTP_B$, a changes in price at a particular level has less influence on A's request rate than B's. Assigning higher priority to the users with higher WTP will improve service provider's income.

4 User Discrimination Measures

The service provider needs two parameters to specify users' importance, penalty and WTP. Penalty is determined in SLA and can be used as a representative measure for willingness to pay for certainty. Since an individual's decision about amount of penalty is not deterministic process [17] and for similar requests at different times user may propose different penalties, so instead of penalty parameter of a request, we use a long term measure. Section 4.1 describes how this long term measure can be extracted from penalty values of user's requests. The second parameter, WTP [15], is unknown for service provider. A selfish user doesn't like to release it truly. So service provider should estimate it using user's reactions to different prices. We present a new method in Section 4.2 for estimation of this parameter.

4.1 Long Term Measure of Penalty

We use a learning automaton to learn the long term measure for user. Probability vector of this learning automaton is kept as part of user profile. We assume that all penalty values are from a discrete and finite set L whereas each member of L has an acceptable value for service provider. Learning automaton has an action associated with each value in L. When service provider handles user's request, loads probability vector of learning automaton from user's profile and using it, chooses an action. Service provider proposes associated value of the selected action as penalty to user. If user accept this value then learning automaton rewards the selected action else punishes it. If user rejects the service provider's proposal then he/she can propose another value from L as penalty. Learning automaton rewards the associated action of the user's proposed value. When a user accepts a penalty proposal or proposes a penalty value, he/she should pay $\alpha = \mathrm{I}(\mathrm{Penalty})$ as well as service price to service provider. Function I is a strictly increasing function and can be defined based on premium calculation concepts in insurance field [36]. We define the long term measure (M_{LT}) using probability vector of learning automaton as (5):

$$M_{LT} = \sum_{i=1}^{num\_of\_actions} p_i \times associated\_value(action(i)). \tag{5}$$

Service provider uses value of M_{LT} to establish importance of a user, but when fails to meet SLA, Penalty value of request must be paid to compensate user's loss.

4.1.1 *Penalty Value: A Useful Measure*

$M_{LT}$is approximately the expected value of penalties which user has proposed or accepted. Higher M_{LT} means user has requested higher penalties in long term so he/she has paid more for certainty (recall user pays $\alpha = I(Penalty)$ in addition to service price). We know from insurance concepts [19] that more payments for certainty means more risk aversion. So M_{LT}can be used for ranking users based on their risk aversion and this ranking is similar to ranking based on user's willingness to pay for certainty. Since we aim to improve USL so according to Proposition 1, M_{LT} is a useful measure to lead us to achieve our aim.

Proposition 1. *consider n similar users but with different level of risk aversion. When a service provider can serve just m users* $(m < \mathrm{n})$ *using its available resources, choosing m users who are more risk averse results in maximum possible USL.*

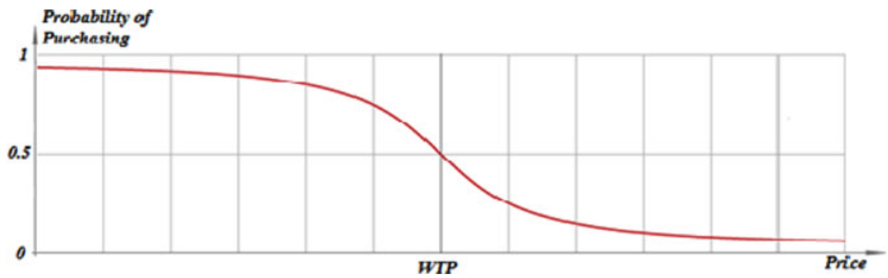


Fig. 2. Probability/Price decision model: This diagrams shows probability of purchasing a service in a particular price.

In following we present a verification for the above proposition for $n = 2$ and $m = 1$. This verification can be extended for any values of n, m.

Consider two users (user1 and user2) who benefit from a cloud service. Service utility for user i ($i = 1, 2$) is shown by $u_i(w_0)$. Consider a situation where service provider faces resource leakage and available resources are enough just to serve one of the users. The user whose request is ignored by service provider, loses X and his utility decreases to $u_i(w_0 - X)$. Assume that user1 is more risk averse than user2. If serving both users has equal profit, ignoring which user is a better choice?

We know a risk averse individual is indifferent between certain condition $(w_0 - EX - \rho_i)$ and risky condition $(w_0 - X)$ [36]. This means (6):

$$u_i(w_0 - EX - \rho_i) = E(u_i(w_0 - X))i = 1, 2. \tag{6}$$

For a risk neutral individual ρ_i is equal to zero, but for a risk averse one ρ_i is greater than zero. $(EX + \rho_i)$ is willingness to pay for certainty. From (6) we have (7):

$$w_0 - EX = u_i^{-1}(E(u_i(w_0 - X))) + \rho_i i = 1, 2. \tag{7}$$

Equation (8) and (9) can be concluded using (7):

$$u_1^{-1}(E(u_1(w_0 - X))) + \rho_1 = u_2^{-1}(E(u_2(w_0 - X))) + \rho_2, \tag{8}$$

$$\rho_1 - \rho_2 = u_2^{-1}(E(u_2(w_0 - X))) - u_1^{-1}(E(u_1(w_0 - X))). \tag{9}$$

Risk aversion of user i (RA_i) can be estimated using its utility function [36]. Since user1 is more risk averse than user2 we have (10):

$$\frac{u_1''(w)}{u_1'(w)} - \frac{u_2''(w)}{u_2'(w)} = \frac{u_1''(w)u\prime_2(w) - u_2''(w)u_1'(w)}{u_1'(w)u_2'(w)} < 0. \tag{10}$$

Equation (11) is a straight forward statement:

$$\int \frac{u_1''(w)u_2'(w) - u_2''(w)u_1'(w)}{u_1'(w)u_2'(w)} dw = \log \frac{u_1'(w)}{u_2'(w)} + c. \tag{11}$$

According to (10), RHS of (11) is decreasing. This means that $u_1'(w)/u_2'(w)$ is decreasing as well. So (12) can be concluded:

$$\frac{d}{dw}\frac{u_1'(w)}{u_2'(w)} < 0. \tag{12}$$

Moreover because $u_i'(w) > 0$ (for $i = 1, 2$) we have (13):

$$\frac{u_1'(w)}{u_2'(w)} > 0. \tag{13}$$

Let $z = u_2(w)$. Using this notation $u_1'(w)/u_2'(w)$ can be rewritten as $u_1'(u_2^{-1}(z))/u_2'(u_2^{-1}(z))$. We have also (14):

$$\int \frac{u\prime_1\left(u_2^{-1}(z)\right)}{u\prime_2\left(u_2^{-1}(z)\right)} dz = u_1\left(u_2^{-1}(z)\right) + c. \tag{14}$$

First derivative of $u_1(u_2^{-1}(z))$ is positive according to (13) and since $dz/dw > 0$ then $u_1'(u_2^{-1}(z))/u_2'(u_2^{-1}(z))$ is decreasing according to (12). So $u_1(u_2^{-1}(z))$ is a concave function and by Jensen's inequality, we have (15):

$$E\left(u_1(u_2^{-1}(z))\right) \leq u_1\left(u_2^{-1}(E(z))\right). \tag{15}$$

Now if we put $z_0 = u_2(w_0 - X)$ or $w_0 - X = u_2^{-1}(z_0)$ then substituting this to (8) results in (16):

$$\rho_1 - \rho_2 = u_2^{-1}(E(z_0)) - u_1^{-1}\left(E\left(u_1\left(u_2^{-1}(z_0)\right)\right)\right). \tag{16}$$

Since $u_1' > 0$ then by (15) we have (17):

$$u_2^{-1}(E(z_0)) \geq u_1^{-1}\left(E\left(u_1\left(u_2^{-1}(z_0)\right)\right)\right) \Rightarrow \rho_1 \geq \rho_2. \tag{17}$$

Now assume that service provider tries to preserve USL equal to $(u_1(w_0) + u_2(w_0))/2$. If user2 is ignored then service provider should pay $(EX + \rho_2)$to compensate user2's loss to attain the mentioned USL while by ignoring user1, these increases to$(EX + \rho_1)$. So a rational service provider ignores user2.

4.2 Estimation of WTP and Demand Function

Decision making about purchasing a product is not a deterministic process and probability of purchasing depends on different parameters like as price, income and, etc. This probability is close to 1 if price is too much smaller than WTP of a user. When price approaches to WTP, this probability decreases and for prices greater than WTP, approaches to zero. It seems that probability of purchasing decrease drastically around WTP and price sensitivity of a user reaches maximal value. For example, Fig. 2 shows a probability function that is proper for describing user behavior in different prices.

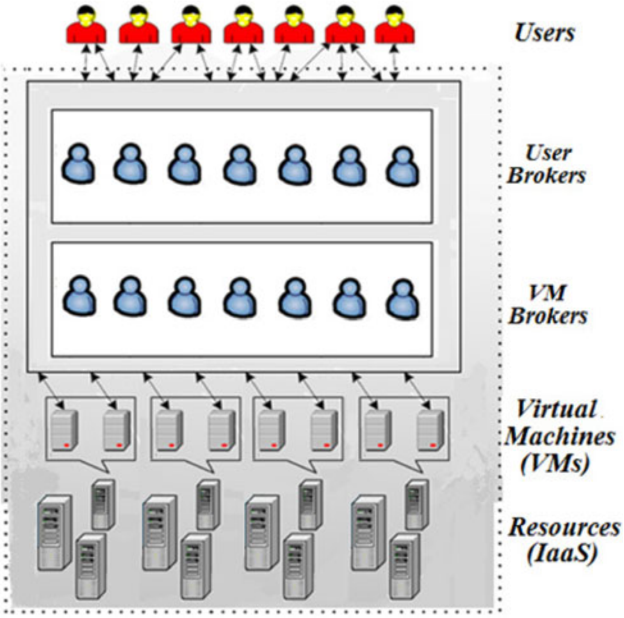


Fig. 3. Architecture of cloud service provider.

In this section using this fact we present a new method to estimate WTP of a user. In economic, concept of price-elasticity of demand (e_p) is used to represent price sensitivity of a user. Definition of e_p is as (18):

$$e_p = \frac{dQ/Q}{dp/p} = \frac{p}{Q} \times \frac{dQ}{dp}. \quad (18)$$

Having demand curve $(Q = D(p))$, we can rewrite (18) as (19):

$$e_p = \frac{p}{D(p)} \times \frac{dD(p)}{dp}. \quad (19)$$

Since e_p represent price sensitivity of a user and around WTP price sensitivity reaches its maximum, so it seems that maximum absolute value of e_p is a good estimation for WTP. So we use (20) to estimate WTP:

$$WTP \approx \max_p \left(\left| \frac{p}{D(p)} \times \frac{dD(p)}{dp} \right| \right). \quad (20)$$

To estimate demand function, service provider establishes the request rate of each user at a given price. Having (price, request_rate) pairs, demand curve can be easily estimated by fitting a curve to the pairs.

5 System Description

Cloud system architecture for supporting our resource allocation approach is shown in Fig. 3. There are some users with different request rates. We assume that maximum request rate of a user will be one request per iteration. In an iterative procedure, service provider charges users with a variable but equal price for all requests. Each user submits his/her request to a queue. Each request has a deadline time and we assume that all requests are similar. In our system, service provider serves the requests in parallel mode (process n requests per iteration from the queue) and in non-critical situation, service provider has enough time to serve all the requests before their deadlines. While this is not the case for critical situations and service provider has to ignore some requests. Having removed a request from the queue, it is assigned to a user broker (UB). Total of n new virtual machines with different level of reliabilities are instantiated and each one is assigned to a VM broker (VB). All VMs are similar in properties except reliability level. When service provider faces risk of resource leakage or its resource provisioning policy fails, different level of reliability can be used as a measure to show which VMs are better candidates to release their resources. Releasing resources of these VMs makes it possible to use the released resources for handling significant requests of the queue. This approach doesn't decrease number of SLA Violations but leastwise service provider tries to discard the requests which are less important. Each iteration, based on importance of users and reliability level of virtual machines, total of n pair wises (each composed of a user broker and a VM broker) can be constituted. More details about the system are as belows:

1. User brokers are informed about different reliability level of VMs, but users are not. A user broker as an autonomous entity and agent of user tries to pair with the associated VM broker of the most reliable VM available to him.
2. Service provider creates a profile for each user. This profile contains two parameters named WTP and M_{LT}. WTP is user's estimated willingness to pay for service and M_{LT} reflects the user's flexibility respect to SLA violations.
3. VM broker i uses a function $d_{user}(WTP_j, M_{LT}(j), R_i)$ to approximate the importance of user j. WTP_j and $M_{LT}(j)$ are user j's parameters and R_i is VM i's reliability parameter. VM broker uses importance of users to rank the user brokers. d_{user} is increasing with WTP and Penalty. Behavior of d_{user} with respect to R depends on values of the two former parameters.
4. User brokers use function $d_{VM}(R)$ to rank VMs. Since $d_{VM}(R)$is independent of user broker, so all user brokers have same ranking of VMs (e.g., reliability based) while this is not the case for VM brokers. $d_{user}(WTP_j, M_{LT}(j), R_i)$ depends to R_i and each VM broker has a different ranking of users. Since a broker prefers A to B if A is located before B in its ranking so d_{VM} or d_{user} also is called preference measures.

Definition 1. *Preferences are cycle-free if and only if there is no sequence of brokers $b_1, b_2, \ldots, b_k$ of length $k > 2$ such that each broker b_i prefers b_{i+1} to b_{i-1} (if $i = k$ put $i + 1 = 1$). Notice that in $b_1, b_2, \ldots, b_k$ for each i, $b_{i-1}$and b_{i+1} are from same type (e.g., VM broker) and different with b_i.*

Lemma 1. *If brokers use d_{VM} and $d_{user}$to determine their preferences, then the obtained preferences are cycle-free.*

Proof. Assume that these preferences are not cycle-free. So based on definition 1, we must have a sequence of brokers, $b_1, b_2, \ldots, b_k$, such that each broker b_i prefers b_{i+1} to b_{i-1}. Let b_i to be a user broker. Since all the user brokers have same ranking of VMs and preferring is transitive, so each broker b_i prefers b_{i-1} to b_{i+1}. This result is in contradiction with definition, so existence of such sequence of brokers is not possible. Then the obtained preferences are cycle-free. □

Definition 2. *A pairing process is an iterative process that each iteration, two brokers (one user broker and one VM broker) who are guaranteed to reach maximum utility by pairing to each other, are eliminated from the sets. After last iteration both sets should be empty.*

Theorem 1. *A pairing problem among user brokers and VM brokers by the obtained preferences using d_{VM} and d_{user} has a pairing process.*

Proof. Start with an arbitrary broker b_1 and constitute a sequence $b_1, b_2, \ldots$ in which b_{i+1} is the most preferred user (VM) broker by VM (user) broker b_i out of other user (VM) brokers. Since number of brokers is finite, so the sequence must have a cycle. Based on Lemma 1, the obtained preferences using d_{VM} and d_{user} are cycle-free so the cycle must be of length 2. This means that we have found two brokers that prefer each other the most, so they reach maximum utility by pairing to each other. By eliminating these two brokers from broker sets, at the next iteration we have a similar process with fewer brokers (with $n-1$ broker in each set). This process resumes until last iteration (iteration n). □

To achieve profitability and proper USL, it seems that each broker should try to maximize its utility. But does the mentioned pairing method satisfy aim of the brokers and is the pairing stable? To answer let to study this problem using game theory concepts. Mapping of this problem into games is done at the next section.

5.1 Pairing Games

We define the pairing problem in form of two games: UB and VB Games. In below definitions $r_i(j)$ denotes user/VM broker j's rank in VM/user broker i's ranking. S is set of all strategy profiles that players may choose and s_i denotes strategy of player i.

5.1.1 UB Game

The user brokers are players of this game and we assume that VM brokers are part of environment. Since brokers (both type) are utility maximizer, they choose best strategy. In UB Game, each player's strategy space is set of all VM brokers. For each strategy profile $s \in S$, we have utility of player $i, u_i(s) = n - r_i(j) + 1$ iff $s_i = j$ and there is not a player $k(k \neq i)$ such that $s_k = j$ and $r_j(k) < r_j(i)$, otherwise $u_i(s) = 0$.

5.1.2 VB Game

Definition of this game is similar to UB game, but here VM brokers are players.

5.1.3 Game Theoretic Analysis

Theorem 2. *The outcome of pairing process for pairing of user brokers and VM brokers is a pure Nash Equilibrium (PNE) point of UB and VB games.*

Proof. All the user brokers have same ranking of VM brokers. Let VM_i denotes to VM broker which is in ith place of this ranking and UB_i is a user broker which has paired to VM_i in pairing process. Assume VM_i deviates unilaterally and chooses UB_k $(k \neq i)$. Then for the case $r_{VM_i}(UB_k) > r_{VM_i}(UB_i)$ clearly this deviation is not profitable, but if $r_{VM_i}(UB_k) < r_{VM_i}(UB_i)$ then UB_k has paired with a VM_k such that VM_k has higher rank than VM_i and every user broker prefers VM_k to VM_i. So UB_k rejects VM_i and based on definition of VB game $U_{VM_i}(s_{VM_i} = UB_k) = 0$. This shows that the latter case is not profitable as well. So the outcome of pairing process is a pure Nash Equilibrium of VB game. For UB Game, All the players of UB game prefer VM_i to VM_{i+1}. So deviation of $UB_1$is not profitable. Deviation of UB_2 will be profitable, if VM_1 prefers UB_2 to UB_1. But this is not the case for UB_2, since VM_1 prefers UB_1 to all other user brokers. This means that deviation of UB_2 is not profitable as well. Similarly for the rest of user brokers deviation is not profitable and outcome of pairing process is a pure Nash Equilibrium of UB game, too. □

Theorem 3. *Both UB and VB games have unique pure Nash Equilibrium point.*

Proof. For VB Game, let $\overline{NE}$ denote the outcome of pairing process. Assume that there exists another Nash Equilibrium $\underline{NE}$ $(\underline{NE} \neq \overline{NE})$. Definitions of VM_i and UB_i are similar to Theorem 2 and $s_B(N)$ denotes to the selected strategy of user/VM broker in associated strategy profile to point N of the game. If $s_{VM_1}(\underline{NE}) \neq UB_1$ then deviation of $VM_1$to UB_1 is profitable and $\underline{NE}$ is not a Nash. In case of $s_{VM_1}(\underline{NE}) = UB_1$ we have $s_{VM_1}(\underline{NE}) = s_{VM_1}(\overline{NE})$. In a similar way it can be shown that for each i, the condition$s_{VM_i}(\underline{NE}) = s_{VM_i}(\overline{NE})$ is needed for $\underline{NE}$ to be a Nash. So $\overline{NE}$ is unique. Proof for UB game is similar to VB game. □

As a result of Theorems 2 and 3, the outcome of pairing process is unique pure Nash Equilibrium point of UB and VB games. Since playing best response strategy in games with unique pure Nash equilibrium point converges to that PNE [28] so the pairing process satisfies the utility maximizer brokers and it will be stable.

6 Numerical Results

In this section we present some numerical results. But first we describe user decision model, assumptions and user types. Defining such user decision or demand models is usual in economical literatures [2].

6.1 User's Decision Model

We said before that decision making of a user is not a deterministic process [17]. Moreover it seems that this process is not even stationary and changes according to different parameters, but we assume it is stationary for the interval that service provider offers its service. We use probability function (21) for modeling the user's decision making:

$$p(pr) = 0.5 - \arctan(pr - WTP)/\pi. \tag{21}$$

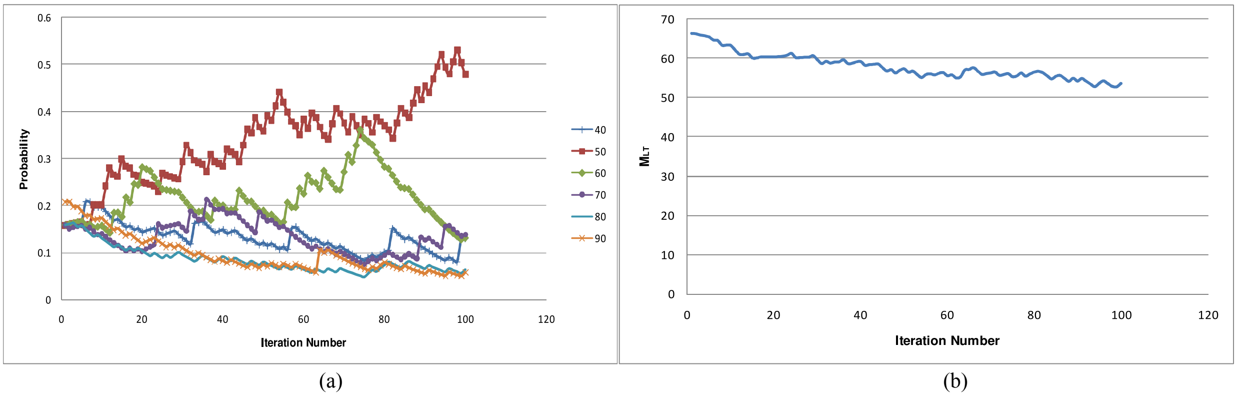


Fig. 4. (a) Probability vector of learning automaton during iterations. (b) Estimated value of long term measure.

Where $p(price)$ is the probability a user with WTP, as his/her willingness to pay, requests for a service in price *pr*. Penalty is another parameter that should be chosen by user as described before. Since this option act as a lateral insuring service and is not part of main service, so user should be charged for it separately. Another parameter is violation impact β. Each SLA violation provokes a user to change his/her service provider. More risk averse user is more excitable and has a higher violation impact. Violation impact parameter represents user sensitivity to SLA violations and differs from one user to another. For example, probability of reusing a service by a particular user, who has experienced SLA violation for n times, will be $(p(pr) - n \times \beta)$.

6.2 Configuration and Results

In this section, first we present results of the proposed methods for estimation of M_{LT} and WTP. After that, the results for the proposed approach will be discussed. Suppose a user with WTP $= 55$, who is interested to receive 50 unit as penalty for each SLA violation. The user may accept other penalty values as well but prefers the mentioned penalty. Discussion about decision making of individuals about how much is enough for protection against damage or loss is out of scope of this paper and we refer readers to premium calculation literatures [21].

Long-term measure estimation. Because more risk averse users have higher willingness to pay for certainty, so choosing or accepting the penalty values from L which are appropriate to their risk aversion is more probable. To model this behavior, we use a vector for each user. ith element of this vector is the probability that user will accept or choose ith element of L as penalty. We use a learning automaton with L_{R-P} algorithm to learn this decision making model. Fig. 4a shows the probability vector of a learning automaton during 100 iterations. Number of actions is $|L|$ and according to changes of probability vector in this figure, user prefers action 50. Fig. 4b depicts changes of M_{LT} during the mentioned iterations in which user requested for service 100 times. The estimated M_{LT} is earned using (5) and as illustrated in Fig. 4b approaches to 53.

WTP: To estimate a user's WTP, the proposed method uses request rate of a user in n different prices to earn its demand curve $D(p)$. Having this curve, the method finds the maximum of $|e_p|$ in (19). Fig. 5 shows the estimated diagram of $|e_p|$ for different number of points. As depicted in Fig. 5, the estimated WTP is either 55 or very close to 55. This figure also shows that more number of points doesn't guarantee more precise estimation.

The proposed approach. For simulations we considered five different user types and five classes of virtual machines with different reliability level labels. Reliability level of a VM from class (i) is higher than a VM from class $(i+1)$. Table 1 contains characteristics of each user type. $d_{user}(WTP_j, Penalty_j, R_i)$ and $d_{VM}(R)$ are as (22) and (23) and total number of users are 500 (100 users from each type). Resource releasing policy (VM selection to release its resources) is either user based (our proposed approach) or random. For simulations we assume that after a particular iteration, which is labeled by iteration 0, for 30 percent of requests there are not required resources and provisioning of new resources is impossible. This is while the incoming request rate doesn't change. All the reported results are gathered after iteration 0 at price $= 50$:

$$d_{user}(WTP_j, M_{LT}(j), R_i) = WTP_j + M_{LT}(j) \times (1 - R_i), \quad (22)$$

$$d_{VM}(R) = R. \quad (23)$$

Simulation 1. Figs. 6a, 6b, 6c, 6d, and 6e show average of the penalties that belong to the requests which are assigned to virtual machines from different classes during hundered iterations by the proposed (user based) approach. Comparing these results, we can see that having same WTP, requests with higher penalties are assigned to VMs with higher reliability. While Fig. 6f illustrates that using random approach for releasing resources, average of penalties for

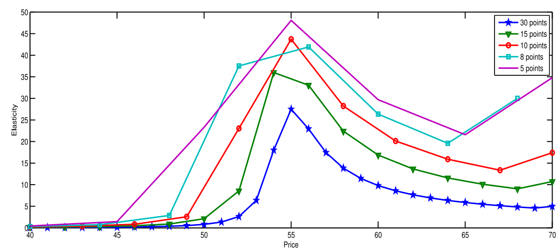


Fig. 5. Estimated WTP using different number of (price, request_rate) points.

TABLE 1
Simulation Parameters

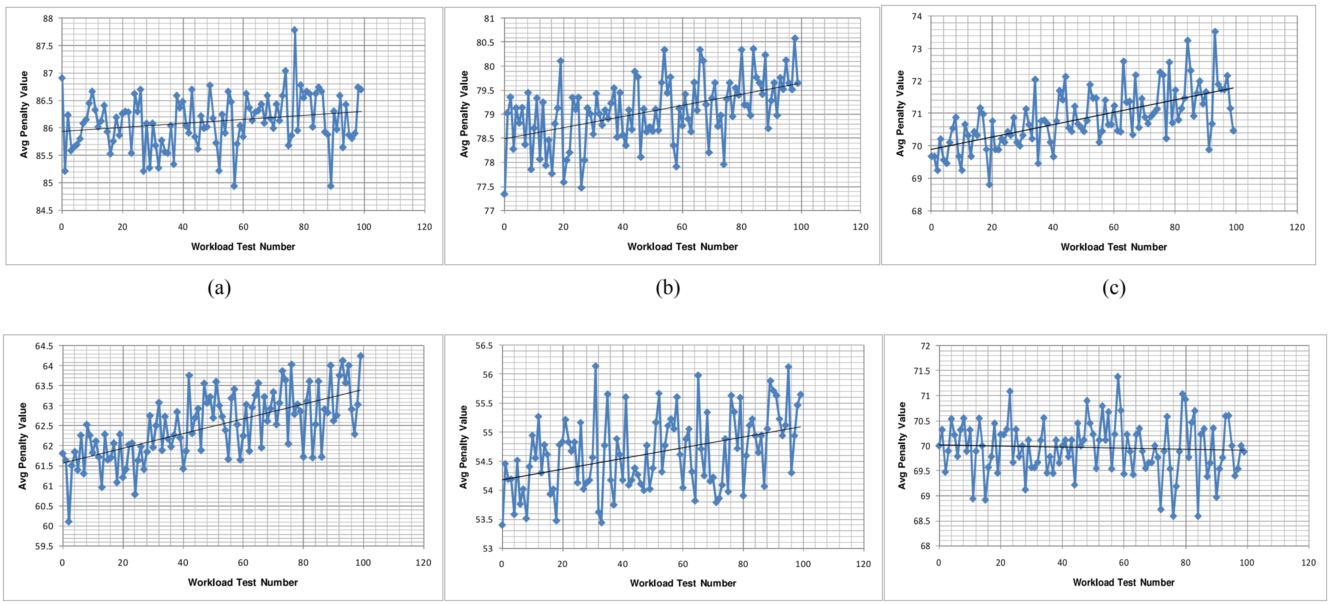
	Simulation 1			Simulation 2			Simulation 3		
	WTP	Risk Aversion	Violation Impact	WTP	Risk Aversion	Violation Impact	WTP	Risk Aversion	Violation Impact
User type 1	55	0.5	0.005	52	0.8	0.008	52	0.5	0.005
User type 2	55	0.6	0.006	54	0.8	0.008	54	0.6	0.006
User type 3	55	0.7	0.007	56	0.8	0.008	56	0.7	0.007
User type 4	55	0.8	0.008	58	0.8	0.008	58	0.8	0.008
User type 5	55	0.9	0.009	60	0.8	0.008	60	0.9	0.009

each class of virtual machines is approximately equal. Fig. 8a shows the total penalties paid by service provider to users for different user types. Clearly user based approach tried to satisfy more risk averse users and pay lesser amount of penalty. While this is not the case for random approach and total of paid penalties in random approach is greater than user based approach. Fig. 8b demonstrates number of SLA violations for each user type. If $(i > j)$, user type i has larger penalty value than user type j and discarding its request has higher cost for service provider. Performance of the user based approach according to Figs. 8a and 8b seems to be appropriate because more risk averse users have faced less SLA violations.

Simulation 2. This simulation contains user types with different WTP but similar risk aversion. Figs. 7a, 7b, 7c, 7d, and 7e show the average WTP of the users which their requests are assigned to virtual machines from different classes using the user based approach. Comparing the results, we can see the requests from users with higher WTP are assigned to virtual machines with higher reliability. This helps service provider to satisfy the users with higher WTP and improve its profitability. Fig. 7f illustrates that using random approach, average of WTP for each class of virtual machines is approximately equal. Fig. 8c depicts the total value of penalties paid by service provider to different user types. For $i > j$, user type i has higher WTP than user type j and discarding request of user type j is better option. Like as simulation 1, Fig. 8d demonstrates number of SLA violations for each user type. If $(i > j)$ discarding request of user type i decreases loyalty of users with higher WTP and this influence on profitability of service provider in long term.

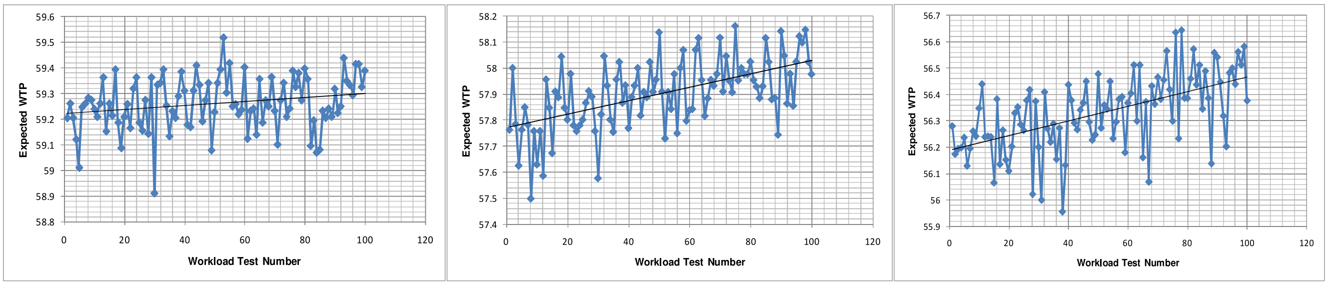
Increasing averages in Figs. 6a, 6b, 6c, 6d, and 6e, and Figs. 7a, 7b, 7c, 7d, and 7e is because of violation impact factor. User based approach focus on users with high WTP and risk aversion. As a result, users with lower WTP and risk aversion face more SLA violations and leave this service provider earlier than what occurs in random approach. Therefore average of WTP and penalties increases with iteration number.

Simulation 3. To be more natural, simulation 3 contains users with different WTP and risk aversion. As we said before, when a user faces SLA violation, this negatively influences on his decision about reusing the services of a service provider anymore. To model this influence we uses violation impact factor. Value of this factor is appropriate to user's risk aversion as illustrated by Table 1. Since SLA violations cause the service provider loses its users in long term, this will decrease profit of the service provider (income minus paid penalties). Fig. 9a shows the number of users who are dissuaded to reuse a service provider's services when they have faced SLA violations for many times. For example, approximately after 350 iterations, all the users with type 5, have

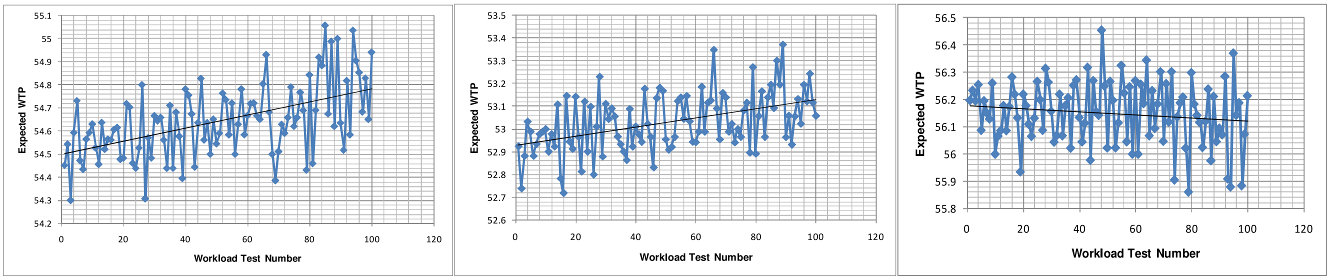


(a) (b) (c) (d) (e) (f)

Fig. 6. Average of the penalties that belong to the requests which are assigned to virtual machines from (a) class 1 (b) class 2 (c) class 3 (d) class 4 (e) class 5 during hundered iterations using user based approach. (f) Same diagram using random approach.



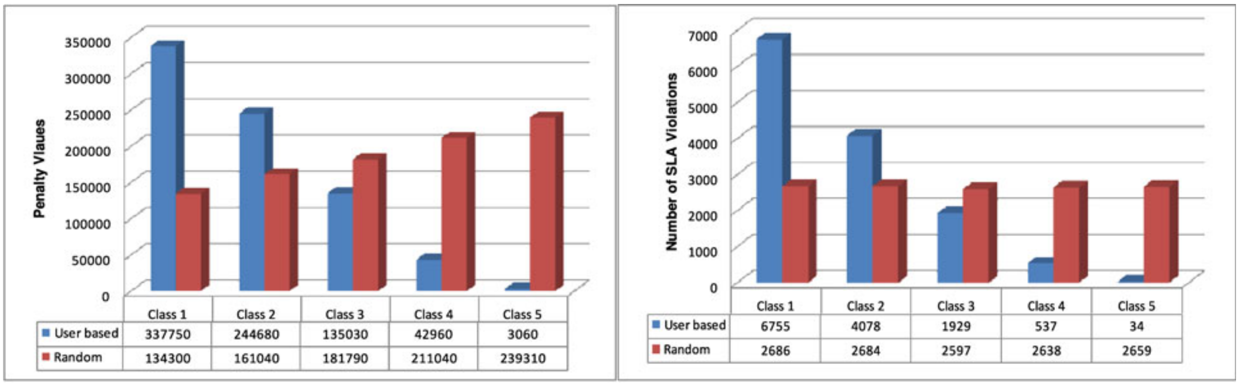
(a) (b) (c)



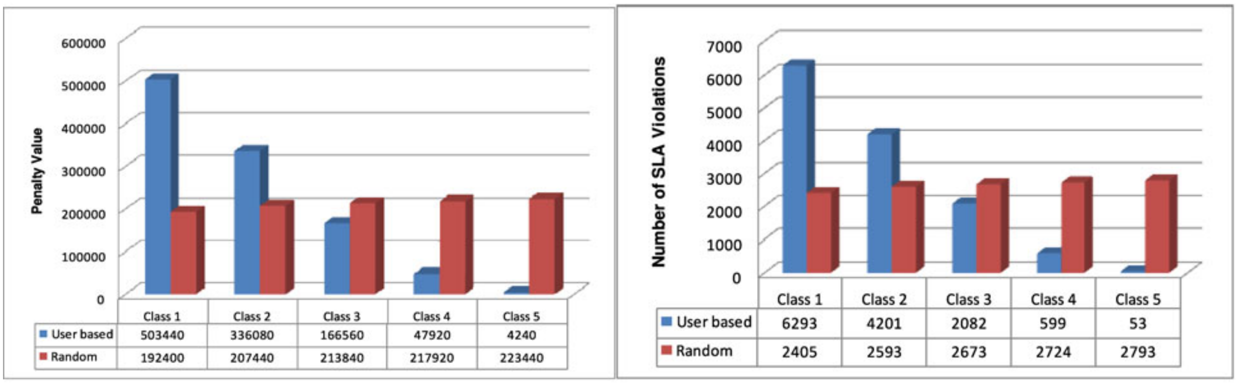
(d) (e) (f)

Fig. 7. Average WTP of the users which their requests are assigned to virtual machines from (a) class 1 (b) class 2 (c) class 3 (d) class 4 (e) class 5 using the user based approach. (f) Same diagram using random approach.

leaved service provider and migrated to another one. Diagrams of Fig. 9a are obtained by applying random approach. Similar diagrams using user based approach are illustrated in Fig. 9b. The diagrams of Fig. 9b states that the user based approach succeeded to achieve higher satisfaction level of users with higher WTP and risk aversion. Although this approach fails to improve satisfaction level of users with low WTP and risk aversion. Figs. 9c, and 9d show the same



(a) (b)



(c) (d)

Fig. 8. The paid penalties to each class of user types in (a) Simulation1 (c) Simulation2. Number of SLA violations for each class of user types in (b) Simulation1 (d) Simulation2.

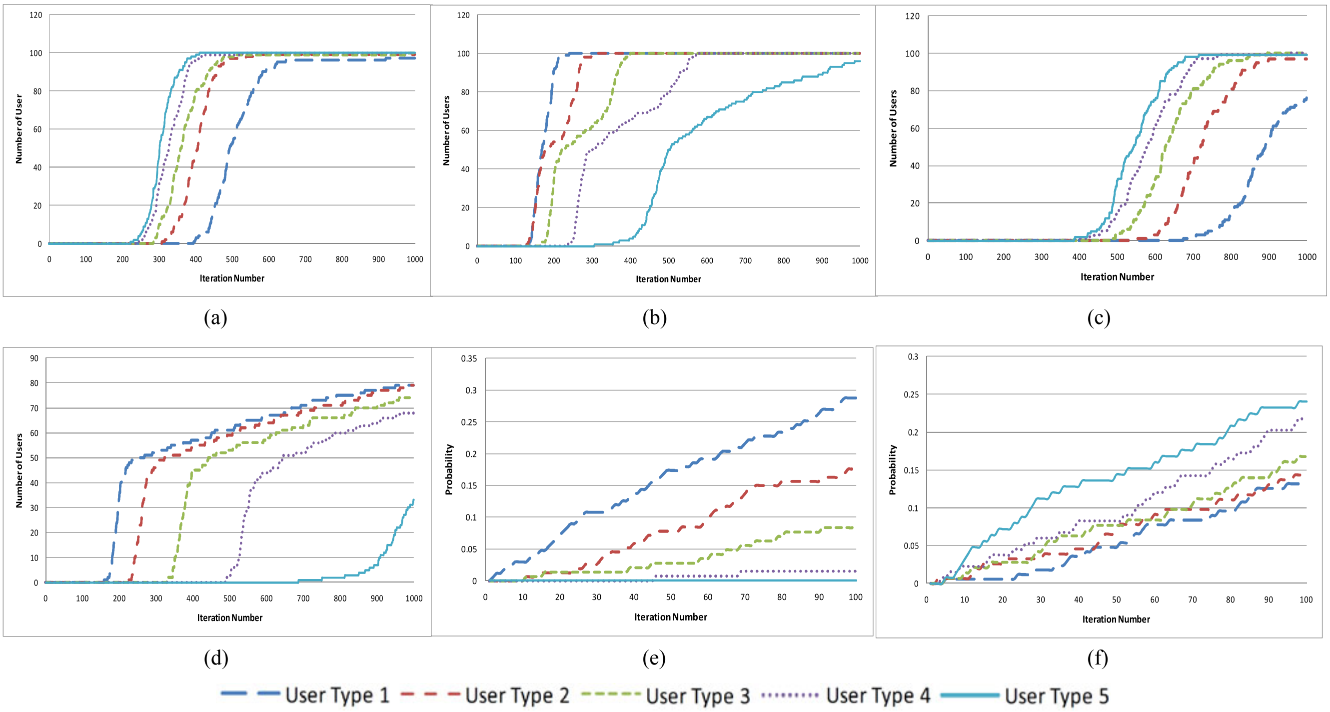


Fig. 9. Number of users who are dissuaded to reuse a service when f percent of the required resources are unavailable (a) $f = 30$, random approach (b) $f = 30$, user based approach (c) $f = 10$, random approach (d) $f = 10$, user based approach. (e) Amount of decrease of reusing a service during 100 iterations when 30 percent of resources are unavailable by applying user based approach (f) Same diagrams to (e) by applying random approach.

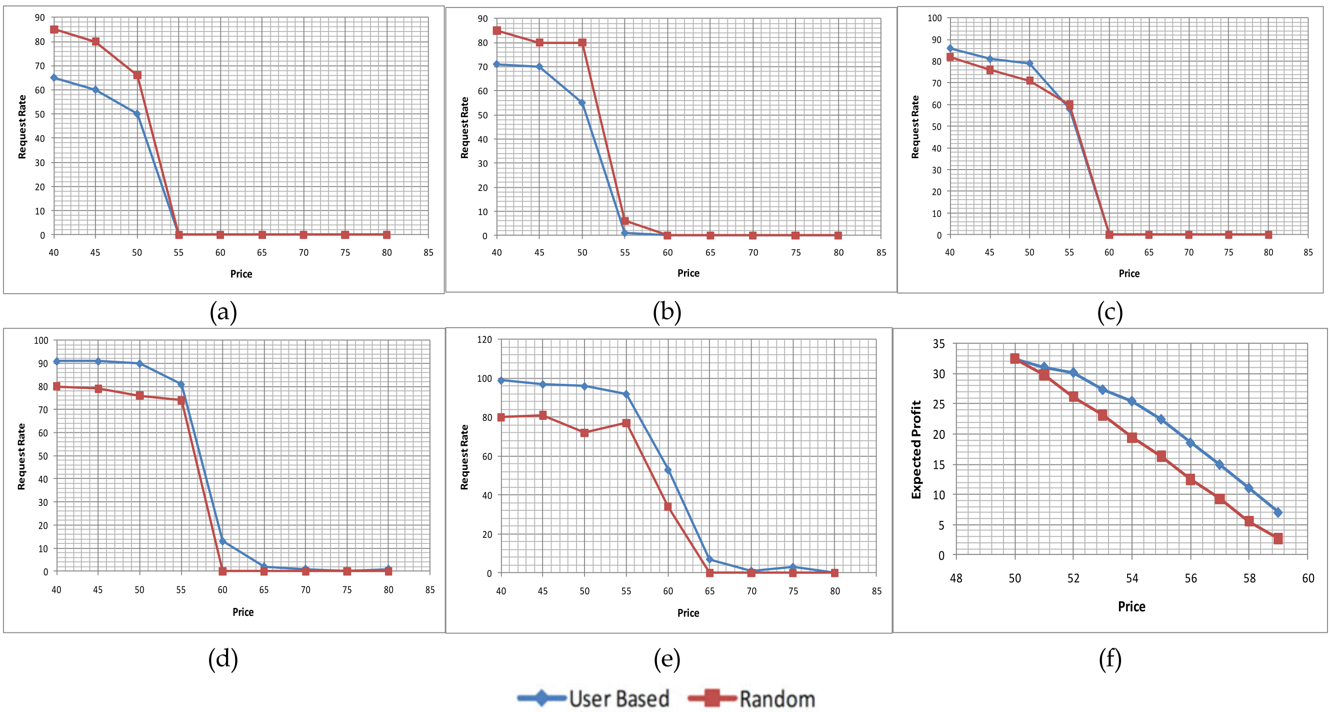


Fig. 10. Comparing the request rate of different user types in different prices by applying user based and random approaches (a) User Type 1 (b) User Type 2 (c) User Type 3 (d) User Type 4 (e) User Type 5. (f) Comparing profitability difference for various prices between user based approach and random.

diagrams while instead of 30 percent, just 10 percent of required resources to serve requests are unavailable.

As said before, when a user faces with SLA violation, probability of reusing that service decreases. Figs. 9e and 9f show the amount of this decrease during 100 iterations when for 30 percent of the requests there are not required resources. Diagrams of Fig. 9e belong to results of user based approach while those in Fig. 9f belong to random approach. Previous results shows that the proposed user based approach behave users based on their characteristics. More important users receive more attention from service provider. In addition to improving user satisfaction level, this increases profit of service provider. Figs. 10a, 10b, 10c, 10d, and 10e compare the request rate of different user types in different prices between user based and random approaches. The depicted demands or request rates are measured after 1,000 iterations. The figures explicitly state that in higher prices, the user based approach performs better and makes more profits.

Difference between profitability of user based and random approaches originates from this fact that user based approach behave users based on their profitability potential. Fig. 10f shows profitability difference (income minus the paid penalties) for various prices and this difference reach its maximum value, when price is close to expected willingness to pay of the users.

7 Conclusion

In this paper, we considered users satisfaction level as an important factor in profitability for cloud service providers. We tried to investigate influence of two characteristics in user satisfaction level. Since these characteristics, called willingness to pay for service and willingness to pay for certainty, are unknown for service providers, so new methods for estimation of them are provided. Also a new approach presented to reduce impact of SLA violations on users' satisfaction level. We investigate the mentioned approach in a resource allocation scenario. The conducted experiments demonstrate that in critical situations, the estimated characteristics can help the service provider to decide about which users should be served and which ones can be discarded. This can raise user satisfaction level as much as possible and leads to more loyalty of users and higher profit for service provider. According to results the proposed approach has high applicability in service oriented environments like as cloud.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions which improved this paper.

References

[1] A.J. Torkestani and M.R. Meybodi, "Finding Minimum Weight Connected Dominating Set in Stochastic Graph Based on Learning Automata," *Information Sciences*, vol. 200, pp. 57-77, 2012.
[2] G. Allon and A. Federgruen, "Competition in Service Industries," *Operations Research*, vol. 55, no. 1, pp. 37-55, 2007.
[3] H. Beigy and M. R. Meybodi, "An Adaptive Call Admission Algorithm for Cellular Networks," *Computers & Electrical Eng.*, vol. 31, no. 2, pp. 132-151, 2005.
[4] H. Beigy and M.R. Meybodi, "A Mathematical Framework for Cellular Learning Automata," *Advances in Complex Systems*, vol. 7, no. 03n04, pp. 295-319, 2004.
[5] H. Beigy and M.R. Meybodi, "Asynchronous Cellular Learning Automata," *Automatica*, vol. 44, no. 5, pp. 1350-1357, 2008.
[6] H. Beigy and M.R. Meybodi, "Cellular Learning Automata with Multiple Learning Automata in Each Cell and its Applications," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 1, pp. 54-65, Feb. 2010.
[7] R.N. Bolton, "A Dynamic Model of the Duration of the Customer's Relationship with a Continuous Service Provider: the Role of Satisfaction," *Marketing Science*, vol. 17, no. 1, pp. 45-65, 1998.
[8] R. Buyya, C.S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities," *Proc. 10th IEEE Int'l Conf. High Performance Computing and Comm. (HPCC'08)*, pp. 5-13, 2008.
[9] R. Buyya, S.K. Garg, and R.N. Calheiros, "SLA-Oriented Resource Provisioning for Cloud Computing: Challenges, Architecture, and Solutions," *Proc. Int'l IEEE Conf. Cloud and Service Computing (CSC)*, pp. 1-10, 2011.
[10] V. Cardellini, E. Casalicchio, Presti, F. Lo, and L. Silvestri, "SLA-Aware Resource Management for Application Service Providers in the Cloud," *Proc. First Int'l IEEE Symp. Network Cloud Computing and Applications (NCCA)*, pp. 20-27, 2011.
[11] A. Cuomo, G. Di Modica, S. Distefano, A. Puliafito, M. Rak, O. Tomarchio, S. Venticinque, and U. Villano, "An SLA-Based Broker for Cloud Infrastructures," *J. Grid Computing*, vol. 11, pp. 1-25, 2013.
[12] D.G. Modica, O. Tomarchio, and L. Vita, "Dynamic SLAs Management in Service Oriented Environments," *J. Systems and Software*, vol. 82, no. 5, pp. 759-771, 2009.
[13] M. Esnaashari and M.R. Meybodi, "Data Aggregation in Sensor Networks Using Learning Automata," *Wireless Networks*, vol. 16, no. 3, pp. 687-699, 2010.
[14] M. Esnaashari and M.R. Meybodi, "Dynamic Point Coverage Problem in Wireless Sensor Networks: A Cellular Learning Automata Approach," *Ad Hoc & Sensor Wireless Networks*, vol. 10, nos. 2/3, pp. 193-234, 2010.
[15] A. Gafni, "Willingness to Pay," *Pharmacoeconomics*, vol. 14.5, pp. 465-470, 1998.
[16] S.K. Garg, S.K. Gopalaiyengar, and R. Buyya, "SLA-Based Resource Provisioning for Heterogeneous Workloads in a Virtualized Cloud Datacenter," *Algorithms and Architectures for Parallel Processing*, pp. 371-384, Springer, 2011.
[17] I. Gilboa, *Theory of Decision Under Uncertainty*, vol. 1, Cambridge Univ. Press, 2009.
[18] A.B. Hashemi and M.R. Meybodi, "A Note on the Learning Automata Based Algorithms for Adaptive Parameter Selection in PSO," *Applied Soft Computing*, vol. 11, no. 1, pp. 689-705, 2011.
[19] S.M. Keaveney, "Customer Switching Behavior in Service Industries: An Exploratory Study," *The J. Marketing*, vol. 59, pp. 71-82, 1995.
[20] S.Y. Lam, V. Shankar, M.K. Erramilli, and B. Murthy, "Customer Value, Satisfaction, Loyalty, and Switching Costs: an Illustration from a Business-to-Business Service Context," *J. Academy of Marketing Science*, vol. 32, no. 3, pp. 293-311, 2004.
[21] Z. Landsman and M. Sherris, "Risk Measures and Insurance Premium Principles," *Insurance: Math. and Economics*, vol. 29, no. 1, pp. 103-115, 2001.
[22] Y.C. Lee, C. Wang, A.Y. Zomaya, and B.B. Zhou, "Profit-Driven Service Request Scheduling in Clouds," *Proc. 10th IEEE /ACM Int'l Conf. Cluster, Cloud and Grid Computing*, pp. 15-24, 2010.
[23] M. Mao, J. Li, and M. Humphrey, "Cloud Auto-Scaling with Deadline and Budget Constraints," *Proc. 11th IEEE/ACM Int'l Conf. Grid Computing (GRID)*, pp. 41-48, 2010.
[24] V. Mittal and W.A. Kamakura, "Satisfaction, Repurchase Intent, and Repurchase Behavior: Investigating the Moderating Effect of Customer Characteristics," *J. Marketing Research*, vol. 38, pp. 131-142, 2001.
[25] K.S. Narendra and M.A. Thathachar, "Learning Automata-A Survey," *IEEE Trans. Systems, Man and Cybernetics*, vol. SMC-4, no. 4, pp. 323-334, July 1974.
[26] K.S. Narendra and M. Thathachar, *Learning Automata: An Introduction*. Prentice-Hall, 1989.
[27] N. Nisan, *Algorithmic Game Theory*. Cambridge Univ. Press, 2007.
[28] N. Nisan, M. Schapira, G. Valiant, and A. Zohar, "Best-Response Mechanisms," *Proc. Conf. Innovations in Computer Science (ICS)*, pp. 155-165, 2011.
[29] G.M. Siddesh and K.G. Srinivasa, "SLA-Driven Dynamic Resource Allocation on Clouds," *Proc. Int'l Conf. Advanced Computing, Networking and Security*, pp. 9-18, 2012.
[30] Y. Song, H. Wang, Y. Li, B. Feng, and Y. Sun, "Multi-Tiered on-Demand Resource Scheduling for VM-Based Data Center," *Proc. Ninth IEEE/ACM Int'l Symp. Cluster Computing and the Grid*, pp. 148-155, 2009.

[31] D. Villegas, A. Antoniou, S.M. Sadjadi, and A. Iosup, "An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds," *Proc. 12th IEEE/ACM Int'l Symp. Cluster, Cloud and Grid Computing (CCGrid)*, pp. 612-619, 2012.
[32] L. Wu, S.K. Garg, and R. Buyya, "SLA-Based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments," *Proc. 11th IEEE/ACM Int'l Symp. Cluster, Cloud and Grid Computing (CCGrid)*, pp. 195-204, 2011.
[33] J. Xu, M. Zhao, J. Fortes, R. Carpenter, and M. Yousif, "On the Use of Fuzzy Modeling in Virtualized Data Center Management," *Proc. Fourth Int'l Conf. Autonomic Computing (ICAC'07)*, pp. 25-25, 2007.
[34] D. Yazdani, B. Nasiri, A. Sepas-Moghadam, and M.R. Meybodi, "A Novel Multi-Swarm Algorithm for Optimization in Dynamic Environments Based on Particle Swarm Optimization," *Applied Soft Computing*, vol. 13, no. 4, pp. 2144-2158, 2013.
[35] C.S. Yeo and R. Buyya, "Service Level Agreement Based Allocation of Cluster Resources: Handling Penalty to Enhance Utility," *Proc. IEEE Int'l Cluster Computing*, pp. 1-10, 2005.
[36] P. Zweifel and R. Eisen, "Insurance Economics," *Business & Economics*. Springer, 2012.

Hossein Morshedlou received the BSc and MSc degrees in computer engineering in Iran, in 2005 and 2008, respectively. He is currently working toward the PhD degree in computer engineering in AmirKabir University of Technology, Iran. His research interests include cloud computing, distributed systems, distributed artificial intelligence, learning automata, reinforcement learning, parallel algorithms, and soft computing.

Mohhamad Reza Meybodi received the BSc and MSc degrees in economics from Shahid Beheshti University in Iran, in 1973 and 1977, respectively. He also received the MSc and PhD degrees in computer science from Oklahoma University, in 1980 and 1983, respectively. Currently, he is a full professor in the Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran. Prior to current position, he was an assistant professor at Western Michigan University (1983 to 1985), as an associate professor at Ohio University, (1985 to 1991). His research interests include wireless networks, fault tolerant systems, learning systems, parallel algorithms, soft computing, and software development.