Online Cloud Provider Selection for QoS-Sensitive Users: Learning with Competition

Zhihao Wang, Junfang Wang, Bowen Li, Yijing Liu, Jinlong Ma

Abstract—**The expanding cloud computing services offer great opportunities for consumers to obtain better and cheaper service conveniently, which however raises new challenges on how to select the best service provider out of the huge pool. Although existing literatures have proposed several provider selection frameworks, none of them considered the performance unpredictability and dynamics caused by competition among cloud users. In this paper, we consider the online provider selection framework, where users dynamically and individually select their service providers based on experienced performance, and investigate the distributed decision-making strategy to achieve overall and individual performance guarantee. Specifically, we propose the learning-based selection policy, named Exp3.C, which regulates the system converging to a set of pure Nash equilibriums (PNE) of a congestion game in the homogeneous scenarios. Further, we show that even in a chaotic scenario where cloud users maybe irrational (which results in disordered and unpredictable behaviors) and the available resource of providers may change, the user's profit is guaranteed to approach that of selecting the best provider (which is derived with the assumption that all providers' status evolution are known) at the rate $O(\sqrt{T})$ in T rounds. Finally, numerical results validate the effectiveness of the proposed algorithm.**

Index Terms—**Cloud computing, Provider Selection, Online Learning, Congestion Game**

I. Introduction

MANY of today's Information Technology (IT) applications rely on access to state-of-the-art computing facilities. In response to the resulting demand for flexible computing resources, cloud computing has taken the IT industry by storm over the past few years [1]. Cloud computing emerges as a paradigm to deliver on-demand service (e.g., infrastructure, platform, software, etc.) to customers, much akin to electricity or cable television [2], [3]. The paradigm shift from IT as a product to IT as a service and the accompanying flexibility proliferate the cloud applications [4], [5]. According to [6], [7], the public cloud services market is expected to expand from $109 billion in 2012 to $207 billion by 2016.

With the growth of public cloud service offerings, for cloud customers it has become increasingly imperative to determine which service provider to use, for performance optimization as well as cost minimization [8], [9]. Commonly, cloud providers declare their available resource as well as corresponding price on their webs. Thus, an intuitive method for the cloud user is to select the provider according to the published information and its own demand. Most of the existing cloud marketplaces use service level agreements (SLAs) to express and negotiate user requirements and offers for services. Consequently, plenty of work has been paid on SLA-based provider selection [10], [11], [12], [13]. Unfortunately, although SLAs document QoS levels, the actual performance have not been found to be consistent among providers [14] and the provider-aroused QoS-violation (such as cloud outages) happens from time to time [15], [16]. This is partially due to the fact that the actual performance of any complex software system (such as a modern cloud platform or a cloud application) is intrinsically unpredictable [17]. The computer operating systems that drive cloud platforms today do not provide real-time guarantees; meanwhile, there is no foundational theory to guide us in building practical tools to predict and control the performance of programs. Moreover, the multi-tier interaction behavior in the cloud marketplace further adds the complexity of performance prediction. e.g., for a end user to select SaaS (software as a service) provider, a particular SaaS is likely running on top of another PaaS (platform as a service) or IaaS (infrastructure as a service). The interaction between the SaaS and PaaS/IaaS highly influence the end user's QoS performance [18], [19].

Therefore, an alternative approach for customers is to select its service provider by evaluating the real performance of cloud providers [14], [20], [21]. These solutions to some extent derive the actually perceived QoS for users with specific applications, by measuring the cloud with some indicating parameters, such as storage service response time, network latency, available bandwidth, job execution time, etc. However, they yield heavy measuring overhead in practical scenarios. Firstly, the real cloud performance is always dynamic and unpredictable due to the resource sharing, e.g., it is revealed by the latest measurements on Amazon EC2 that the standard medium instances experience up to 66% variation of the network I/O throughput [20], and the write I/O bandwidth of standard small instances can vary by as much as 50% from the mean [21]. Therefore, the user has to probe enough samples for deriving accurate performance statistic, which results in time-consuming measurements. Secondly, in real-world cloud marketplace, the heterogeneity of data centers (infrastructures of cloud providers) [22] and application traffics (workloads from cloud users) [23] becomes common. The off-line measurements on each type of workload-hardware pair in the huge cloud pool will undoubtedly bring noticeable processing overhead. Thirdly, in case that the cloud providers may upgrade their hardware and software infrastructure, and new providers may enter the market [14], periodical probing the huge cloud pool is required for achieving optimal decision-making, resulting in unacceptable cost for most of the cloud customers.

In this paper, we investigate the online provider selection framework, which helps cloud users to select proper service

Manuscript received August 01, 2015; revised April 26, 2016.

Zhihao Wang and Junfang Wang are with the 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China. E-mail: wangzhihao6416@163.com.

Bowen Li is with Xichang Satellite Launch Center.

Yijing Liu is with Beijing Institute of Electronic System Engineering.

Jinlong Ma is with Harbin Institute of Technology.

providers without dedicated probing overhead. Specifically, each user in the clouds individually selects service provider according to its own historical experience on the providers' QoS performance. The goal of our work is to devise efficient decision-making policy, guiding users to promote their long-term benefits. Note that similar online selection framework has been proposed for choosing low-risk provider for obtaining secure cloud service [24] and replica server for improving QoE in content distribution networks [25]. However, none of them considered the interactive user behaviors as well as the performance dynamics caused by competition among cloud users. While for the general cloud scenario, dynamic sharing of resource across customers and applications is common. As a consequence, the uncontrollable competition among users would inevitably decreasing the QoS performance [20]. What makes the problem worse is that, the quantitative analysis on the impact of competition over system performance is still an open problem in cloud system. In this work, we proposed an online learning algorithm, named **Exp3.C**, for guiding the user's decision-making. The main results are as follows.

- In homogeneous scenarios, the system would converge to a set of pure Nash equilibriums (PNE) of a congestion game, if all users apply the proposed strategy individually.
- In a chaotic scenario where the available resource of providers are non-stochastically changing, and/or the users are heterogeneous (or some users maybe irrational), the user's per-round profit approaches that of selecting the best single provider (obtained by a prophet) at the rate $O(\sqrt{T})$ in T rounds.

The remainder of this paper is organized as follows. Section II list the related work. Our considered system model and problem statement are given in Section III. In Section IV, we present the proposed learning policy and analyze the regret performance and convergence. Section V evaluate and analyze the effectiveness of learning policy by extensive simulations. Finally, Section VI concludes our work.

II. Related Work

Service-Level-Agreement (SLA) is a well-recognized concept in describing the quality of cloud service, and thus is widely studied in cloud provider selection. SLA ontologies [10], [11], [12], [13] try to facilitate the user-provider matching process by introducing ontologies as enhancement to plain SLA documents. The market participants are required to specify the semantics of their requirements as well as service offerings in such ontologies, which is a costly procedure. Unlike ontologies, the SLA mapping technique [26], [27], [28] are focusing on bridging the differences between two SLAs that with differing syntax but the same semantic. These studies greatly facilitate users to choosing a proper cloud provider automatically. However, as stated before, the inherent limitations of SLA restricts its performance in QoS-sensitive applications. The reasons are twofold. On one hand, some factors that really influence the user-perceived QoS are uncontrolled by the cloud provider, e.g., the wide area network bandwidth and delay, and thus can hardly be specified by existing SLA description framework. On the other hand, with the dynamic resource sharing nature as well as the intrinsic complexity of cloud system, it is highly intractable for the cloud operator to make accurate performance prediction and hard QoS provision to its diverse users [29]. As a result, for the cloud users it is impractical to achieve optimal decisions by just reading the providers published SLA specifications.

In achieving automatic cloud provider selection, several literatures have paid attention on the issue of system development. A semi-automated, extensible, and simplified system for infrastructure services selection, called CloudRecommender is proposed in [30], [31]. The core idea in CloudRecommender is to formally capture the domain knowledge of services using a declarative logic-based language, and then implement it in a recommender service on top of a relational data model. The author in [32] presented a system architecture that helps deciding the cloud providers based on the given requirements, and manages the desired resources by automatically creating new virtual machines from available providers. In contrast to these work focusing on realize automatic provider selection system, we try to devise intelligent decision-making framework with the goal of optimizing the cloud users' profit. So, these studies are complementary to our work.

Meanwhile, several studies have focused on evaluation of the real performance of cloud providers by off-line measurements, so as to help customers pick a cloud that fits their needs. CloudCmp [14] built a cloud measuring tool for helping end user to evaluate the QoS performance of customer's application running on a particular cloud. The elastic computing, persistent storage, and networking services offered by a cloud along metrics are measured for investigating their impact on the performance of different applications. The Cloud Service Measurement Index Consortium (CSMIC) [33] proposed a framework based on common characteristics of Cloud services. The aim of this consortium is to define each of the QoS attributes given in the framework and provide a methodology for computing a relative index for comparing different Cloud services. Later, SMICloud [34] is developed to systematically measure all the QoS attributes proposed by CSMIC and rank the Cloud services based on these attributes. The Analytical Hierarchical Process (AHP) is used to evaluate the cloud services based on the diverse QoS requirements of different applications. Although with great advantages in QoS provision, these off-line solutions face two issues in practical implementation. On one hand, it is impracticable to request all cloud providers to open their APIs for public query, thus the accurate QoS performance of a cloud could only be obtained by the user trying corresponding service. On the other hand, the vast cloud marketplace with diverse providers as well as the dynamic nature of cloud, requires sufficient and periodical measurements to be aware of providers' status, which results in huge investigating cost.

Most recently, online selection scheme has been introduced in service selection for secure cloud computing [24] and content distribution networks [25]. Focusing on the reliability of cloud service, SelCSP [24] dynamically selects service provider according to estimated risk of interaction. Both trustworthiness and competence are considered in choosing cloud provider, which are derived from personal experiences gained through feedbacks. In [25], a QoE-based replica server selection algorithm in the context of a content distribution network architecture is proposed. Online learning framework is introduced for improving the user's QoE by adaptively regulating server selection according to the user's past experience. Unlike the considered online learning problem in [24] and [25], where the user's object (or reward function) is independent to other users' actions. The reward of a user's action in our problem is really influenced by other user's behaviors, since the QoS of a cloud customer using a provider relates to the number of users associated to that provider. As a result, learning with competition framework is introduced in this work, assisting decision-making with the performance uncertainty caused by competition behavior among users as well as dynamic nature of cloud resource.

III. System Model and Problem Formulation

Consider there are $M \geq 1$ cloud customers (termed "users"), denoted by set $\mathbf{M} = \{1, 2, \ldots, M\}$, and $N > 1$ cloud service providers, denoted by set $\mathbf{N} = \{1, 2, \ldots, N\}$. Each user, says i, has a candidate provider set $\mathbf{N}_i \subseteq \mathbf{N}$. The candidate set $\mathbf{N}_i$ is composed of providers that fulfill the basic requirements of user i. At each round, the user selects exactly one provider for obtaining cloud service. Each user could dynamically change its provider for benefiting its own profit from round to round. The cloud service prices are given by $\{c_1, c_2, \ldots, c_N\}$ and is known to all users. We consider that the user knows the exact number of its candidate providers, i.e., $\mathbf{N}_i$ (This is consistent with real scenarios, as the user could query such information via web); however, the total number of users present in the system (i.e., M) and the candidate provider set of others are unknown to users. This model is richer and more realistic than previously used models [17], [18] that assume the number of users is known or $\mathbf{N}_i = \mathbf{N}$ for all $i \in \mathbf{M}$.

Let $r_j(t) \in [0, 1]$ be the total achievable service of provider j at time t, which relies on multiple factors, such as the dynamic network condition between the service provider and its backup PaaS/IaaS server, the available computing/storage resource provider attained from its PaaS/IaaS, etc. Without loss of generality, we consider that $r_j(t)_{t=1,2,\ldots}$ is generated by an i.i.d process with support $[0, 1]$ and mean $\mu_j \in [0, 1]$, for characterizing the performance uncertainty of provider j. Note that we assume that all rewards belong to the $[0, 1]$ interval here, however, the generalization of our results to rewards in $[a, b]$ for arbitrary $a < b$ is straightforward. Heterogeneous system is considered, i.e., the N providers are commonly with diverse value of μ; moreover, the statistic information is unknown to users.

Let $g_j \in [0, 1]$ be the QoS reward discount function (RDF) on provider j due to congestion or resource sharing, where $g_j(k)$ represents the reward discount factor when there are k users connecting provider j concurrently. RDF is assumed to be a decreasing function of k, however, the exact form is unknown to users. We hold the strict constrain for covering most realistic scenarios that, no direct communication is allowed among users; moreover, the user could observe the QoS reward (i.e., $r_j(t) g_j(k_j(t))$) after trying a particular service, but never know the exact value of $r_j(t)$ and $g_j(k_j(t))$.

Let $\mathbf{S}_i = \mathbf{N}_i$ be the set of feasible actions of user i and $\sigma_i \in S_i$ be the specific action, i.e., provider selected by user i. Let $\mathbf{S} = \mathbf{S}_1 \times \mathbf{S}_2 \times \ldots \times \mathbf{S}_M$ be the set of feasible action profiles and $\sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_M\} \in \mathbf{S}$ be the action profile

Algorithm Exp3.C

Parameter: a non-increasing sequence of real numbers η_t.

Initialize: set $\hat{U}_{ij}(0) = 0$ and $p_{ij}(1) = \frac{1}{N}$ for $j = 1, 2, \ldots, N_i$.

for t=1,2,. . . ,T **do**
  Select provider $\sigma_i(t)$ randomly according to the probability distribution $p_i(t) = [p_{i1}(t), p_{i2}(t), \ldots, p_{iN_i}(t)]$.
  Receive service from $\sigma_i(t)$ and attain reward $u_{\sigma_i}(t)$.
  for $j = 1, 2, \ldots, N_i$ **do**
    if $j = \sigma_i(t)$ **then**
      Set $\hat{U}_{ij}(t) = \hat{U}_{ij}(t-1) + \frac{u_{ij}(t)}{p_{ij}(t)}$.
    else
      Set $\hat{U}_{ij}(t) = \hat{U}_{ij}(t-1)$.
    end if
  end for
  Update probability distribution $p_i(t+1) = [p_{i1}(t+1), p_{i2}(t+1), \ldots, p_{iN_i}(t+1)]$, where

$$p_{ij}(t+1) = \frac{\exp\left(\eta_t \hat{U}_{ij}(t)\right)}{\sum_{j=1}^{N} \exp\left(\eta_t \hat{U}_{ij}(t)\right)} \tag{2}$$

end for

Fig. 1. Pseudo-code of **Exp3.C** in user i

of users. Suppose that a user selects provider j at time t. The profit it attained is then given by

$$u_j(t) = r_j(t) g_j\left(\sum_{m=1}^{M} I_{\{\sigma_m(t)=j\}}\right) - c_j \tag{1}$$

where $I_{\{.\}}$ is the indicator function. The objective of this work is to devise an online policy π_i for user i, mapping from its historical actions $\{\sigma_i(1), \sigma_i(1), \ldots, \sigma_i(t-1)\}$ and corresponding rewards $\{u_i(1), u_i(1), \ldots, u_i(t-1)\}$ to action $\sigma_i(t)$, for profit maximization.

IV. Online Policy for Cloud Provider Selection

In this section, we present and analyze the online learning algorithm **Exp3.C**. We will show that: 1) if all users follow our algorithm then the system converges to a set of pure Nash equilibria (PNE) of a congestion game; and 2) even in a chaotic system that users maybe irrational (which results in disordered and unpredictable behaviors) and/or the available resource of providers are non-stochastically changing, the user's profit is guaranteed——specifically, the user's per-round profit of **Exp3.C** approaches that of selecting the best single provider (by a prophet) at the rate $O(\sqrt{T})$ in T rounds.

The algorithm **Exp3.C**, described in Figure 1, is a variant of the algorithm **Exp3** [35], [36], which is used for solving exploration and exploitation trade-off in adversary environment. On each time step t, the user, says user i, draws an action $\sigma_i(t)$ according to a predefined probability vector $\{p_{i1}(t), p_{i2}(t), \ldots, p_{iN_i}(t)\}$, where each element in the vector indicates the probability of choosing a particular provider. The selection probability of a provider is initialized to be $\frac{1}{N}$, and is later updated round by round according to the derived cumulative reward of the provider. Specifically, the algorithm assigns to each action a probability mass exponential in the estimated cumulative reward for that action, as in Equ.(2). It is worth to note that for the drawn action $\sigma_i(t)$, **Exp3.C** sets the estimated reward $\hat{u}_{\sigma_i}(t)$ to $u_{\sigma_i}(t)/p_{i\sigma_i}(t)$. Such process, i.e., dividing the actual gain by the probability that the provider was chosen compensates the reward of cloud providers that are unlikely to be chosen. This choice of estimated rewards guarantees that their expectations are equal to the actual rewards for each action; that is, $E[\hat{u}_{ij}(t)|\sigma_i(1), \sigma_i(2), \ldots, \sigma_i(t-1)] = u_{ij}(t)$ where the expectation is taken with respect to the random choice of it at trial t given the choices $\{\sigma_i(1), \sigma_i(2), \ldots, \sigma_i(t-1)\}$ in the previous $t-1$ trials.

A. Congestion Game Formulation

Consider that all users are using **Exp3.C** independently for selecting service provider from its own candidate set. We explicitly note that users' behaviors are dependent, e.g., a user's action would affect other users' reward and updates, thus impact their decisions. In order to formulate such interaction behavior, we introduce the concept of congestion game [37], [38]. Congestion games are non-cooperative games in which the utility of each player depends only on the player's strategy and the number of other players that either choose the same strategy, or some strategies that "overlaps" with it. A congestion game has a potential function and the local maxima of the potential function corresponds to PNE, and every sequence of asynchronous improvement steps is finite and converges to PNE [39], [40].

Formally, a congestion game is given by the tuple $\left(\mathbf{M}, \mathbf{N}, (\mathbf{S}_i)_{i\in\mathbf{M}}, (u_j)_{j\in\mathbf{N}}\right)$, where $\mathbf{M}$ denotes a set of users, and $\mathbf{N}$ is a set of resources. $\mathbf{S}_i \subset 2^{\mathbf{N}}$ is the action space of user i, and u_j is a pay-off function associated with resource j, which is a function of the number of users competing for that resource. Recalling the user's reward function described in Equ.(1), it is straightforward to formulate our problem as congestion game. With this formulation, we have the following statement:

Theorem 1: From all but a measure 0 set of starting points, when $\eta_t = \eta$ is arbitrarily small, the solution of the **Exp3.C** converges to the set of pure Nash equilibrium of the congestion game $\left(\mathbf{M}, \mathbf{N}, (\mathbf{S}_i)_{i\in\mathbf{M}}, (u_j)_{j\in\mathbf{N}}\right)$.

Proof: Let $w_{ij}(t) = \exp\left(\eta \hat{U}_{ij}(t-1)\right)$ and define $u_{ij}(t) = 0$ for $j \neq \sigma_i(t)$. Then, the Equ.(2) can be rewritten as follows.

$$p_{ij}(t+1) = \frac{w_{ij}(t)\exp\left(\eta \frac{u_{ij}(t)}{p_{ij}(t)}\right)}{\sum_{j=1}^{N}\left\{w_{ij}(t)\exp\left(\eta \frac{u_{ij}(t)}{p_{ij}(t)}\right)\right\}}$$

Note that $p_{ij}(t) = \frac{w_{ij}(t)}{\sum_{j=1}^{N} w_{ij}(t)}$. The above equation can be further expressed as

$$p_{ij}(t+1) = \frac{p_{ij}(t)\left[\exp\left(\frac{\eta}{p_{ij}(t)}\right)\right]^{u_{ij}(t)}}{\sum_{j=1}^{N}\left\{p_{ij}(t)\left[\exp\left(\frac{\eta}{p_{ij}(t)}\right)\right]^{u_{ij}(t)}\right\}}$$

This indicates that our learning algorithm falls into the class so called *aggregate monotonic selection (AMS) dynamics* [41], [42], and the update equation is identical to that of [40, Equ.(1)]. As a result, the proof of converging to PNE follows from [40, Theorem 3.9, 4.1 and 4.4]. We recommend the readers refer to [40] for detailed and strict proof, and here briefly explain the steps in the proof only.

The deduction is based on analyzing a differential equation expressing a continuum limit of the multiplicative-weights update process, as the multiplicative factor approaches 1 and time is renormalized accordingly. The first step is to show that every flow line of the differential equation converges to the set of fixed points, by proving that the potential function associated with the congestion game is a Lyapunov function for any AMS dynamics. Then the stability analysis using the Jacobian matrix yields that every stable fixed point corresponds to a Nash equilibrium. Then one can prove that for any stable fixed point the eigenvalues of the Jacobian must be zero. This implies that every stable fixed point corresponds to a weakly stable Nash equilibrium strategy in the game theoretic sense. Then using techniques from algebraic geometry, one can prove that the existence of a non-pure weakly stable equilibrium implies the vanishing of a non-zero polynomial function of the edge costs, which implies that almost every weakly stable Nash equilibrium is a pure Nash equilibrium of the congestion game. We further need to investigate the error introduced by treating the discrete time update rule as a continuous time process. However, by taking the parameter η infinitesimal we can approximate the discrete time process by the continuous time process. For a discussion when η is not infinitesimal one can define approximately stable equilibrium. This concludes the proof of Theorem 1. ■

B. Worst Case Performance Analysis

As in previous section, we consider that the users are using **Exp3.C** and thus the user could learn the stochastic pay-off of different providers and predict other users' behaviors. However, in practical scenario there are much more uncertain factors such as other users may act under their own (irrational) policies (that we don't exactly know), the number of users in the system may change (some users may enter the cloud market and some others may exit), the volume of providers' resource may non-stochastically vary, etc. In respect with the environmental chaos, it is meaningless to analyze the system overall performance. In this section, we consider all the influencing factors as an adversary who decides the reward assignment among the providers in each round, and thus derive the worst-case performance bound.

We start by introducing the reward model with adversary and the concept of regret. At each round $t = 1, 2, \ldots$, simultaneously with the user's choice of the provider $\sigma_t \in \{1, 2, \ldots, N\}$, an adversary assigns to each $j = 1, \ldots, N$ the reward $u_j(t) \in [0, 1]$. For any reward assignment and for any $T > 0$, the user's accumulated reward is $\sum_{t=1}^{T} u_{\sigma_t}(t)$. We measure the performance of the user compared with the performance of the best single provider through the pseudo-regret defined as

$$\bar{R}(T) = \max_{j=1,\ldots,N} E\left[\sum_{t=1}^{T} u_j(t)\right] - E\left[\sum_{t=1}^{T} u_{\sigma_t}(t)\right] \quad (3)$$

Then, we have the following statement:

Theorem 2: 1) If **Exp3.C** is run with $\eta_t = \sqrt{\frac{\ln N}{tN}}$, then the pseudo-regret is upper bounded by

$$\bar{R}(T) \leq 2\sqrt{TN\ln N} \tag{4}$$

2) If the time horizon T is known, then the pseudo-regret bound can be further revised as

$$\bar{R}(T) \leq \sqrt{2TN\ln N} \tag{5}$$

by setting $\eta_t = \eta = \sqrt{\frac{2\ln N}{TN}}$.

Proof: For prove this theorem, we introduce the following lemma.

Lemma 1: For any non-increasing sequence $(\eta_t)_{t\in\mathbf{N}}$, **Exp3.C** satisfies

$$\bar{R}_T \leq \frac{N}{2}\sum_{t=1}^{T}\eta_t + \frac{\ln N}{\eta_T} \tag{6}$$

Proof: We give a skeleton proof for Lemma 1 here and refer the readers to [35], [36] for more details.

As the first step, we can verify that:

$$E_{j\sim p(t)}[\hat{u}_j(t)] = u_{\sigma_t}(t), \qquad E_{\sigma_t\sim p(t)}[\hat{u}_j(t)] = u_j(t) \tag{7}$$

$$E_{j\sim p(t)}[\hat{u}_j^2(t)] = \frac{u_{\sigma_t}^2(t)}{p_{\sigma_t}(t)}, \qquad E_{\sigma_t\sim p(t)}\frac{1}{p_{\sigma_t}(t)} = N \tag{8}$$

Then, they imply

$$\sum_{t=1}^{T}u_n(t) - \sum_{t=1}^{T}u_{\sigma_t}(t) = \sum_{t=1}^{T}E_{\sigma_t\sim p(t)}[\hat{u}_n(t)] - \sum_{t=1}^{T}E_{j\sim p(t)}[\hat{u}_j(t)] \tag{9}$$

Moreover, $E_{j\sim p(t)}[\hat{u}_j(t)]$ can be rewritten as

$$E_{j\sim p(t)}[\hat{u}_j(t)] = \frac{1}{\eta_t}\ln\left\{E_{j\sim p(t)}\left[\exp\left(\eta_t\left(\hat{u}_j(t)\right)\right)\right]\right\} - \frac{1}{\eta_t}\ln\left\{E_{j\sim p(t)}\left[\exp\left(\eta_t\left(\hat{u}_j(t) - E_{n\sim p(t)}[\hat{u}_n(t)]\right)\right)\right]\right\} \tag{10}$$

Further, we have the following equality for the first term of right hand in Equ.(10):

$$-\frac{1}{\eta_t}\ln\left\{E_{j\sim p(t)}\left[\exp\left(\eta_t\left(\hat{u}_j(t) - E_{n\sim p(t)}[\hat{u}_n(t)]\right)\right)\right]\right\} \leq \frac{\eta_t}{2p_{\sigma_t}(t)} \tag{11}$$

and for the second term:

$$\frac{1}{\eta_t}\ln\left\{E_{j\sim p(t)}\left[\exp\left(\eta_t\left(\hat{u}_j(t)\right)\right)\right]\right\} = \Phi_{t-1}(\eta_t) - \Phi_t(\eta_t) \tag{12}$$

where $\Phi_t(\eta) = \frac{1}{\eta}\ln\left\{\frac{1}{N}\sum_{j=1}^{N}\exp(\eta\hat{U}_j(t))\right\}$.

Putting Equ.(9)(10)(11) and (12) together, we have

$$\sum_{t=1}^{T}u_n(t) - \sum_{t=1}^{T}u_{\sigma_t}(t) \leq \sum_{t=1}^{T}\frac{\eta_t}{2p_{\sigma_t}(t)} + \sum_{t=1}^{T}\{\Phi_{t-1}(\eta_t) - \Phi_t(\eta_t)\} - \sum_{t=1}^{T}E_{\sigma_t\sim p(t)}[\hat{u}_n(t)]$$

Note that $E\left[\sum_{t=1}^{T}\frac{\eta_t}{2p_{\sigma_t}(t)}\right] = \frac{N}{2}\sum_{t=1}^{T}\eta_t$, and

$$\sum_{t=1}^{T}\{\Phi_{t-1}(\eta_t) - \Phi_t(\eta_t)\} = \sum_{t=1}^{T}\hat{u}_n(t) + \sum_{t=1}^{T-1}\{\Phi_t(\eta_{t+1}) - \Phi_t(\eta_t)\} + \frac{\ln N}{\eta_T}$$

We have

$$E\left[\sum_{t=1}^{T}u_n(t) - \sum_{t=1}^{T}u_{\sigma_t}(t)\right] \leq \frac{N}{2}\sum_{t=1}^{T}\eta_t + \frac{\ln N}{\eta_T} + E\left[\sum_{t=1}^{T-1}\{\Phi_t(\eta_{t+1}) - \Phi_t(\eta_t)\}\right]$$

With the condition that "η_t is a non-increasing sequence with respect to t", we further establish that $E\left[\sum_{t=1}^{T-1}\{\Phi_t(\eta_{t+1}) - \Phi_t(\eta_t)\}\right] \leq 0$, which finally concludes the proof of Lemma 1. ■

With the results of Lemma 1, the second term of Theorem 1 could be obtained directly by putting $\eta_t = \sqrt{\frac{2\ln N}{TN}}$ into Equ.(6).

For the first term, we have

$$\begin{aligned}\bar{R}_T &\leq \frac{N}{2}\sum_{t=1}^{T}\sqrt{\frac{\ln N}{tN}} + \sqrt{TN\ln N}\\ &\leq \frac{1}{2}\sqrt{N\ln N}\int_{t=0}^{T}\frac{1}{\sqrt{t}} + \sqrt{TN\ln N}\\ &= 2\sqrt{TN\ln N}\end{aligned}$$

which completes the proof of Theorem 2. ■

Theorem 2 shows that no matter how the cloud market changes, the user's per-round profit of **Exp3.C** approaches that of selecting the best single provider (by a prophet) at the rate $O(\sqrt{T})$ in T rounds, which greatly boosts the confidence of implementing the algorithm in real scenarios.

V. Numerical Results

In this section, we show the convergence as well as effectiveness of the proposed learning algorithm by numerical simulations.

Four cloud providers are considered offering computing service, whose computing capacity are normalized to be 1, 0.8, 0.6 and 0.4, respectively. The number of cloud users is set to be 20. The users' traffic needs are set to be ranging from 0.1 to 0.2. Specifically, user 1 has a normalized traffic of 0.1, and the traffic of user 20 is 0.2. The minimum traffic gap between two users is 0.005. Note that here we raised the normalized traffic needs of each cloud user, so as to simulate the competition behaviors. We consider a typical cloud resource-sharing scheme, in where all the users requirements should be satisfied if the cloud resource is sufficient; otherwise, the rewards of all the cloud customers would decrease proportionately to their request. By setting the parameter $\eta = 1$, we carried out the dynamic provider selection and utilization process.

To show the learning and convergence process, we depict the evolution of users and providers' reward in a particular run. From the perspective of cloud offering, the provider's resource utility ratio is critical. We depict the achieved

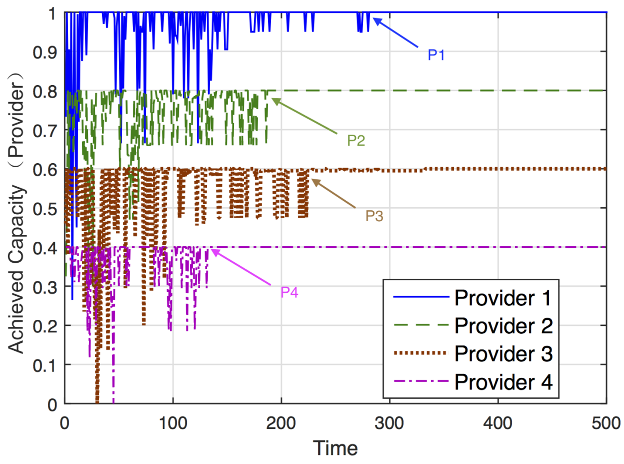

Fig. 2. The Variation of Cloud-provider-achieved Capacity

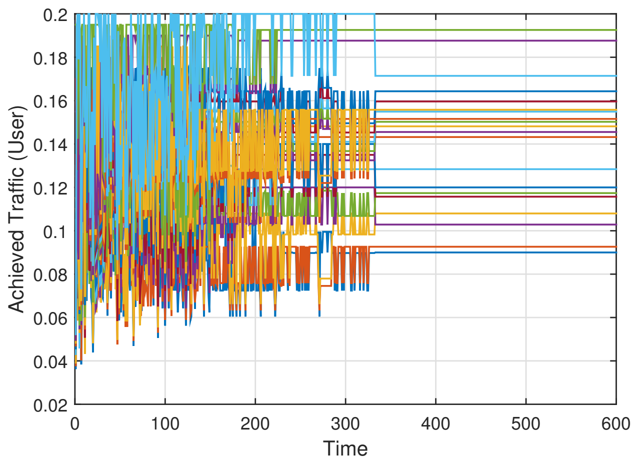

Fig. 3. The Variation of Cloud-user-derived Traffic

capacity of each service provider in Fig.2. The achieved capacity is defined as the actual served traffic load, which is differing from the provider's achievable capacity as well as the total traffic uploaded to the provider. Recall that the summation of users' normalized traffic needs is 3, which is higher than the providers' total computing capacity 2.8. Thus, all the providers could reach their capacity in ideal situation. As shown in the figure, at the beginning the conflicting choices among users boiled down the capacity. Even in such resource shortage case, considerable proportion of the cloud resource is under-utilized. The learning algorithm eliminates the resource waste by virtually coordinating users selections. We are inspired by the simulation result, which shows that all the four providers' capacity are reached after 300 rounds.

In respect of cloud consumers, the users' actual traffic served by cloud providers could be used as an indicator showing the learning process. In Fig.3, all the 20 users' actual traffic served by cloud providers are depicted as a function of time. Due to space limitation, the label of each curve is omitted in the figure. From the figure, we find that the system is highly dynamic due to the competition behavior at the first 200 rounds; and within about 320 rounds, all users converge to a stable status, which reveals system equilibrium.

We further show the learning process in Fig.4, by depicting the evolution of users' selection policy (i.e., the probability of selecting a particular provider). For space saving, we exhibit the first 6 users' policy evolution only in the figure. The label of "select P1" in the figure indicates "the probability of selecting the first cloud provider ", and the rest can be done in the same manner. The curves clearly reveal the evolution process and convergence of users' decisions: the selections are highly conflicting at the beginning, and then users dynamically adapt their selections according to their own observation, and finally the system achieves a coordinating status.

It is shown in Fig.4 that the convergence speeds of different users are diverse. For some users, typically represented by the sixth user in this figure, their selection policy converges soon as the learning and competition process goes on. While for some other users, their convergence speed is relatively long, e.g., the first user in the figure achieves stable policy with nearly 350 selections. Moreover, the convergence of learning process is actually uncertain due to the stochastic nature of environment as well as users' behaviors. Hence, the convergence speeds of different experimental runs are independent.

In order to reveal the statistical results of convergence rate, we conduct 600 independent runs, where each run lasts for 800 time slots. The convergence rate is represented by the learning time before a user arriving its stable policy. Here, a user achieves a stable policy is defined as the situation that its probability of choosing a particular provider is higher than 0.99. We count and record the times of all the users achieving stable state, in all the 600 runs. In addition to the overall distribution, two other related indicators are derived, i.e., the time of the first user and the last user achieving stable policy in each run. The former shows the fastest convergence time of users, while the latter could be considered as the system convergence time. The results are shown in Fig.5. It reveals that some of the users attain rapid convergence: in 99% of the 600 runs, the fastest converging user achieves stable within only 30 time rounds. For the general cases, over 60% of the overall users achieve stable in 100 time rounds, and nearly 90% users' policies converge in 250 steps. In respect of system convergence time, over 75% of the cases the system converges in 500 time steps.

Finally, the system totally achieved capacity is explored, where the intuitive randomized policy is introduced for comparison. Three kinds of scenarios are considered, i.e., the low traffic mode, medium traffic mode and high traffic mode. The users' traffic needs of the three modes are set to be ranging from 0.05 to 0.1 with step of 0.0025, ranging from 0.1 to 0.2 with step of 0.005, and ranging from 0.2 to 0.4 with step of 0.01, respectively. For each scenario, 600 independent runs are conducted. Each run lasts for 800 time slots.

The results are shown in Fig.6. In all the three scenarios, the effectiveness of learning algorithms is apparent. As time goes on, the system capacity with learning policies increases, widening the gap over the randomized policy. Typically, as shown in the medium traffic case, although the total capacity provided by the cloud providers is 2.8 (normalized value, the summation of 1, 0.8, 0.6 and 0.4), the intuitive policy only achieved 2.3, resulting in 18% capacity loss. While with the proposed learning algorithm, the system performance is

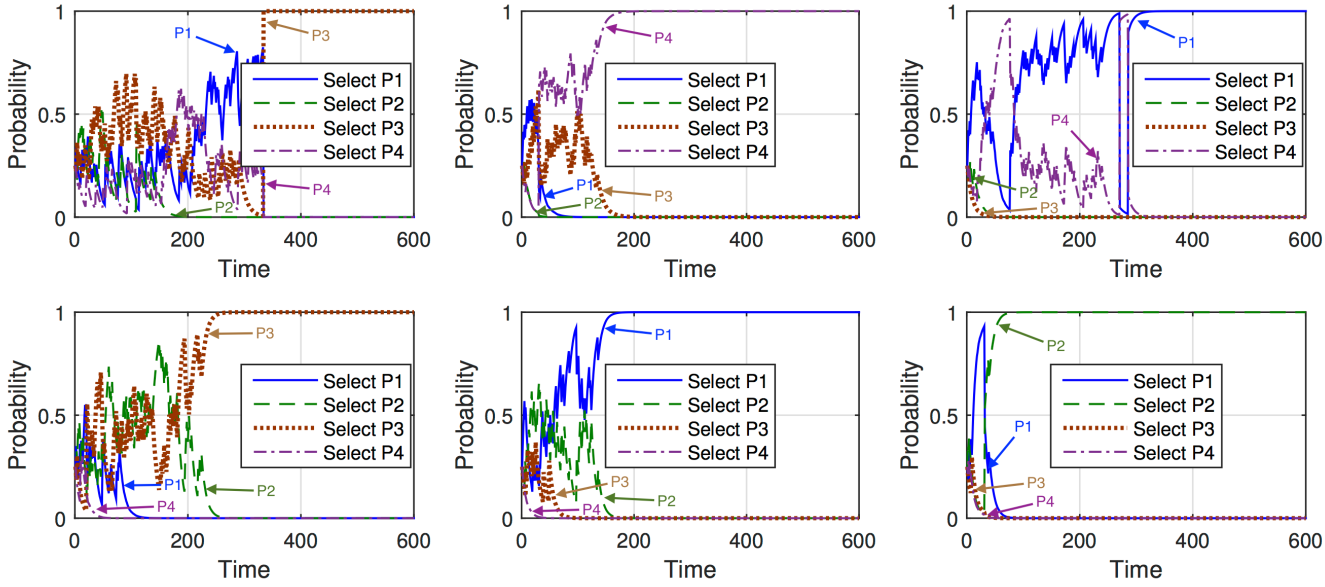

Fig. 4. Convergence of Users' Action

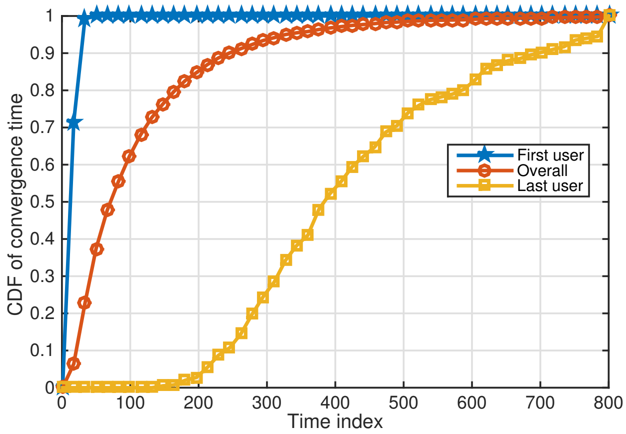

Fig. 5. Cumulative distribution function of users' convergence time

improved quickly and efficiently as time goes on, and finally converges to nearly full load of cloud capacity. Another observation we derived from this figure is that, the system converges faster in the scenario with higher traffic load. e.g., the number of time slots for achieving stable in low traffic case is nearly 700, and that in medium and high traffic are 300 and 100 respectively.

VI. Conclusion

Cloud computing has become an important paradigm for outsourcing various IT needs of organizations. Currently, there are many cloud providers who offer different cloud services with different price and performance attributes. With the growing number of cloud offerings, it has also becomes challenging for cloud customers to find the best cloud services which can satisfy their QoS requirements, especially when there are plenty of cloud users dynamically selecting the cloud service concurrently. In order to handle the unpredictability caused by both the randomness of cloud status and competition among users, we devised the learning-based dynamic provider selection framework and proposed

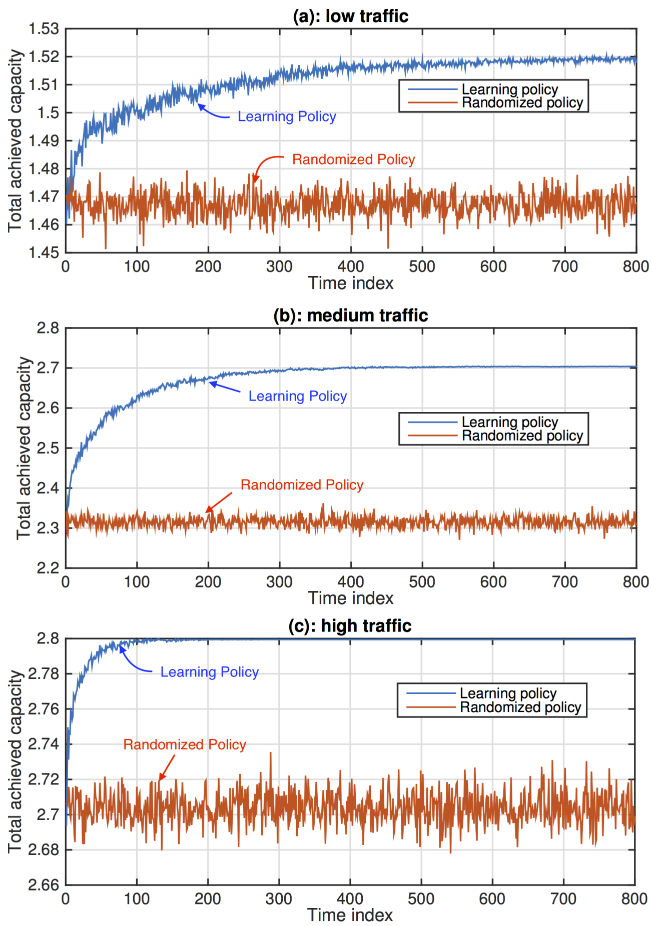

Fig. 6. The Total Cloud Capacity Variation

corresponding online algorithm. We proved the stability and advancement of the proposed learning algorithm, by appealing to congestion game formulation. The robustness of our algorithm is further shown by regret analysis in the chaotic environment.

REFERENCES

[1] R. Mahowald, C. Sullivan, and A. Konary, "Market analysis perspective: Worldwide saas and cloud services, 2012–new models for delivering software (2012)."

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[3] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, no. 1, pp. 7–18, 2010.

[4] Z. Chen, Y. Zhu, Y. Di, and S. Feng, "Optimized self-adaptive fault tolerance strategy in simulation system based on virtualization technology." *IAENG International Journal of Computer Science*, vol. 42, no. 4, pp. 305–312, 2015.

[5] X. Zhong, G. Yang, L. Li, and L. Zhong, "Clustering and correlation based collaborative filtering algorithm for cloud platform." *IAENG International Journal of Computer Science*, vol. 43, no. 1, pp. 108–114, 2016.

[6] C. Pettey and B. Tudor, "Gartner says worldwide cloud services market to surpass $109 billion in 2012," *Gartner Inc., Stamford, Press release*, 2012.

[7] "The future of cloud adoption," *See http://cloudtimes.org/2012/07/14/the-future-of-cloud-adoption*, 2012.

[8] S. Liu, F. T. Chan, and W. Ran, "Decision making for the selection of cloud vendor: An improved approach under group decision-making with integrated weights and objective/subjective attributes," *Expert Systems with Applications*, vol. 55, pp. 37–47, 2016.

[9] L. Qi, W. Dou, and J. Chen, "Weighted principal component analysis-based service selection method for multimedia services in cloud," *Computing*, vol. 98, no. 1-2, pp. 195–214, 2016.

[10] K. Kritikos and D. Plexousakis, "Requirements for QoS-based web service description and discovery," *Services Computing, IEEE Transactions on*, vol. 2, no. 4, pp. 320–337, 2009.

[11] V. X. Tran and H. Tsuji, "A survey and analysis on semantics in QoS for web services," in *Advanced Information Networking and Applications, 2009. AINA'09. International Conference on*. IEEE, 2009, pp. 379–385.

[12] E. Daehnhardt and Y. Jing, "An approach to software selection using semantic web," *IAENG International Journal of Computer Science*, vol. 40, no. 4, pp. 238–249, 2013.

[13] K. Fakhfakh, S. D. Tazi, K. Drira, T. Chaari, and M. Jmaiel, "Implementing and testing a semantic-driven approach towards a better comprehension between service consumers and providers," in *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*. IEEE, 2010, pp. 183–188.

[14] A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: comparing public cloud providers," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 1–14.

[15] S. M. Habib, S. Ries, M. Mühlhäuser, and P. Varikkattu, "Towards a trust management system for cloud computing marketplaces: using caiq as a trust information source," *Security and Communication Networks*, vol. 7, no. 11, pp. 2185–2200, 2014.

[16] M. Mohaupt and A. Hilbert, "Integration of information systems in cloud computing for establishing a long-term profitable customer portfolio," *IAENG International Journal of Computer Science*, vol. 40, no. 2, pp. 124–133, 2013.

[17] B. Jennings and R. Stadler, "Resource management in clouds: Survey and research challenges," *Journal of Network and Systems Management*, vol. 23, no. 3, pp. 567–619, 2015.

[18] J. Anselmi, D. Ardagna, and A. Wierman, "Congestion and price competition in the cloud," *Extended version: http://users. cms. caltech. edu/ adamw/publications-0. html*.

[19] E. R. L. Jalao, D. L. Shunk, and T. Wu, "Life cycle costs and the analytic network process for software-as-a-service migration," *IAENG International Journal of Computer Science*, vol. 39, no. 3, pp. 269–275, 2012.

[20] A. Shieh, S. Kandula, A. Greenberg, C. Kim, and B. Saha, "Sharing the data center network," in *Proceedings of the 8th USENIX conference on Networked systems design and implementation*. USENIX Association, 2011, pp. 309–322.

[21] S. K. Barker and P. Shenoy, "Empirical evaluation of latency-sensitive application performance in the cloud," in *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*. ACM, 2010, pp. 35–46.

[22] Z. Ou, H. Zhuang, J. K. Nurminen, A. Ylä-Jääski, and P. Hui, "Exploiting hardware heterogeneity within the same instance type of amazon ec2," in *Proceeding 4th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud'12), Boston, USA*, 2012, pp. 1–5.

[23] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proceedings of the Third ACM Symposium on Cloud Computing*. ACM, 2012, pp. 1–13.

[24] N. Ghosh, S. K. Ghosh, and S. K. Das, "SelCSP: A framework to facilitate selection of cloud service providers," *Cloud Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 66–79, 2015.

[25] H. A. Tran, S. Hoceini, A. Mellouk, J. Perez, and S. Zeadally, "QoE-based server selection for content distribution networks," *Computers, IEEE Transactions on*, vol. 63, no. 11, pp. 2803–2815, 2014.

[26] I. Brandic, D. Music, and S. Dustdar, "Service mediation and negotiation bootstrapping as first achievements towards self-adaptable cloud services," in *Grids and Service-Oriented Architectures for Service Level Agreements*. Springer, 2010, pp. 119–132.

[27] M. Maurer, V. C. Emeakaroha, I. Brandic, and J. Altmann, "Cost–benefit analysis of an SLA mapping approach for defining standardized cloud computing goods," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 39–47, 2012.

[28] C. Redl, I. Breskovic, I. Brandic, and S. Dustdar, "Automatic SLA matching and provider selection in grid and cloud computing markets," in *Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing*. IEEE Computer Society, 2012, pp. 85–94.

[29] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: a survey, state of the art, and future directions," *Proceedings of the IEEE*, vol. 102, no. 1, pp. 11–31, 2014.

[30] M. Zhang, R. Ranjan, S. Nepal, M. Menzel, and A. Haller, "A declarative recommender system for cloud infrastructure services selection." in *GECON*. Springer, 2012, pp. 102–113.

[31] M. Zhang, R. Ranjan, A. Haller, D. Georgakopoulos, and P. Strazdins, "Investigating decision support techniques for automating cloud service selection," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*. IEEE, 2012, pp. 759–764.

[32] J. Son, "Automated decision system for efficient resource selection and allocation in inter-clouds," Ph.D. dissertation, The University of Melbourne, 2013.

[33] S. K. Garg, S. Versteeg, and R. Buyya, "SMICloud: a framework for comparing and ranking cloud services," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. IEEE, 2011, pp. 210–218.

[34] ——, "A framework for ranking of cloud computing services," *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012–1023, 2013.

[35] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[36] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[37] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[38] R. W. Rosenthal, "A class of games possessing pure-strategy Nash equilibria," *International Journal of Game Theory*, vol. 2, no. 1, pp. 65–67, 1973.

[39] R. Kleinberg, G. Piliouras, and E. Tardos, "Multiplicative updates outperform generic no-regret learning in congestion games," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 2009, pp. 533–542.

[40] C. Tekin and M. Liu, "Performance and convergence of multi-user online learning," in *Game Theory for Networks*. Springer, 2011, pp. 321–336.

[41] K. Ritzberger and J. W. Weibull, "Evolutionary selection in normal-form games," *Econometrica: Journal of the Econometric Society*, pp. 1371–1399, 1995.

[42] L. Samuelson and J. Zhang, "Evolutionary stability in asymmetric games," *Journal of economic theory*, vol. 57, no. 2, pp. 363–391, 1992.