Contents lists available at ScienceDirect
International Journal of Approximate Reasoning
www.elsevier.com/locate/ijar

An online sequential procurement mechanism under uncertain demands in multi-cloud environment ☆

Jingti Han [a,b], Xiaohong Wu [a,c], Jian-Guo Liu [a,b,*]

[a] School of Information Management and Engineering, Shanghai Key Lab. of Financial Info. Tech., Shanghai University of Finance and Economics, Shanghai 200433, People's Republic of China
[b] Institute of financial science and technology, Shanghai University of Finance and Economics, Shanghai 200433, People's Republic of China
[c] School of Information Engineering, Huzhou University, Zhejiang, 313000, People's Republic of China

ARTICLE INFO

Article history:
Received 31 December 2017
Received in revised form 5 August 2018
Accepted 18 September 2018
Available online xxxx

Keywords:
Resource procurement
Cloud computing
Uncertain demands
Budget constraint

ABSTRACT

The uncertainty of demands brings challenges for the private cloud providers, leading to low utilization of resources during periods of low-demand and low quality of service during periods of peak-demand, which has attracted much attention. In this paper, taking account into both uncertainty of demands and budget constraint, we design an online sequential procurement auctions of residual resources, which helps the busy cloud provider make an irrevocable decision about how to purchase resources during period of uncertain peak-demand. The crucial part of the mechanism is the seller accepting-rule based on a value-density threshold which is learned dynamically from the historical information. Given the condition that all the sellers are myopic, we prove that the mechanism is truthful, budget feasible and individual rational. Furthermore, we obtain the competitive ratio of the proposed mechanism when the demands of the BCP are δ-degree balance. Using real data from parallel computing centers, we construct 60 scenarios in six data settings, in which we compare our mechanism with average budget allocation and offline proportional sharing mechanism, the results show that in more than 85% scenarios the proposed mechanism has better performance than allocation with average budget, and it improves more than 20% valuation on average for the buyer, even if we use the estimate value of balance degree δ.

1. Introduction

The uncertainty of demands is one of the most critical problems faced by private cloud providers, which results in low resource utilization during periods of low-demand and low service quality during periods of peak-demand. It is an unsolved problem in a single cloud environment, but in a multi-cloud environment, cloud providers can improve the resource utilization or service quality by trading and sharing of residual resources.

Economic mechanisms have been applied widely in trading and sharing of residual resources among cloud networks, in which cloud providers either win more economic benefits or enlarge their resource capacity by sharing the resources dynamically [1]. As described in [2], economic mechanism is the key ingredient for effective resource utilization in cloud

☆ This paper is part of the Virtual special issue on 1st International Workshop on Uncertainty in Cloud Computing - DEXA 2017, Edited by Allel Hadjali, Haithem MEZNI and Sabeur ARIDHI.

* Corresponding author.
E-mail addresses: hanjt@mail.shufe.edu.cn (J. Han), xhwu@zjhu.edu.cn (X. Wu), liujg004@ustc.edu.cn (J.-G. Liu).

https://doi.org/10.1016/j.ijar.2018.09.006

network which has received more attention [3–7]. Especially, due to data security and other reasons, many enterprises and universities would like to set up their private clouds instead of outsourcing their entire infrastructure to public clouds, which makes the trading and scheduling of resources among those private clouds more necessary [8].

Procurement mechanism is one of the major resource trading mechanisms, which has been applied to cloud networks [9–12]. In these works, the cloud providers (CPs) in the cloud network obtain the utility by sharing and scheduling the resources in the following ways: (i) when the cloud is in period of low-demand, the increased utility of the idle cloud provider (ICP) mainly comes from rented resources, i.e., its income from leasing VMs to other clouds subtracting its operational costs; (ii) when the CP is in period of peak-demand, the increased utility of the busy cloud provider (BCP) mainly comes from the improved service quality associated with the quantity of rented resources. For example, Hassan et al. [9] derived the procurement price based on Stackelberg Game for the busy cloud provider in peak-demand, in which the demand valuation function of resource buyer is concave. Prasad et al. [11] proposed three QoS-awared procurement mechanisms for the busy cloud provider, which adopted reverse auction to decide the allocation of sellers and the pricing.

The above-mentioned approaches only take into account static demands of the BCP, while resource buyers (BCPs) are always faced with more complex scenario – online uncertain resource demands in practice. For example, large scale promotion activities of a large online store, such as “double 11” in China, will attract a large number of customers to access the website, and during this period, it is necessary to increase the resource provisioning to release the pressure of the website. Since the uncertainty of workloads, the valuation of a resource for each provider varies with their workload status, and it represents the marginal price of the ICP or the marginal demand valuation of the BCP which is uncertain and unpredictable for each provider. In addition, similar to the usual purchase behavior, the resource expansion for a BCP is constrained by a limited budget. Obviously, the procurement mechanisms aforementioned cannot be applied to this scenario directly. Zhao et al. [13] investigate online procurement of storage resources but without considering the budget constraint. To the best of our knowledge, there are few works addressing the online procurement with uncertain demands and budget constraint in multi-cloud environment.

Therefore, in this paper we aim to design a truthful, budget feasible, individual rational mechanism for above online procurement under uncertain demands, the objective of which is to improve the total valuation of purchased resources of the BCP. Among these properties of the mechanism, truthfulness can incentivize ICPs to report their truthful costs, which is the basic condition for decreasing purchasing cost of resources, budget feasibility can guarantee that the total payment does not exceed the limited budget, and individual rationality insures that each ICP participating the mechanism obtains non-negative utility, which can incentivize ICPs to share their residual resources. However, there are two main difficulties in our work: (1) given a limited budget, how to allocate the budget to each time step of peak-demand period; (2) with non-increasing marginal demand valuations at each time, how to incentivize the ICPs to share their idle resources with truthful costs at each time step. To address those problems, we novelly combine a truthful static auction with a learning-accept rule, and form an online sequential procurement mechanism with budget (OSPB). The contributions of this work are summarized as follows:

Firstly, we address the budget allocation problem during peak-demand period when the demand valuation functions of the BCP and the bids of ICPs are uncertain in the next time. By dividing the demand period into L stages: $g_1, g_2, \ldots g_L$, we design an accepting rule for the ICPs according to value-density threshold instead of a direct budget allocation, where value-density is the valuation brought by unit cost.

Secondly, to incentivize the ICPs to share their idle resources with truthful costs at each time step, we design truthful reverse auctions both in stage g_1 with fixed budget and stage g_l, $l > 1$ with value-density, which constitute our online sequential procurement mechanism.

Thirdly, we obtain the competitive ratio of OSPB mechanism which is closely related with the distribution of the workloads of the BCP and the bids of ICPs, and the results of simulations show that it is a more efficient mechanism compared with average budget allocation under the random environment.

The rest of this paper is organized as follows. We discuss the related work in Section 2. After describing the system model in Section 3, we propose an online sequential resource procurement mechanism in cloud market in Section 4. In Section 5 we analyze the properties of our mechanism and the competitive ratio. Section 6 reports the performance evaluation of our mechanism with real data from parallel computing centers. Section 7 finally concludes this paper.

2. Related work

Economic models are used to analyse sharing and scheduling resources among cloud networks which have attracted more interests. There are mainly three concerns in those economic models: one is how to improve the revenue of CPs by selling the residual resources, which has been investigated in many literatures [4,14–16]; the other is how to improve the valuation or profit in peak-demand state by renting resources from CPs, which also has received more attention [11,9,17]; the third is how to improve the utility of whole cloud network, in which cloud federation is introduced in the cloud networks [18,19]. In our work, we mainly focus on the second concern which is related to both cloud computing and online procurement mechanism design with budget constraint, and we only review the most related ones in this part.

Based on Stackelberg leadship model and cooperation game model, Hassan et al. [9] studied two resource purchase models in a horizontal cloud federation environment, in which the valuation function of resources is concave, and the objective of buyer is to maximize the utility. Prasad et al. [11,17] presented a cloud resource procurement approach which not only automates the selection of an appropriate cloud vendor but also implements dynamic pricing considering the quality of service. However, the above works considered only one-time procurement model without budget constraint. Zhao et al. [13] investigated online procurement of storage resources but without considering the budget constraint.

Budget feasible procurement mechanisms in general were first introduced by [20]. Singer [20] focused on the case of procurement auctions in which sellers have private costs, and the auctioneer aims to maximize a utility function on subsets of items, under the constraint that the sum of the payments provided by the mechanism does not exceed a given budget, and he provided constant approximation mechanisms for submodular functions. Furthermore, Chen et al. [21] provided constant approximation mechanisms for submodular functions, and [22,23] respectively provided approximation mechanisms for subadditive functions. All of above works still focused on one-time procurement scenario.

In the online environment, there is no related research about the online procurement with budget constraint in multi-cloud environment. However, there are some related researches in general model and other applications. With the assumption that all bidders are independent distributed, Singer et al. [24] introduced a framework for designing mechanisms with provable guarantees in crowdsourcing markets. Badanidiyuru et al. [25] studied online procurement markets where agents arrive in a sequential order and a mechanism must make an irrevocable decision whether or not to accept the service as the agent arrives which are subject to a budget constraint. The above two online mechanisms both adopted post price mechanisms. Zhao et al. [26] proposed a threshold-based accepting approaches to the online scenario which also assumed that the users' bids are independent distributed and each user only bids one time during whole online auction period.

Different from above works, the characteristics of cloud resource procurement in our work are as follows:

- The marginal demand valuation of one resource to the BCP is non-increasing at each time step. And the demand valuation functions of the BCP and the bids of ICPs are uncertain in the next time steps.
- The total payment during whole peak-demand period is constrained by a limited budget.
- Our proposed mechanism should help BCPs make an irrevocable decision about the quantity and price of purchased resources at each time step.

Thus, the problem model in our work is very different from above mentioned models in application and from those studied in the budget-feasibility framework in general.

3. System model

3.1. Description of system

We consider a set of small-scale CPs in the cloud network denoted by $F = \{1, 2, ..., |F|\}$. Each CP might become a buyer (BCP) during period of peak-demand and become a seller (ICP) during period of low-demand.

For simplicity, we will focus on a single resource type. This assumption can be easily relaxed by assuming a different procurement for each type of resources. We consider a single BCP and multiple ICPs. The BCP needs to purchase extra resources to deal with peak demand requirement during period $[t_s, t_e]$ under budget B.

We consider that the marginal demand valuation of a resource to the buyer is monotonic non-increasing. As described in literature [27,28], the valuation of a cloud provider can be expressed by the difference of jobs' value and the expect aggregate delay cost per unit of time, i.e., $V(\mu, \lambda) = V_0(\lambda) - c \cdot \lambda \frac{1}{\mu - \lambda}$, where λ is the job arrival rate, and μ is the job process rate, $V_0(\lambda)$ is total valuation of jobs per unit of time. $c \cdot \lambda \frac{1}{\mu - \lambda}$ presents the total delay cost of users per unit time. Given the arrival rate λ, the valuation function is concave and monotonically increasing in μ, and μ is linear increasing with the capacity of resources. Therefore, to improve the utility, the buyer can purchase resources (Virtual Machines) in their peak-demand period from sellers to process high job arrival rate. To decrease the dependence on single small-scale ICP, we assume the BCP purchases at most one resource from a single ICP. One resource also can express a bundle of resources. For example, the BCP can define one resource which can improve 0.5μ of process rate.

Let $v^t(j)$ denote the marginal demand valuation to the BCP brought by the j-th resource. Given the arrival rate λ^t of time t and original process rate μ_0, the marginal demand valuation of the resource is monotonic non-increasing satisfying $v^t(1) \geq v^t(2) \geq v^t(3) \geq \ldots$. At each unit of time t, the valuation of purchased resources is denoted by $V^t(k^t)$, where k^t is the total number of purchased VMs at time t, and total improved valuation at time t is $V^t(k^t) = \sum_{j \leq k^t} v^t(j)$. However, due to the uncertain demands of the BCP, it cannot obtain the actually valuation $v^t(\cdot)$ beforehand until at the beginning of time t.

At the other hand, at each time t, each ICP i associates a private cost information b_i^t for provisioning one resource. All the bids of ICP i is denoted by vector b_i, $b_i = (b_i^1, b_i^2, ..., b_i^{|T|})$.

The *protocol* for our mechanism is as follows:

- First, the BCP posts the type of requested resource, peak-demand period $T = [t_s, t_e]$, and the total budget B. The peak-demand period can be divided into multiple time units, and the time units can be one hour, half an hour, etc.
- Second, the ICP i submits the bid b_i^t to the BCP, which means that the ICP i would like to provision resources for the BCP with cost b_i^t per unit time at next unit of time t.

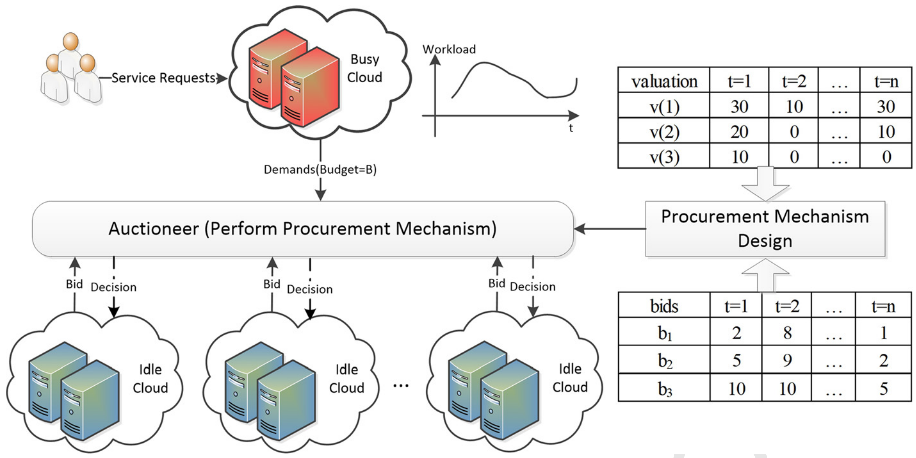


Fig. 1. Online resource procurement in multi-cloud environment with following characteristics: (1) the demand valuation functions of the BCP and the bids of residual resources are uncertain in the next time, (2) the total payment during whole peak-demand period is constrained by a limited budget, (3) the mechanism should help BCP make an irrevocable decision about the quantity and price of purchased resources at each time step.

- At the beginning of time t, the BCP collects all the bids from ICPs, and decides the winners who provision the resources and the payments at time t.

The system is illustrated in Fig. 1. The peak-demand period of the BCP is $T = [1, n]$. In the tables of the figure, the marginal demand valuation of one resource to the BCP is monotonic non-increasing at each time, and the bids of ICPs are sorted according to ascending order, both of which are uncertain and unpredictable for the auctioneer. The BCP who acts as an auctioneer receives the bids from ICPs at each time, and then performs the procurement mechanism which should make an irrevocable decision about the quantity and price of purchased resources at each time. For example, at time $t = 1$, the auctioneer receives the bids $\{2, 5, 10\}$ from three ICPs respectively, with the non-increasing marginal demand valuations $\{30, 20, 10\}$. Constrained by a total budget B, at that time, the auctioneer will decide whose resources should be purchased and how much the BCP should pay for each resource. All of the decisions should be made without the knowledge about demands and bids at time $t = 2, 3, \ldots, n$.

3.2. Design objectives

We aim to design a procurement mechanism for the BCP who is interested in maximizing the total valuation from purchased resources during the period of peak-demand T. The total valuation obtained by the buyer is expressed by $\sum_{t\in T}\sum_{j\le k^t} v^t(j)$. The problem is:

$$\begin{aligned} \max & \sum_{t\in T}\sum_{j\le k^t} v^t(j) \\ s.t. & \sum_{t\in T}\sum_{i\in F} p_i^t \le B \\ & \sum_{i\in F} x_i^t = k^t \ \forall t \in T \\ & x_i^t = 0 \ or \ 1 \ \forall i \in F, t \in T \end{aligned} \tag{1}$$

where, x_i^t is the allocation of i at time t. $x_i^t = 1$ if i is the winner at time t, otherwise, $x_i^t = 0$. p_i^t is the payment of i at time t.

Since resources are possessed by different ICPs, it is reasonable to assume that each ICP is selfish but rational. Hence, each ICP only wants to maximize her own utility, and might misreport her cost. Our objective is to design a mechanism satisfying computational efficiency, individually rationality, budget feasibility and truthfulness.

Definition 1. *(Computationally Efficient)* A mechanism is computationally efficient if both the allocation and payment can be computed in polynomial time during the peak-demand period of the BCP.

Definition 2. *(Individually Rational)* A mechanism is individual rational if every ICP i derives a non negative utility by participating the mechanism. Formally, $\forall b_i, \hat{b}_{-i}$,

Please cite this article in press as: J. Han et al., An online sequential procurement mechanism under uncertain demands in multi-cloud environment, Int. J. Approx. Reason. (2018), https://doi.org/10.1016/j.ijar.2018.09.006

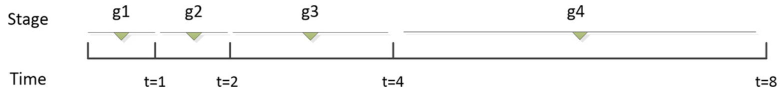


Fig. 2. Illustration of a multiple-stage learning-accepting process when $|T| = 8$.

$$u_i(b_i, \hat{b}_{-i}) \geq 0$$

Definition 3. *(Budget feasible)* A mechanism is budget feasible if the total payment does not exceed a given budget.

Definition 4. *(Truthful)* A mechanism is truthful if bidding true cost of resources maximizes the utility of any ICP i irrespective of the bid of other ICPs. Formally, $\forall b_i, \hat{b}_{-i}$,

$$u_i(b_i, \hat{b}_{-i}) \geq u_i(\hat{b}_i, \hat{b}_{-i})$$

Now, there are several nontrivial challenges we need to overcome: first, each ICP's cost is unknown and need to be reported in a truthful manner; Second, since the payment is constrained by the budget and the demands of the next time are uncertain and change dynamically, the budget B should be allocated reasonably during time period $[t_s, t_e]$ to improve the total valuation.

A straightforward solution for budget constraint problem is to allocate an average budget to each time unit. However, since the fluctuation of the demand valuation functions due to the uncertain demands, it will lose the higher valuation in time units with higher demand. General solutions of online auctions for related problems adopt a two-stage sampling-accepting process: In the sampling stage, the arrival users always are rejected and auctioneer made an informed decision on choosing the winners from the rest of users. However, in this work, the ICPs are non-independent during the whole procurement period, that is, an ICP appearing at current time stage also can appear at next time stage. Traditional sampling-accepting might result in untruthful bid during initial sampling stage. To address the above challenges, we design a mechanism of online sequential procurement auctions with budget constraint(OSPB), based on a multiple-stage learning-accepting process. Similar to literature [20,26], we divide the whole time period $[t_s, t_e]$ into L stages, $L = \lfloor \log_2 |T| \rfloor + 1 \geq 2$. The illustration of stages is shown by Fig. 2. The stage g_l ends at time $\lfloor 2^{l-L}|T| \rfloor$. Correspondingly, the stage-budget in each stage is allocated as $2^{1-L}B, 2^{1-L}B, 2^{2-L}B, \ldots 2^{-1}B$. Before the beginning of a new stage, we extend the learning period to the whole historical allocation stages, and derive a value-density threshold according to the historical bids and stage-budget, which can be used for allocation decision in next stage.

In addition to the properties of the mechanism, we are concerned about the competitive ratio of the mechanism. The competitive ratio expresses how effectively does the performance of the online procurement mechanism 'compete' with that of an offline mechanism that is given complete information about the demands of the BCP and the bids of ICPs. Obviously, there is no mechanism can achieve a constant competitive ratio if there is no restriction on demands or bids of ICPs. Therefore, we make the following assumptions.

- The bid of each ICP at any time step subjects to the same distribution. Let b_{max} denote the maximal bid. We assume $b_{max} \leq \gamma B$ and $\gamma < \frac{1}{2}$.
- Under the scenario of uncertain demands, we assume all ICPs are myopic and only optimize their utility from the current time step.
- We assume that the demands of the BCP are δ-degree balance.

In the last assumption, we introduce a concept about balance degree. Let T_1 and T_2 denote the first and second half of demand period $[t_s, t_e]$, V_{T_1} and V_{T_2} denote the valuation obtained in T_1 and T_2 with half of budget respectively by offline proportional sharing methods, which is described in Algorithm 2. We call the demands of the BCP during whole demand period are *δ-degree balance* if it always satisfies $\frac{\max\{V_{T_1}, V_{T_2}\}}{\min\{V_{T_1}, V_{T_2}\}} \leq \delta, \delta > 1$. This assumption is easy to be satisfied in a periodic scene. On the other hand, since the last assumption implies that we should have a prior knowledge about balance degree δ, we use a estimate value instead of the real balance degree δ in extend simulation experiments, and the results of the experiments show that the performance of mechanism using estimate value of δ is very close to that using the real value of δ.

4. Online sequential procurement with budget (OSPB)

In this section, we design online sequential procurement auctions of cloud resources with budget constraint which consist of three algorithms:

1. StaticPurchasing: It is a static allocation algorithm with budget constraint which is only executed at first stage g_1 shown in Fig. 2;

2. ThresholdGetting: It is a value-density threshold getting algorithm which is executed at the end of stage $g_l, l \leq \lfloor \log_2 |T| \rfloor$;
3. AcceptingRule: It is an online accepting algorithm which is executed at each time step t in stage $g_l, l > 1$.

As aforementioned, traditional sampling in which the users always are rejected in sampling stage will result in untruthful mechanism. Thus, firstly we design a truthful procurement mechanism with average budget $B_a = \frac{B}{|T|}$ performed at initial stage g_1. Then, after g_1, instead of sampling, we perform the procurements according to learning-accept rule.

4.1. OSPB mechanism design

Algorithm 1: StaticPurchasing

In this section, we first introduce algorithm *StaticPurchasing*(b^t, B_a), a static truthful procurement with budget constraint shown in Algorithm 1, where the inputs b^t, B_a are a set of bids of time t and average budget per unit of time. At the beginning of time t in first stage g_1, we sort the bids so that $b_1^t \leq b_2^t \leq \ldots \leq b_{n^t}^t$, where n^t is the number of bids at time t, and find the largest k^t such that $b_{k^t}^t \leq B_a/k^t$. That is, k^t is the place where the curve of the increasing bids intersects the hyperbola B_a/k^t. The allocated set is $\{1, 2, ..., k^t\}$, and payment of each allocated ICP is $\min\{B_a/k^t, b_{k^t+1}\}$. In Section 5 we will show that the allocation rule and the payment policy result in a truthful mechanism.

Algorithm 1: Static procurement with budget constraint (StaticPurchasing).

input : bids at time t b^t, average budget B_a
output: allocation $x^t = \{x_i^t\}$, payment $p^t = \{p_i^t\}$

sort bids s.t. $b_1^t \leq b_2^t \leq \ldots \leq b_{n^t}^t$;
$j \leftarrow 1$;
while $j \cdot b_j^t \leq B_a$ *and* $j \leq n^t$ **do**
  $x_j^t \leftarrow 1; j \leftarrow j + 1$;
end
$k^t \leftarrow j - 1$;
for *each i* **do**
  $p_i^t \leftarrow x_i^t \min\{B_a/k^t, b_{k^t+1}^t\}$;
end

From second stage g_2, we start to perform an accepting rule, under which the allocation at each time step is not constrained by average budget, but constrained by the total budget of current stage. The resource of an ICP can be accepted if her marginal value-density is not less than a threshold value which is designed based on value-density threshold learned from historical information. In Algorithm 2, we design an algorithm to get the value-density threshold denoted by ρ.

Algorithm 2: Proportional share allocation: Value-density threshold getting algorithm (ThresholdGetting).

input : sample time T_s, sample bids b_s, stage budget B_s
output: value-density threshold ρ

1 sort bids s.t. $b_1^t \leq b_2^t \leq \ldots \leq b_{n^t}^t$ for each $t \in T_s$;
2 $V \leftarrow 0$;
3 $(t^*, i^*) \leftarrow \arg\max_{t \in T_s, i \in F}\{\frac{v^t(i)}{b_i^t}\}$;
4 **while** $\frac{v^{t^*}(i^*)}{b_{i^*}^{t^*}} \geq \frac{V}{B_s}$ **do**
5   $V \leftarrow V + v^{t^*}(i^*)$;
6   $v^{t^*}(i^*) \leftarrow 0$;
7   $(t^*, i^*) \leftarrow \arg\max_{t \in T_s, i \in F}\{\frac{v^t(i)}{b_i^t}\}$;
8 **end**
9 $\rho \leftarrow V/B_s$;

Algorithm 2: ThresholdGetting

Algorithm *ThresholdGetting*(T_s, b_s, B_s) realizes the computing method of value-density threshold ρ, which is processed at the end of each stage except the last, where T_s, b_s, B_s denote the historical allocation period, historical bids and total budget in historical stages (also named sample time, sample bids and sample budget).

The main steps of *ThresholdGetting*(T_s, b_s, B_s) shown in Algorithm 2 are as follows:

- Step 1: in Line 1, we sort the sample bids of each time $t \in T_s$ so that $b_1^t \leq b_2^t \leq \ldots \leq b_{n^t}^t$;

Please cite this article in press as: J. Han et al., An online sequential procurement mechanism under uncertain demands in multi-cloud environment, Int. J. Approx. Reason. (2018), https://doi.org/10.1016/j.ijar.2018.09.006

- Step 2: in Lines 3–9, we find the largest marginal value-density $v^t(i)/b_i^t$ among sample period T_s, denoted by $v^{t^*}(i^*)/b_{i^*}^{t^*}$. When $v^{t^*}(i^*)/b_{i^*}^{t^*}$ is not less than V/B_s, the bid can be accepted, $v^{t^*}(i^*)$ is set to zero after added to total valuation V, and this step is repeated, where V is the sum of valuation of accepted resources;
- Step 3: if $v^{t^*}(i^*)/b_{i^*}^{t^*}$ is less than V/B_s, the value-density threshold ρ is obtained in Line 10 , $\rho = V/B_s$.

Actually, Algorithm *ThresholdGetting* is a proportional share allocation derived from [20], which realizes an offline allocation with a constant competitive ratio. The result of this method also is a benchmark in our experiments.

Algorithm 3: AcceptingRule

After value-density threshold learning by algorithm *ThresholdGetting*, next we explain how the accepting rule performs based on value-density threshold at time $t \in g_l, l > 1$.

Firstly, at the beginning of time t, we sort all bids of that time satisfying $b_1^t \le b_2^t \le \ldots \le b_{n^t}^t$. This sorting, in the presence of non-increasing marginal demand valuations, implies:

$$v^t(1) \ge v^t(2) \ge \ldots \ge v^t(n^t).$$

Then, denote ρ' be the threshold value selected by the accepting rule which is according to the value-density threshold ρ learned by Algorithm *ThresholdGetting*. That is, the marginal value-density of each accepted resource at time t is not less than threshold value ρ'. Then, at each time step t, we find a maximal bid $b_{k^t}^t$ satisfying $\frac{v^t(k^t)}{b_{k^t}^t} \ge \rho'$, and the payment is $\min\{v^t(k^t)/\rho', b_{k^t+1}^t\}$.

Algorithm 3: Allocation with accepting rule (AcceptingRule).

```
input  : remaining budget B̄, bids at time t b^t, threshold value ρ'
output: remaining budget B̄, allocation x^t, payment p^t
1  sort bids s.t. b^t_1 ≤ b^t_2 ≤ ... ≤ b^t_{n^t};
2  B_r ← B̄;
3  k^t ← max{i | v^t(i)/b^t_i ≤ ρ'};
4  for j = 1 to n^t do
5      if v^t(j)/b^t_j ≥ ρ' then
6          x'_j ← 1;
7          p'_j ← min{v^t(k^t)/ρ', b_{k^t+1}};
8          B̄' ← B̄ − p'_j;
9      end
10 end
11 if B̄' ≥ 0 then
12     B̄ ← B̄'; x^t ← x'; p^t ← p';
13 else
14     B̄ ← B_r;
15     while B̄ > 0 do
16         Select j s.t. x'_j = 1 randomly;
17         if B̄ ≥ p'_j then
18             x^t_j ← 1;
19             p^t_j ← p'_j;
20             B̄ ← B̄ − p_j;
21         end
22     end
23 end
```

The steps of allocation with accepting rule at time t shown in Algorithm 3 is as follows:

- Step 1: In Lines 1–10, we sort the bids satisfying $b_1^t \le b_2^t \le \ldots \le b_{n^t}^t$, and find a set of all ICPs whose bids satisfy the threshold value condition ($\frac{v^t(j)}{b_j^t} \ge \rho'$). For easy description, we denote the set of those ICPs by A. At the same time, for each ICP $i \in A$ we compute the payment which is the minimal value between $\{v^t(k^t)/\rho', b_{k^t+1}\}$, where k^t is the number of bids satisfied threshold value condition, i.e., $k^t = |A|$.
- Step 2: In Lines 11–23, check the feasibility of the stage budget at current time and make the decision for the allocation. If the remaining budget is sufficient, accept all the ICPs in set A, otherwise, select the ICP in set A randomly and accept it until the budget is not sufficient.

It is worth noting that, to incentivize ICPs to report their truthful cost, when there is not enough budget at some time t of one stage, we will perform randomly selection among all ICPs who satisfy $\frac{v^t(j)}{b_j^t} \ge \rho'$ until it meets an ICP whose payment

Please cite this article in press as: J. Han et al., An online sequential procurement mechanism under uncertain demands in multi-cloud environment, Int. J. Approx. Reason. (2018), https://doi.org/10.1016/j.ijar.2018.09.006

is larger than budget remaining. Algorithm 4 shows the complete process of the online sequential procurement mechanism, the steps of which are as follows.

- Step 1: In Line 1, it initializes the start time t, the start stage l, the end time and the budget of first stage.
- Step 2: In Lines 4–13, it adopts the static procurement at each time step in stage g_1, and adopts the auction based accepting rule at each time step in other stages. At each time step $t \in g_l, 1 < l \le \lfloor \log_2 |T| \rfloor$, we preform the accepting rule using threshold value $\rho' = \rho$, where ρ is the value-density threshold learned by historical stages. At last stage $l = \lfloor \log_2 |T| \rfloor + 1$, with probability $(\delta - 1)/(2\delta - 1)$ we preform the accepting rule using threshold value $\rho' = \rho$, and with probability $\delta/(2\delta - 1)$ using threshold value $\rho' = \rho/\delta$.
- Step 3: In Lines 14–19, at the end time of each stage except the last, there are two works need to be completed. One is that the value-density threshold ρ should be updated which is completed by Algorithm 2, and the other is that the end time, the stage and the budget of next stage are updated, which also means a beginning of a new stage.

Algorithm 4: Online sequential procurement with budget constraint (OSPB).

input : Budget B, Demand Peroid T
output: allocation $\{x^t\}$, payment $\{p^t\}$,$t \in T$

1 $(t, l, T', B') \leftarrow (1, 1, \frac{|T|}{2^{\lfloor \log_2 |T| \rfloor}}, \frac{B}{2^{\lfloor \log_2 |T| \rfloor}})$;
2 $B_g \leftarrow B'$;
3 **while** $t \le |T|$ **do**
4   **if** $l = 1$ **then**
5     $(x^t, p^t) \leftarrow$ Algorithm 1: StaticPurchasing(b^t, $B/|T|$);
6   **end**
7   **if** $l > 1$ *and* $l \le \lfloor \log_2 |T| \rfloor$ **then**
8     $(x^t, p^t, B_g) \leftarrow$ Algorithm 3: AcceptingRule(B_g, b^t, ρ);
9   **end**
10   **if** $l = \lfloor \log_2 |T| \rfloor + 1$ **then**
11     $(x^t, p^t, B_g) \leftarrow$ Algorithm 3: AcceptingRule(B_g, b^t, ρ) with probability $(\delta - 1)/(2\delta - 1)$;
12     $(x^t, p^t, B_g) \leftarrow$ Algorithm 3: AcceptingRule($B_g, b^t, \rho/\delta$) with probability $\delta/(2\delta - 1)$;
13   **end**
14   **if** $t = \lfloor T' \rfloor$ *and* $t \ne |T|$ **then**
15     $\rho \leftarrow$ Algorithm 2: ThresholdGetting(T',b_s,B');
16     $T' \leftarrow 2T'$; $B' \leftarrow 2B'$;
17     $B_g \leftarrow B'$;
18     $l \leftarrow l + 1$;
19   **end**
20   $t \leftarrow t + 1$;
21 **end**

5. Analysis of OSPB mechanism

5.1. Properties of OSPB

Firstly, we introduce the related definitions and theorems used in this section. In a single round procurement mechanism, let b be a profile of sellers bids, b_i, b_{-i} express the bid of i and bid profile of others excluding i.

Definition 5 *(Monotonic Allocation).* An allocation x on a single parameter domain is called monotonic in b_i if for every b_{-i} and every $b_i \ge b'_i \in R$ we have that $x_i(b_i, b_{-i}) = 1$ implies that $x_i(b'_i, b_{-i}) = 1$. That is, if bid b_i makes i win, then so will every lower bid $b'_i < b_i$.

Definition 6 *(Critical Value).* The critical value of a monotonic allocation function x on a single parameter domain is $c_i(b_{-i}) = \inf_{b_i : x(b_i, b_{-i}) = 0} b_i$.

Theorem 1 *(Myersons Theorem[29]). A normalized mechanism $(x, p_1, \ldots, p_n)$ on a single parameter domain is truthful if and only if the following conditions hold:*
(a) Allocation x is monotonic in every b_i;
(b) The payment to every winner is the critical value.

Now, based on Myersons Theorem, we can obtain the following properties of our mechanism. The main notation used in following is shown in Table 1.

Table 1
Description of notations.

Notation	Description
F	set of all sellers
B	total budget
L	the number of stages
T	the period of peak demand, $T = [t_s, t_e]$
B_a	average budget per unit of time
b_i^t	the bid of seller i at time t
$v^t(i)$	the marginal demand valuation of i-th resource at time t
n^t	the number of sellers at time t
k^t	the number of purchased resources at time t
ρ	value-density threshold learned from sample periods
ρ'	threshold value selected by the buyer based on value-density threshold ρ
p_i^t	the payment of seller i at time t

Lemma 5.1. *The auction of static procurement shown in Algorithm 1 is truthfulness, individually rationality and budget feasibility.*

Proof. In Algorithm 1, each allocated ICP at time t in first stage g_1 gets same payment $p_i^t = \min\{B_a/k^t, b_{k^t+1}^t\}$, where B_a/k^t implies that each purchase resource costs a same budget, and $b_{k^t+1}^t$ is the lowest bid among the losers.

Next, we prove the truthfulness of the auction of static procurement. According to Myersons Theorem, the mechanism is truthful if it satisfies: (1) the allocation is monotonic; (2) the payment to each winning provider is her critical value. In the auction, the monotonicity of the selection rule is obvious. If ICP i is selected at time step t by reporting b_i^t, it must be selected as reporting a smaller cost $\hat{b}_i^t < b_i^t$.

To obtain allocation under the auction of Algorithm 1, it must satisfy $b_i^t \le b_{k^t+1}^t$ and $b_i^t \le B_a/k^t$ for each allocated ICP $i \le k^t$. Therefore, the critical value for each allocated ICP i is $\min\{B_a/k^t, b_{k^t+1}^t\}$. Thus, the payment to each winning provider is her critical value, and the truthfulness is proved.

According to the payment, we have that $p_i^t \le B_a/k^t$, and $\sum_i p_i^t \le B_a$. Since $b_i^t \le B_a/k^t$ and $b_i^t \le b_{k^t+1}^t$ for each allocated ICP, $b_i^t \le p_i^t$, we can have that the utility $u_i(b_i^t, b_{-i}^t) = p_i^t - b_i^t \ge 0$. Therefore, the mechanism shown by Algorithm 1 is budget feasibility and individually rationality. □

Lemma 5.2. *The auction using accepting rule according to the value-density threshold is truthfulness.*

Proof. Firstly, we discuss the auction using accepting rule with sufficient budget. It is obvious that the allocation is monotonic, so we only need to demonstrate that the payment is critical value. According to Algorithm 3, the payment of an allocated ICP i is $\min\{v^t(k^t)/\rho', b_{k^t+1}^t\}$. Since a bid $\hat{b}_i^t > v^t(k^t)/\rho'$ or $\hat{b}_i^t > b_{k^t+1}^t$ cannot be accepted, $\min\{v^t(k^t)/\rho', b_{k^t+1}^t\}$ is the minimal bid $\hat{b}_i^t$ that makes the ICP i obtain allocation, which is the critical value payment. According to Myersons Theorem, for each i, reporting true bid at time t can maximize the utility of current auction.

In addition, randomly selecting rule when the budget is not sufficient obviously does not affect the truthfulness of the auction. □

Lemma 5.3. *The OSPB mechanism is individually rationality and budget feasibility.*

Proof. We have proved that, in each time $t \in g_1$, the auction with fixed budget is individually rationality and budget feasibility. Next, we prove that these properties still be remained in the auctions using accepting rule according to value-density threshold.

Let I_{t_j} denote the ICP with the j-th lowest bid at time $t \notin g_1$, whose bid is b_j^t. k^t is the maximal j which satisfies $v^t(j)/b_j^t \ge \rho'$. In OSPB, the payment of ICP I_{t_j} is $p_j^t = \min\{v^t(k^t)/\rho', b_{k^t+1}^t\}$ if it is allocated at time t, otherwise $p_j^t = 0$. Since $v_{k^t}^t/\rho' \ge b_j^t$ and $b_{k^t+1}^t \ge b_j^t$ for each allocated ICP at each time t, we can have that $p_j^t \ge b_j^t$.

Therefore, the OSPB satisfies individually rationality.

In each stage g_l, the auction satisfies budget feasibility, i.e., $B_l \ge \sum_{t \in g_l} p^t$, where B_l is the budget at stage g_l. Let the total payment denote by P.

$P = \sum_{t \in g_1} \sum_i p_i^t + \sum_{t \notin g_1} \sum_i p_i^t \le \sum_{t \in g_1} \frac{B}{|T|} \lfloor 2^{1-L}|T| \rfloor + \sum_{l=2}^{L} 2^{l-L} B \le 2^{1-L}B + 2^{2-L}B + \dots + 2^{-1}B = B.$

Thus, the OSPB mechanism is individually rationality and budget feasibility. □

Lemma 5.4. *The OSPB mechanism is computationally efficient.*

Proof. Since the mechanism runs online, we only need to focus on the computation complexity at each time step $t \in T$.

Please cite this article in press as: J. Han et al., An online sequential procurement mechanism under uncertain demands in multi-cloud environment, Int. J. Approx. Reason. (2018), https://doi.org/10.1016/j.ijar.2018.09.006

In OSPB mechanism shown in Algorithm 4, either function StaticPurchasing (runs at time $t \in g_1$) or function AcceptingRule (runs at time $t \in g_l, l > 1$) is called at each time step. Both function StaticPurchasing and AcceptingRule take $O(n^t \log n^t)$ time. Since there are at most $|F|$ bidders at each time, the computation complexity of each function is at most $O(|F| \log |F|)$.

At some special time steps which are the ends of the stages, we also need to compute value-density threshold by running function ThresholdGetting (Line 15 in Algorithm 4). Next, we analyze the complexity of function ThresholdGetting shown in Algorithm 2. The computation complexity of the sorting (Line 1 in Algorithm 2) is $O(|T_s||F| \log |F|)$, and the running time of finding value-density threshold ρ (Lines 4–8 in Algorithm 2) is bounded by $O(|F||T_s|^2)$. Thus, the running time of Algorithm 2 is bounded by $O(|F||T_s| \max\{\log |F|, |T_s|\})$. The maximum value of sample time $|T_s|$ is $|T|/2$, and the computation complexity of Algorithm 2 is bounded by $O(|F||T| \max\{\log |F|, |T|/2\})$.

Since $|F| \log |F| \leq |F||T| \max\{\log |F|, |T|/2\}$, the computation complexity at each time step is $O(|F||T| \max\{\log |F|, |T|/2\})$. The lemma is proved. □

Theorem 2. *The OSPB mechanism is computationally efficient, individually rationality, budget feasibility and truthfulness.*

OSPB mechanism is truthfulness according to Lemma 5.1 and 5.2. Then, according to 5.3 and Lemma 5.4, the theorem is derived directly.

5.2. Competitive analysis of OSPB

The goal of our mechanism is to maximize the total valuation of the BCP. To quantify the performance of the mechanism we compare the online solution with the optimal solution that can be obtained in the offline scenario where the BCP has full knowledge about ICPs' types. The ratio between the online solution and the optimal solution in the worst case is defined as *competitive ratio*.

Let V_{opt} denote the valuation obtained by offline optimal allocation, V_{pro} denote the maximal valuation by offline proportional share allocation in Algorithm 2, both of which are obtained during the demand period T with budget B.

Lemma 5.5. $\frac{V_{pro}}{V_{opt}} > \frac{1-\gamma}{2-\gamma}$*, where* $b_{max} \leq \gamma B$ *and* $\gamma < \frac{1}{2}$.

Proof. According to the offline bids, we sort all the bids by $\frac{v^t(j)}{b^t_j}$. To simplify notation in offline sorting we will write $\frac{v_i}{b_i}$ instead of $\frac{v^t(j)}{b^t_j}$, where i denote the i-th highest ratio of the valuation to bid during the period $[t_s, t_e]$. The optimization problem in this scenario is essentially a budgeted maximum coverage problem, which is a well-known NP-hard problem. By relaxing the problem to allow the fraction allocation, we can obtain an relaxed optimal solution using a greedy algorithm. Let k^* and k denote the allocation number of optimal allocation and proportional share allocation in Algorithm 2, respectively. Obviously, it satisfies $k^* \geq k$. Therefore, the sorting is:

$$\frac{v_1}{b_1} \geq \frac{v_2}{b_2} \geq \ldots \geq \frac{v_k}{b_k} \geq \frac{v_{k+1}}{b_{k+1}} \geq \ldots \frac{v_{k^*}}{b_{k^*}}.$$

k satisfies $\frac{v_k}{b_k} \geq \frac{\sum_{i \leq k} v_i}{B}$ and $\frac{v_{k+1}}{b_{k+1}} < \frac{\sum_{i \leq k+1} v_i}{B}$.
$V_{pro} = \sum_{i \leq k} v_i$, and $V_{opt} = \sum_{i \leq k^*} v_i = \sum_{i \leq k} v_i + \sum_{i \in [k+1, k^*]} v_i$.
Since $\frac{v_{k+1}}{b_{k+1}} < \frac{\sum_{i \leq k} v_i + v_{k+1}}{B}$,

$$v_{k+1} < b_{k+1}(\sum_{i \leq k} v_i + v_{k+1})/B \leq \gamma(\sum_{i \leq k} v_i + v_{k+1}).$$

We can have that $v_{k+1} < \frac{\gamma}{1-\gamma} \sum_{i \leq k} v_i$.

For each $j > k$, it cannot be allocated by Proportional Share allocation in Algorithm 2, and it satisfies $\frac{v_j}{b_j} \leq \frac{v_{k+1}}{b_{k+1}} < \frac{\sum_{i \leq k} v_i + v_{k+1}}{B} < \frac{\sum_{i \leq k} v_i}{B} \frac{1}{1-\gamma}$.

So, $\sum_{j \in [k+1, k^*]} v_j < \frac{\sum_{i \leq k} v_i}{B} \frac{1}{1-\gamma} \sum_{j \in [k+1, k^*]} b_j \leq \frac{1}{1-\gamma} \sum_{i \leq k} v_i$. The last inequality satisfies because the total payment is constraint by budget B.

$$V_{opt} = \sum_{j \leq k^*} v_j < (1 + \frac{1}{1-\gamma}) \sum_{i \leq k} v_i = \frac{2-\gamma}{1-\gamma} V_{pro}.$$

The conclusion can be derived: $\frac{V_{pro}}{V_{opt}} > \frac{1-\gamma}{2-\gamma}$. □

Please cite this article in press as: J. Han et al., An online sequential procurement mechanism under uncertain demands in multi-cloud environment, Int. J. Approx. Reason. (2018), https://doi.org/10.1016/j.ijar.2018.09.006

Before the last stage, we have obtained the value-density threshold ρ_1 which is learned from the first half of time period by Algorithm 2 with budget $B/2$. Let ρ_2 denote the value-density threshold learned from the second half of time period by Algorithm 2 with budget $B/2$. According to the assumption of δ-degree balance of demands, $\frac{\max\{V_{T_1}, V_{T_2}\}}{\min\{V_{T_1}, V_{T_2}\}} \le \delta$, we can have that $\max\{\rho^1/\rho^2, \rho^2/\rho^1\} \le \delta$.

Theorem 3. *Let $E[V'_{T_2}]$ be the expected valuation obtained during the second half of the demand period by OSPB mechanism. We have*

$$E[V'_{T_2}] \ge \frac{1}{\delta+1}\frac{(1-\gamma)(1-2\gamma)}{2(2-\gamma)}V_{opt}.$$

Proof. Since $p_j^t = \min\{v^t(k^t)/\rho', b^t_{k^t+1}\}$ for any $j < k^t$, we have that $p_j^t \le v^t(k^t)/\rho' \le v^t(j)/\rho'$.

$$V'_{T_2} = \sum_{t\in T_2}\sum_{j\le k^t} v^t(j) \ge \sum_{t\in T_2}\sum_{j\le k^t} p_j^t \rho'.$$

In the OSPB mechanism, we respectively use $\rho' = \rho^1$ and $\rho' = \rho^1/\delta$ with probability $(\delta-1)/(2\delta-1)$ and $\delta/(2\delta-1)$ in the last stage corresponding to the period T_2. Denote ρ be the value-density threshold by learning during whole peak-demand period T, the total valuation is $V_{pro} = B\rho$. There are only two cases: $\rho^1 \le \rho \le \rho^2 \le \rho^1\delta$ and $\rho^1/\delta \le \rho^2 \le \rho \le \rho^1$.

Case 1: $\rho^1 \le \rho \le \rho^2 \le \rho^1\delta$**.**

(1) Since $\rho' = \rho^1$, we have $\sum_{t\in T_2}\sum_{j\le k^t} p_j^t\rho' \ge \sum_{t\in T_2}\sum_{j\le k^t} p_j^t\rho^1$. The reason that an ICP satisfied the value-density constraint cannot be selected is when there is not enough budget to pay. According to the assumption that the maximal bid is not larger than γB, the remaining budget must be less than γB. So, we have that $\sum_{t\in T_2}\sum_{j\le k^t} p_j^t\rho^1 \ge (B/2-\gamma B)\rho^1$. In this case, $\rho^1 \le \rho \le \rho^2$. We can have that

$$V'_{T_2} \ge B(1/2-\gamma)\rho/\delta = V_{pro}(1/2-\gamma)/\delta = V_{pro}\frac{1-2\gamma}{2\delta}.$$

(2) If $\rho' = \rho/\delta$, $\sum_{t\in T_2}\sum_{j\le k^t} p_j^t\rho' = \sum_{t\in T_2}\sum_{j\le k^t} p_j^t\rho^1/\delta$. Similar to (1), the reason that an ICP satisfied the value-density constraint cannot be selected also is when there is not enough budget to pay for those users. Therefore, the remaining budget must be less than γB. So, we have that $\sum_{t\in T_2}\sum_{j\le k^t} p_j^t\rho^1/\delta \ge B(1/2-\gamma)\rho^1/\delta$. Since $\rho \le \rho^1\delta$,

$$B(1/2-\gamma)\rho^1/\delta \ge B(1/2-\gamma)\rho/\delta^2 = V_{pro}(1/2-\gamma)/\delta^2 = V_{pro}\frac{1-2\gamma}{2\delta^2}.$$

In this case we get expect valuation

$$E[V'_{T2}] \ge V_{pro}\frac{(1-2\gamma)}{2\delta}\frac{\delta-1}{2\delta-1} + V_{pro}\frac{1-2\gamma}{2\delta^2}\frac{\delta}{2\delta-1} = V_{pro}\frac{1-2\gamma}{2}\frac{1}{2\delta-1}.$$

Case 2: $\rho^1/\delta \le \rho^2 \le \rho \le \rho^1$**.**

(1) If $\rho' = \rho^1$, the worst case is that there is no ICP allocated due to the value-density threshold constraint.

(2) If $\rho' = \rho^1/\delta$, $\sum_{t\in T_2}\sum_{j\le k^t} p_j^t\rho' = \sum_{t\in T_2}\sum_{j\le k^t} p_j^t\rho^1/\delta \ge \frac{B(1/2-\gamma)\rho^1}{\delta} \ge \frac{B(1/2-\gamma)\rho}{\delta} = \frac{V_{pro}(1/2-\gamma)}{\delta} = V_{pro}\frac{1-2\gamma}{2\delta}$.

In this case we get expect valuation

$$E[V'_{T2}] \ge V_{pro}\frac{1-2\gamma}{2\delta}\frac{\delta}{2\delta-1} = V_{pro}\frac{1-2\gamma}{2}\frac{1}{2\delta-1}.$$

Therefore, we can have the expect valuation obtained at the second half of the time:

$$E[V'_{T2}] \ge V_{pro}\frac{1}{2\delta-1}\frac{1-2\gamma}{2}$$

According to the Lemma 5.5, we have

$$E[V'_{T2}] > \frac{1}{2\delta-1}\frac{(1-2\gamma)(1-\gamma)}{2(2-\gamma)}V_{opt}. \quad \square$$

Corollary 5.1. $\frac{E[V_{OSPB}]}{V_{opt}} > \frac{1}{2\delta-1}\frac{(1-\gamma)(1-2\gamma)}{4(2-\gamma)}$.

Let $E[V'_{T_1}]$ be the expected valuation obtained during the first half of the demand period by OSPB mechanism. Since $E[V_{OSPB}] = E[V'_{T_1}] + E[V'_{T_2}] \ge E[V'_{T_2}]$, the corollary can be derived directly.

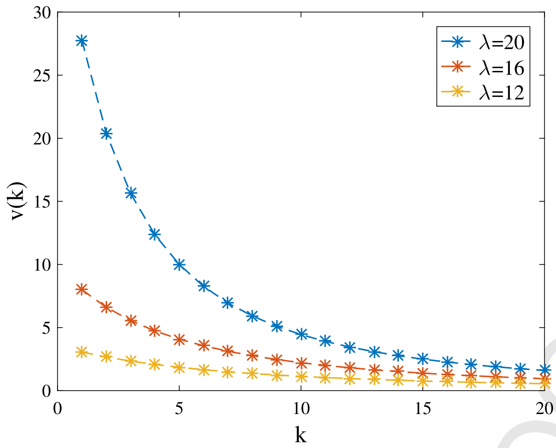


Fig. 3. Illustration of the non-increasing marginal demand valuations. The marginal demand valuation of k-th resource we selected is $v(k) = \frac{c\cdot\lambda}{(\mu_0+k-\lambda)^2}$. The original process speed of the BCP without purchasing resources is $\mu_0 = 25$, and delay cost per unit of time $c = 50$, $v(k)$ is the valuation brought by k-th purchased resource. For any k, the higher job arrival rate brings higher marginal demand valuation. $v(k)|_{\lambda=20} > v(k)|_{\lambda=16} > v(k)|_{\lambda=12}$, and marginal valuation is non-increasing $v(1)|_\lambda \geq v(2)|_\lambda \geq \ldots$ for any given λ at each time. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

6. Performance evaluation

To evaluate the performance of our sequential procurement mechanism, we made simulations of proposed mechanism using real data from two parallel computing centers, and compared OSPB mechanism against the following three benchmarks. The first benchmark is the offline optimal solution which has full knowledge about all the bids without considering the strategy behavior of ICPs. As mentioned before, the optimization problem is a well-known NP-hard problem in this scenario. So we obtain a relaxed optimal solution by allowing fraction allocation and using a greedy algorithm. The second benchmark is the offline proportional share mechanism under budget constraint proposed by Singer Y. [20], which is also truthfulness and budget feasibility. The third benchmark is an approach which adopts average budget at each time step. In addition, to deal with the case that balance degree δ is unknown, we introduce a estimate value of δ according to the historic information.

The performance metric is the total valuation obtained by the BCP.

6.1. Simulation setup

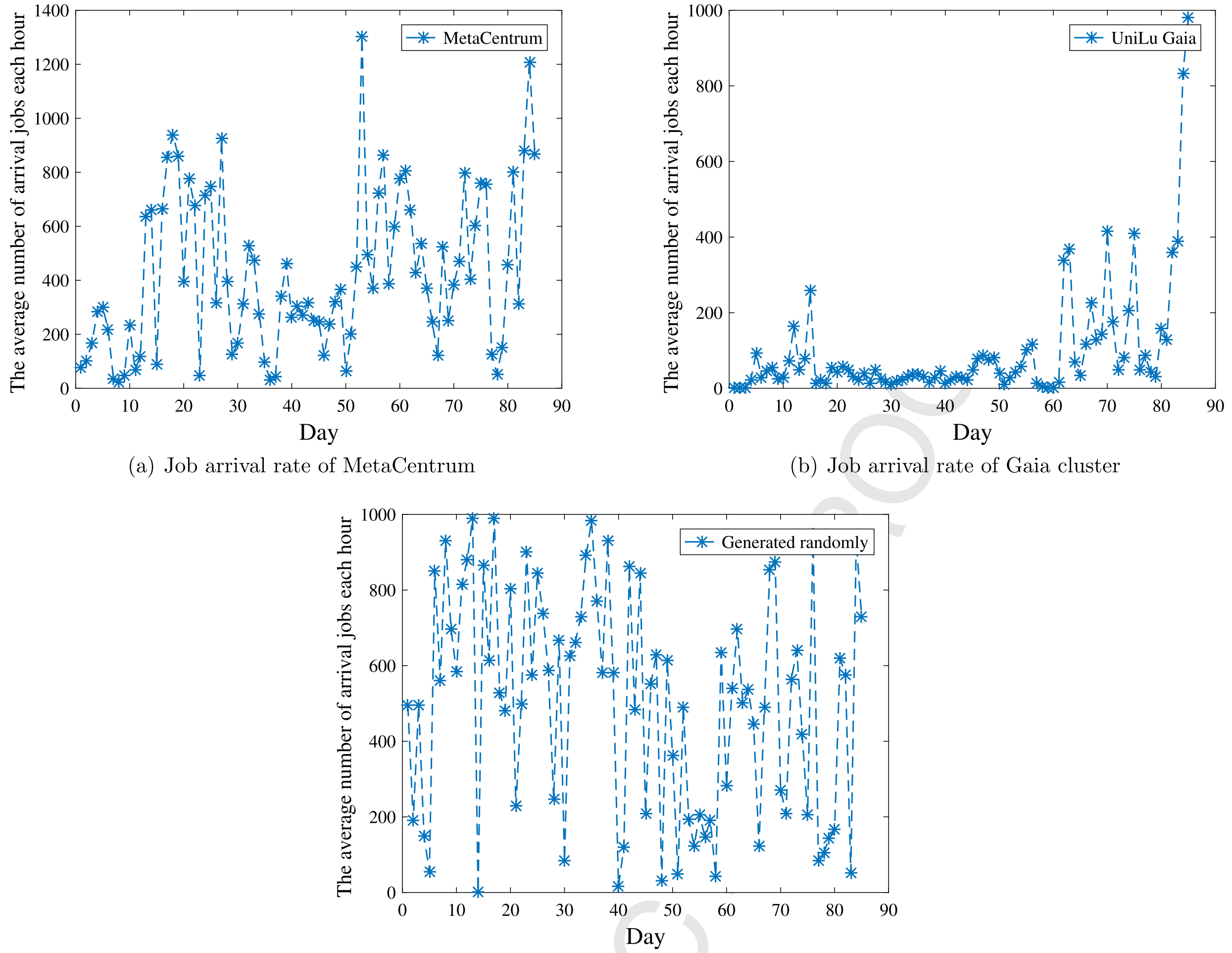
In this subsection, we introduce how to obtain the data used in our experiments, which are divided into two classes: demand valuation of each resource to the buyer and bids of the sellers in each time step.

Firstly, we introduce the generation of demand valuation of each resource. Let λ be the job arrival rate of users, and μ the job process rate decided by resource capacity. Given job arrival rate λ, we consider the BCP whose valuation function is: $V(\mu) = V_0(\lambda) - c\cdot\lambda\frac{1}{\mu-\lambda}$, $\mu > \lambda$. Assume the original job process rate using home resources is μ_0. In our simulation, one resource is corresponding to a group of Virtual Machines which can improve unit of process rate. Thus, we can improve the process rate to $\mu_0 + k$ if we purchase k resources, the valuation function becomes $V(k) = V_0(\lambda) - c\cdot\lambda\frac{1}{\mu_0+k-\lambda}$, $\mu_0 + k > \lambda$. The marginal demand valuation of the k-th resource $v(k)$ is the first derivative of the valuation $V(k)$ expressed by following equation:

$$v(k) = \frac{c\cdot\lambda}{(\mu_0+k-\lambda)^2} \tag{2}$$

The marginal demand valuation of each resource is monotonically decreasing which is shown in Fig. 3. At the same time, Fig. 3 also shows that the valuation of each resource is related with the current arrival rate λ. The valuation of j-th resource is monotonically increasing with current arrival rate.

To generate the demand valuation of each resource, according to equation (2) we need to obtain arrival rate λ of each time. We used the real data from two parallel computing centers [30]. The unit of time is one day. The fluctuation of the workloads can be shown in Fig. 4(a) and Fig. 4(b). Fig. 4(a) shows the distribution of job arrival rates (the average number of arrival jobs per hour in one day) from May to August 2014, the data of which are from the national grid of the Czech republic, called MetaCentrum. Fig. 4(b) shows the distribution of job arrival rates during the same time internal as Fig. 4(a), the data of which are from the Gaia cluster at the University of Luxemburg. We construct 5 data settings by selecting parts of data from above parallel computing centers. As shown in Table 2, DS1 and DS2 are generated by the job arrival rates of the first 32 days and 64 days from MetaCentrum; DS3, DS4 and DS5 are generated by the job arrival rates of 1–64 days and



(a) Job arrival rate of MetaCentrum (b) Job arrival rate of Gaia cluster

(c) Job arrival rate generated randomly

Fig. 4. The selected workload data of the BCP in simulation, expressed by job arrival rates. The first one is from workload logs of the Gaia cluster at the University of Luxemburg, the second is from workload logs of the national grid of the Czech republic, both of them are selected during the period from May to August 2014. The last one is generated randomly and uniform distributed in interval of [1,1000].

21–85 days from Gaia cluster. In addition, to evaluate the performance adequately, we also construct the 6-th data setting (DS6), by generating job arrival rates randomly with uniform distribution in [1,1000] which is shown in Fig. 4(c). The main characteristics of those data settings are as follows:

- In order to analyze the effect of the length of period on the results, we select two time intervals as the length of demand period: 32 days in DS1 and 64 days in DS2, and both of the data are from MetaCentru.
- In DS1, DS2, DS3, the workloads during whole demand period are relatively balanced and balance degree $\delta < 3$.
- In DS4, we can see from Fig. 4(b), the job arrival rates during the first 1–32 days are far less than that during 33-64 days which means that the whole demands of the first half of the time T is far lower than that of the second half of the time T.
- To generate another case, we modified the data in DS4 by exchanging the job arrival rates of the first 32 days and the last 32 days. Therefore, we obtain the data setting DS5 that the whole demands of the first half of the time T is far higher than that of the second half of the time T.

In the above 6 data settings, we respectively change the budgets from 200 to 2000 by increasing 200 each time. So there are 60 scenarios we perform the mechanisms.

We set original process rate $\mu_0 = \frac{1.5\sum_{i=1}^{|T|}\lambda_i}{|T|}$ in DS1, DS2, DS3, DS6. Since the workloads in DS4 and DS5 are significant unbalanced, we decrease the original process speed to make the BCP be in peak-demand during whole time period T by setting $\mu_0 = \frac{0.5\sum_{i=1}^{|T|}\lambda_i}{|T|}$ in DS4, DS5.

Table 2
Data settings used in simulation (six data settings are constructed based on the workload data shown in Fig. 4).

Data Setting	Data Source	Days	B	μ_0
DS1	MetaCentrum	1–32	200–2000	$\frac{1.5\sum_{i=1}^{\|T\|}\lambda_i}{\|T\|}$
DS2	MetaCentrum	1–64	200–2000	$\frac{1.5\sum_{i=1}^{\|T\|}\lambda_i}{\|T\|}$
DS3	Gaia cluster	1–64	200–2000	$\frac{1.5\sum_{i=1}^{\|T\|}\lambda_i}{\|T\|}$
DS4	Gaia cluster	21–85	200–2000	$\frac{0.5\sum_{i=1}^{\|T\|}\lambda_i}{\|T\|}$
DS5	Gaia cluster	53–85, 21–52	200–2000	$\frac{0.5\sum_{i=1}^{\|T\|}\lambda_i}{\|T\|}$
DS6	Generated Randomly	1–64	200–2000	$\frac{1.5\sum_{i=1}^{\|T\|}\lambda_i}{\|T\|}$

In the sellers' side, there are 20 ICPs who can provide idle resources with private costs. We assume that, at each time there always exist sufficient idle resources in the cloud network to satisfy the demands of the BCP, and we consider that the costs of ICPs are independent identically distributed and subject to some known distributions. In the experiments, we generate the bids of ICPs in each time unit with uniform distribution in $[b_{min}, b_{max}]$, and we choose $b_{min} = 1$ and $b_{max} = 5$. To reduce the impact of randomness, the valuation we compared is the average valuation of 50 experiments.

In addition, although we have obtained a competitive ratio in Corollary 5.1 with the assumption of known δ, in most cases we only can estimate the value of δ. In our experiments, we introduce two value of balance degree δ: real value denoted by δ^* and estimated value denoted by δ^e. The real balance degree δ^* is obtained according to $\frac{\max\{V_{T_1}, V_{T_2}\}}{\min\{V_{T_1}, V_{T_2}\}}$, where, V_{T_1}, V_{T_2} are obtained respectively from the first and second half of time T, $T = [t_s, t_e]$ by offline Algorithm 2; the estimated balance degree δ^e is obtained only by the sampling period T_s, expressed by $\frac{\max\{V_{T_{s1}}, V_{T_{s2}}\}}{\min\{V_{T_{s1}}, V_{T_{s2}}\}}$, where, $V_{T_{s1}}, V_{T_{s2}}$ are obtained from the first and second half of sampling period T_s by Algorithm 2, respectively.

6.2. Evaluation results

Fig. 5(a)–Fig. 5(e) show the simulation results. The value of Y-axis is the ratio of valuation obtained by the mechanisms to that by optimal allocation, where the mechanisms include OSPB with real balance degree δ^*, OSPB with estimate balance degree δ^e, proportional share and the allocation with average budget. The value of X-axis of the figures is the total available budget.

Firstly, we compare the performance of the mechanisms in DS1, DS2, DS3 which are three data settings with relatively balanced demands with real balance degree $\delta^* < 3$. Fig. 5(a)–Fig. 5(c) show that, regardless of the time interval length of 32 or 64, both OSPB with δ^* and δ^e have higher total valuation than allocation with average budget in most cases except the cases with lower budget in DS1. More interestingly, OSPB with estimated balance degree δ^e, has more significant effects than that of OSPB with δ^*. That is because, to obtain the competitive ratio, the worst case $\rho_2 = \rho^1/\delta^*$ should be considered in OSPB with real balance degree δ^* which causes the loss of part of valuation in many cases.

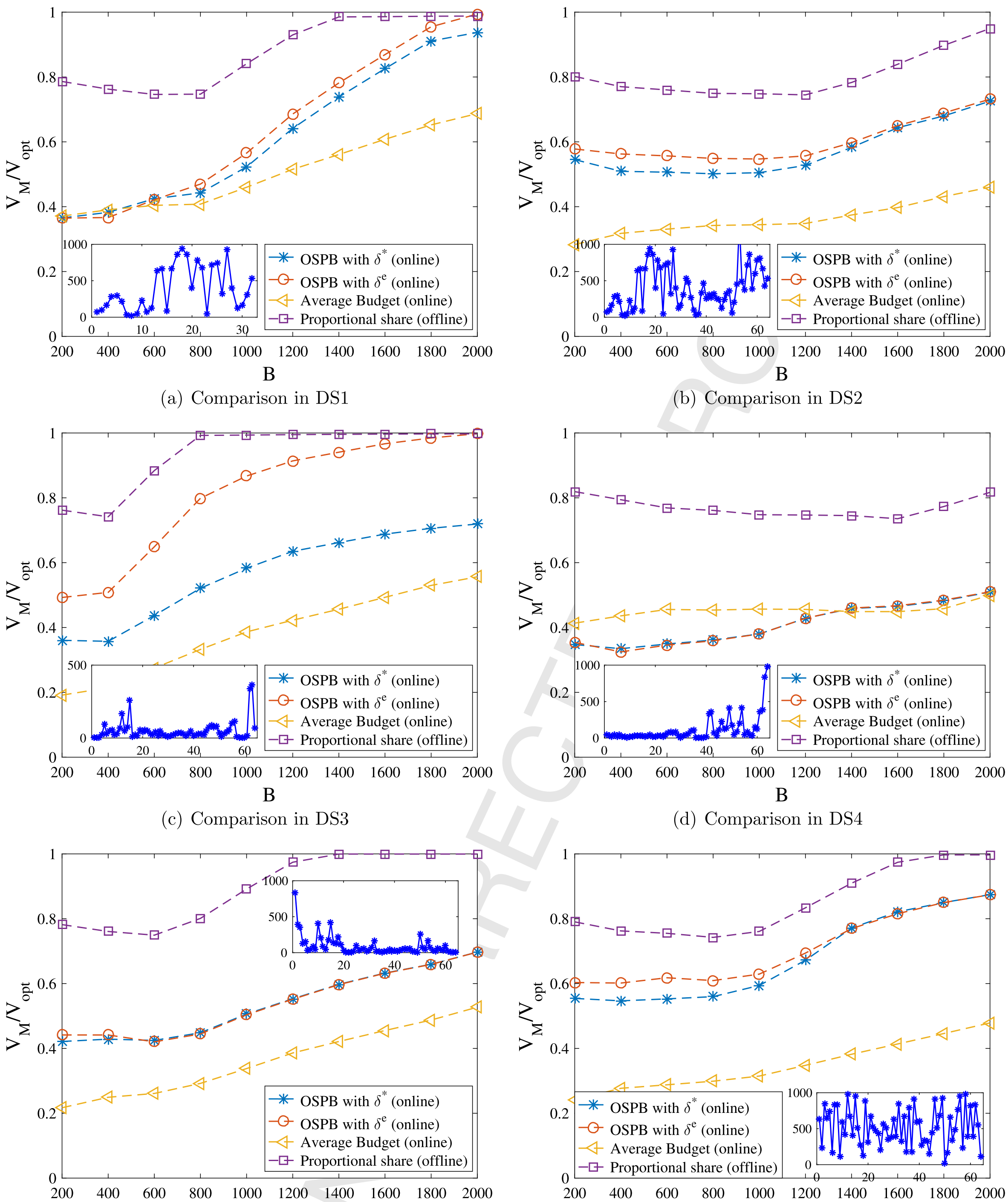
Secondly, we observe the performance of mechanisms in DS4. As described above, in the data setting DS4 the whole demands of the first half of the time T is far lower than that of the second half of the time T. Fig. 5(d) shows the results using data of 21–85 days from the parallel computing center Gaia cluster. As shown in Fig. 4(b), the workloads of 21–85 days are unbalanced, and the average workload in first half of the time is significantly lower than that of the second half of the time. Fig. 5(d) shows the experiment results of DS4. When the budget is less than 1400, the OSPB has lower valuation compared with allocation with average budget. It also implies that our mechanism has no significant effect in this case.

Thirdly, the results of DS5 are shown in Fig. 5(e). The workloads of the data setting DS5 also are unbanlanced. Contrary to DS4, the whole demands of the first half of the time T is far higher than that of the second half of the time T in DS5. We can see that, OSPB with δ^* and δ^e show better results than allocation with average budget. This results also are obtained in DS6 where the arrival rate of each day is generated randomly.

Through the comparison and analysis of the mechanisms in above 60 scenarios of six data settings, there are more than 85% scenarios in which OSPB mechanism has a better performance than allocation with average budget, and it improves more than 20% valuation on average for the buyer. Compared to the offline truthful mechanism, the total valuation obtained by our online mechanism (OSPB) is at least 40% of that obtained by offline proportional share mechanism. The valuation obtained by OSPB mechanism will be close to optimal allocation if increasing the budget. At the same time, since the valuation obtained by OSPB mechanism with estimate balance degree δ^e is not less than that by OSPB mechanism with real balance degree δ^* on average, we can use δ^e to substitute the δ in Algorithm 4 if the δ^* cannot be obtained in prior.

7. Conclusions and discussions

In this paper, we proposed an OSPB mechanism to deal with complex online resource procurement under uncertain demands, in which budget constraint and non-increasing marginal demand valuations at each time are considered. The



(a) Comparison in DS1 (b) Comparison in DS2

(c) Comparison in DS3 (d) Comparison in DS4

(e) Comparison in DS5 (f) Comparison in DS6

Fig. 5. Comparison of mechanisms. V_M/V_{opt} is the ratio of valuation obtained by the mechanisms to that by optimal allocation, B is the budget. The inside small figures are the workload distributions of corresponding data setting. We construct 60 scenarios in six data settings with different budgets, the results show that in more than 85% scenarios the OSPB mechanism with real balance degree δ^* has better performance than allocation with average budget, and it improves more than 20% valuation on average, even if we use the estimate balance degree δ^e.

Please cite this article in press as: J. Han et al., An online sequential procurement mechanism under uncertain demands in multi-cloud environment, Int. J. Approx. Reason. (2018), https://doi.org/10.1016/j.ijar.2018.09.006

proposed mechanism is designed based on a seller accepting-rule according to a value-density threshold which is learned dynamically from the historic information. The proposed mechanism not only purchases resources and allocates budget at each time step dynamically, but also determines the price that the BCP must pay for her requested resources. Given the condition that all the sellers are myopic, our mechanism provides incentives to the ICPs to reveal their truthful costs which can facilitate a healthy competition among idle cloud providers. We proved that OSPB is individually-rational, budget feasible and truthful if all the ICPs are myopic. Furthermore, we obtained the competitive ratio of the proposed mechanism and investigated the efficiency by performing extensive experiments. The results show that in more than 85% scenarios the OSPB mechanism with balance degree δ has better performance than allocation with average budget, and it improves more than 20% valuation on average, even if we use the estimate value δ^e of balance degree.

There are several interesting open problems. In this work, we aim for maximizing the valuation of the BCP under a budget constraint. One of the future works is to maximize the profit of the BCP at the similar scenario. Another interesting work is to investigate the online procurement policy for minimizing the regret of the BCP, where the bids of the ICPs subject to unknown some distribution.

References

[1] M. Liaqat, V. Chang, A. Gani, S.H.A. Hamid, M. Toseef, U. Shoaib, R.L. Ali, Federated cloud resource management: review and discussion, J. Netw. Comput. Appl. 77 (2016) 87–105.
[2] M. Babaioff, Y. Mansour, N. Nisan, G. Noti, C. Curino, N. Ganapathy, I. Menache, O. Reingold, M. Tennenholtz, E. Timnat, ERA: a framework for economic resource allocation for the cloud, CoRR abs/1702.07311 (2017), http://arxiv.org/abs/1702.07311.
[3] B. Javed, P. Bloodsworth, R.U. Rasool, K. Munir, O.F. Rana, Cloud market maker: an automated dynamic pricing marketplace for cloud users, Future Gener. Comput. Syst. 54 (2016) 52–67.
[4] W. Shi, C. Wu, Z. Li, Rsmoa: a revenue and social welfare maximizing online auction for dynamic cloud resource provisioning, in: International Workshop on Quality of Service, 2014, pp. 41–50.
[5] J. Altmann, M.M. Kashef, Cost model based service placement in federated hybrid clouds, Future Gener. Comput. Syst. 41 (2014) 79–90.
[6] R. Buyya, R. Buyya, Budget-Driven Scheduling of Scientific Workflows in IaaS Clouds with Fine-Grained Billing Periods, vol. 12, ACM Transactions on Autonomous and Adaptive Systems, 2017.
[7] K. Rzadca, A. Datta, G. Kreitz, S. Buchegger, Game-theoretic mechanisms to increase data availability in decentralized storage systems, ACM Trans. Auton. Adapt. Syst. 10 (3) (2015) 1–32.
[8] S. Singh, I. Chana, Cloud resource provisioning: survey, status and future research directions, Knowl. Inf. Syst. 49 (3) (2016) 1005–1069.
[9] M.M. Hassan, M.S. Hossain, J. Sarkar, E.N. Huh, Cooperative game-based distributed resource allocation in horizontal dynamic cloud federation platform, Inf. Syst. Front. 16 (4) (2012) 523–542.
[10] G.D. Modica, G. Petralia, O. Tomarchio, Procurement auctions to trade computing capacity in the cloud, in: Eighth International Conference on P2p, Parallel, Grid, Cloud and Internet Computing, 2013, pp. 298–305.
[11] V.G. Prasad, A.S. Prasad, S. Rao, A mechanism design approach to resource procurement in cloud computing, IEEE Trans. Comput. 63 (1) (2014) 17–30.
[12] P. Bonacquisto, G.D. Modica, G. Petralia, O. Tomarchio, A procurement auction market to trade residual cloud computing capacity, IEEE Trans. Cloud Comput. 3 (3) (2015) 345–357.
[13] J. Zhao, X. Chu, H. Liu, Y.W. Leung, Online procurement auctions for resource pooling in client-assisted cloud storage systems, in: Computer Communications, 2015, pp. 576–584.
[14] Q. Wang, K. Ren, X. Meng, When cloud meets ebay: towards effective pricing for cloud computing, in: International Conference on Computer Communications, IEEE, 2012, pp. 936–944.
[15] C. Wang, W. Ma, T. Qin, X. Chen, X. Hu, T. Liu, Selling reserved instances in cloud computing, in: International Joint Conference on Artificial Intelligence, 2015, pp. 224–230.
[16] A.N. Toosi, K. Vanmechelen, F. Khodadadi, R. Buyya, An auction mechanism for cloud spot markets, ACM Trans. Auton. Adapt. Syst. 11 (1) (2016) 1–33.
[17] V.G. Prasad, A.S. Prasad, S. Rao, A combinatorial auction mechanism for multiple resource procurement in cloud computing, IEEE Trans. Cloud Comput. (2017) 337–344.
[18] N. Samaan, A novel economic sharing model in a federation of selfish cloud providers, IEEE Trans. Parallel Distrib. Syst. 25 (1) (2014) 12–21, https://doi.org/10.1109/TPDS.2013.23.
[19] L. Mashayekhy, M.M. Nejad, D. Grosu, Cloud federations in the sky: formation game and mechanism, IEEE Trans. Cloud Comput. 3 (2015) 14–27.
[20] Y. Singer, Budget feasible mechanisms, in: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, 2010, pp. 765–774.
[21] N. Chen, N. Gravin, P. Lu, On the approximability of budget feasible mechanisms, in: ACM-SIAM Symposium on Discrete Algorithms (SODA), 2011, pp. 685–699.
[22] S. Dobzinski, C.H. Papadimitriou, Y. Singer, Mechanisms for complement-free procurement, in: ACM Conference on Electronic Commerc, 2011, pp. 273–282.
[23] X. Bei, N. Chen, N. Gravin, P. Lu, Budget feasible mechanism design: from prior-free to bayesian, in: Proceedings of the Annual Acm Symposium on Theory of Computing, 2012, pp. 449–458.
[24] Y. Singer, M. Mittal, Pricing mechanisms for crowdsourcing markets, in: International World Wide Web Conferences, 2013, pp. 1157–1166.
[25] A. Badanidiyuru, R. Kleinberg, Y. Singer, Learning on a budget: posted price mechanisms for online procurement, in: ACM Conference on Electronic Commerce, 2012, pp. 128–145.
[26] D. Zhao, X.Y. Li, H. Ma, How to crowdsource tasks truthfully without sacrificing utility: online incentive mechanisms with budget constraint, in: IEEE Conference on Computer Communications, 2014, pp. 1213–1221.
[27] H. Mendelson, Pricing computer services: queueing effects, Commun. ACM 28 (3) (1985) 312–321.
[28] J.H. Jhang-Li, I.R. Chiang, Resource allocation and revenue optimization for cloud service providers, Decis. Support Syst. 77 (2015) 55–66.
[29] R.B. Myerson, Optimal auction design, Math. Oper. Res. 6 (1) (1978) 58–73.
[30] P. workloads archive, Logs of real parallel workloads from production systems, http://www.cs.huji.ac.il/labs/parallel/workload/logs.html, 2015.

Please cite this article in press as: J. Han et al., An online sequential procurement mechanism under uncertain demands in multi-cloud environment, Int. J. Approx. Reason. (2018), https://doi.org/10.1016/j.ijar.2018.09.006