Optimal Posted Prices for Online Cloud Resource Allocation

ZIJUN ZHANG, University of Calgary
ZONGPENG LI, University of Calgary
CHUAN WU, The University of Hong Kong

We study online resource allocation in a cloud computing platform through *posted pricing*: The cloud provider publishes a unit price for each resource type, which may vary over time; upon arrival at the cloud system, a cloud user either takes the current prices, renting resources to execute its job, or refuses the prices without running its job there. We design pricing functions based on current resource utilization ratios, in a wide array of demand-supply relationships and resource occupation durations, and prove worst-case competitive ratios in social welfare. In the basic case of a single-type, non-recycled resource (allocated resources are not later released for reuse), we prove that our pricing function design is *optimal*, in that it achieves the smallest competitive ratio among all possible pricing functions. Insights obtained from the basic case are then used to generalize the pricing functions to more realistic cloud systems with multiple types of resources, where a job occupies allocated resources for a number of time slots till completion, upon which time the resources are returned to the cloud resource pool.

Additional Key Words and Phrases: Cloud Computing; Posted Pricing; Resource Allocation; Online Algorithms; Competitive Analysis

ACM Reference format:
Zijun Zhang, Zongpeng Li, and Chuan Wu. 2017. Optimal Posted Prices for Online Cloud Resource Allocation. *Proc. ACM Meas. Anal. Comput. Syst.* 1, 1, Article 22 (June 2017), 26 pages.
https://doi.org/http://dx.doi.org/10.1145/3084460

1 INTRODUCTION

Over the past decade, cloud computing has proliferated as the new computing paradigm that provides flexible, on-demand computing services in a pay-as-you-go fashion. Various applications and systems are built upon cloud computing models, including big data analytics, cloud radio access networks (C-RAN), network function virtualization (NFV), to name a few. Despite the common illusion that a cloud consists of an unlimited ‘sea’ of resources, real-world clouds are constrained by finite system capacity [17, 24] (*e.g.*, physical capacity of a cloud data center), which may become tight in periods of peak demand [9, 12]. A fundamental problem in cloud computing is cloud resource allocation, *i.e.*, to determine which user demands to satisfy at each time point. A natural goal is to maximize the social welfare of the cloud eco-system, which represents the aggregated ‘happiness’ of the cloud provider and the cloud users [5].

This work is supported by the National Natural Science Foundation of China (NSFC), under grant 61628209; and by the Research Grants Council (RGC) of Hong Kong, under grants HKU 717812, 718513, 17204715, 17225516, C7036-15G (CRF).

Author’s addresses: Z. Zhang and Z. Li, Department of Computer Science, University of Calgary; C. Wu, Department of Computer Science, the University of Hong Kong.

2476-1249/2017/6-ART22 $15.00
https://doi.org/http://dx.doi.org/10.1145/3084460

Cloud resource allocation in practice exhibits a nature of *online decision making*: cloud users with job requests arrive at the cloud system at arbitrary time points, and the cloud provider decides resource allocation upon each arrival. A natural, *de facto* standard of cloud resource allocation is through a *posted pricing mechanism*: the cloud provider publishes resource prices; cloud users act as price takers who utilize the resources if the prices are acceptable (*i.e.*, its valuation of the job exceeds the cost of resource renting), and will otherwise give up the cloud service.

Major cloud providers today, such as Amazon Web Services, Microsoft Azure, and Google Cloud, typically adopt fixed prices, *i.e.*, resource usage is charged at fixed unit prices posted on their websites. However, a dynamic pricing strategy based on realtime demand-supply is more efficient in many scenarios [4], to fully exploit the resource capacity of a cloud system, and to better satisfy user demands. For practical cloud computing systems that employ dynamic pricing strategies, *e.g.*, Amazon EC2 Spot Instances [1], short-term prices may not be driven by realtime demand-supply [3]; however, the price differences across different service regions and over different time periods are still relevant to demand and supply. Inspired by the Spot Instances model, various dynamic pricing strategies have been proposed in recent literature, including auction mechanisms [13, 15, 23, 25, 27, 29, 30], and other dynamic pricing strategies for revenue maximization and efficient cloud resource utilization [14, 19, 26].

This work studies effective pricing functions for a cloud provider to employ, for computing unit resource prices at each time point. The computed prices are posted as ‘take it or leave it’ prices for cloud users to decide whether to rent the cloud resources, while users’ job valuations are not revealed to the cloud provider. Such prices can also serve in a posted-price auction mechanism for cloud job admission and charging. With meticulously designed online prices, our goal is to maximize the social welfare of the cloud, which equals the overall valuation of executed user jobs, minus a possible operational cost, over the entire system span.

Both social welfare maximization and provider revenue maximization are possible goals in cloud resource allocation [16, 20]. Social welfare represents the aggregate gain of the cloud provider and cloud users, indicating overall system efficiency. Compared to maximizing provider revenue, maximizing social welfare ensures good user experience, which is important for long-term market competitiveness of a cloud provider [28]. Furthermore, for public clouds operated by nonprofit organizations, and private clouds serving internal jobs, maximizing social welfare is more relevant than maximizing revenue [18]. In these cases, the pricing schemes studied in this paper can be used as mechanisms for allocating cloud resources to users based on their urgency and priorities. In the auction design literature, there further exist techniques that can relate social welfare maximizing mechanisms with revenue maximizing mechanisms [10].

Our study of the pricing functions has been partly inspired by dual price design in competitive online algorithms based on the classic primal-dual framework [6, 8]. In primal-dual online algorithm design, a key idea is to update dual prices using exponential functions for making primal resource allocation decisions, leading to provable competitive ratios. Nonetheless, no explicit justification exists in the literature on the choice of using *exponential* dual price functions.

In this work, we borrow the exponential form of the pricing function from the literature of primal-dual online algorithms, and propose the optimal form of the exponential pricing functions for a fundamental cloud resource allocation problem. We then provide an intuitive explanation of the optimality of the exponential pricing function. For the first time in the literature, we generalize the pricing function to scenarios with bounded total demand, where the optimal form is no longer necessarily an exponential function. Interestingly, this result also contributes to the literature on knapsack problems, in that our problem is closely related to a variant of the online knapsack problem [11], where the total weight of items is upper bounded.

We start by investigating the basic case of a single type of cloud resource without resource recycling, and design resource pricing functions based on the current resource utilization levels that capture realtime demand-supply of cloud resources. We prove the optimality of our pricing function design. We then investigate the cases of multiple resource types, and limited resource occupation durations. Our contributions are summarized below.

First, we justify the use of exponential pricing functions in the literature of both cloud computing [13, 21, 22, 29, 30] and online algorithms [6, 8], both from a theoretical point of view and with intuitive interpretation. We prove the optimality of the pricing function under mild system assumptions that are standard in recent literature.

Second, we derive the optimal pricing functions for more realistic cloud resource allocation scenarios, where the potential total demand for resources is bounded.

Third, we extend the pricing functions to take into account multiple resource types. We propose a joint pricing and scheduling strategy when the cloud system runs over multiple time slots. We prove tight competitive ratios for these scenarios, which were not properly proven in previous literature. We make no assumptions on the arrival process and the distribution of user valuations.

We further verify effectiveness of our price design in realistic cloud computing scenarios through simulation studies, with assumptions in theoretical analysis relaxed. We show that the parameters involved in our pricing functions can be practically optimized in different scenarios, to achieve consistently good performance ratios against the offline optimal social welfare.

Finally, we note that our pricing models and algorithms are generally applicable to posted pricing mechanism design in other online resource allocation systems, which share similar characteristics as a cloud computing system.

In the rest of the paper, we review related literature in Sec. 2. The basic and general models of cloud resource pricing are studied in Sec. 3 and Sec. 4, respectively. Sec. 5 presents simulation studies, and Sec. 6 concludes the paper.

2 RELATED WORK

Recently, auction mechanisms have been extensively studied for online cloud resource allocation and pricing. Zhang et al. [29] design an online auction mechanism for IaaS clouds, aiming to maximize both social welfare and provider profit. Zhou et al. [30] extend the auction mechanism to deal with computing jobs with soft deadlines. Shi et al. [22] propose an online mechanism for virtual cluster allocation and pricing. These studies exploit the primal-dual framework for online mechanism design, and adopt exponential pricing functions to compute dual prices, for deciding resource allocation and user payments. Competitive ratios of the online mechanisms are proven, but the rational of adopting exponential pricing functions is lacking, and the optimality of such exponential functions are not studied. Indeed, a wide spectrum of increasing functions are conceivable for cloud resource pricing. Our pricing functions are applicable to both posted pricing mechanisms and online auctions. The analysis of optimality of our pricing functions is independent from the primal-dual framework.

Apart from auction mechanisms, a wide range of resource pricing schemes have been studied in the literature. While static pricing is prevalent in today's cloud market, dynamic pricing based on realtime demand-supply can be more efficient in many scenarios [4]. Li et al. [14] design a pricing algorithm for cloud resources, which updates current prices based on historical resource utilization ratios. Their experiment demonstrates the advantage of the pricing algorithm in terms of cost reduction and efficient resource allocation. Mihailescu and Teo [19] propose a dynamic pricing scheme for federated clouds, where different cloud providers share and trade resources for enhanced scalability and reliability. They show that user welfare and the percentage of successful requests are increased by dynamic pricing, as compared to fixed pricing. The pricing schemes developed in this work are both dynamic and usage-based, *i.e.*, the unit price of cloud resource is driven by demand-supply dynamics, and the total price is proportional to the amount and service time of requested resources.

The online social welfare maximization problem studied in this work is related to a variant of the online knapsack problem [11]. Two assumptions are made in this literature: the weight of each item is much smaller than the capacity of the knapsack, and the density (value to weight ratio) of every item falls in a known range $[L, U]$. Under these assumptions, Buchbinder and Naor [6, 7] design an algorithm achieving a competitive ratio of $O(\log(U/L))$, as well as an $\Omega(\log(U/L))$ lower bound on the competitive ratio of any algorithm. In the context

Table 1. Notation and definition

$\mathcal{U}$	set of users
$\mathcal{R}$	set of resource types
$\mathcal{T}$	set of all time slots
$\mathcal{T}_i$	set of time slots required by user i
$d_{i,r}$	amount of resource r demanded by user i
d_i	total amount of resource demanded by user i
v_i	value of successfully finishing user i's job
p	unit resource price at the time of user arrival
$\underline{p}/\overline{p}$	lower/upper bound of v_i/d_i
γ	ratio between $\overline{p}$ and $\underline{p}$
ρ	resource utilization level
${\rho_r}^*$	final resource utilization level as defined by Definition 3.3
β	scarcity level as defined by Definition 3.1
$V_{ol}\left(\rho^*\right)$	total value obtained by an online solution, given a final utilization level ρ^*
$V_{opt}\left(\rho^*\right)$	total value obtained by an optimal offline solution, given a final utilization level ρ^*

of advertising auctions, Zhou et al. [31] design a $(\log(U/L) + 1)$-competitive algorithm for an online knapsack problem under the above assumptions. Interestingly, their algorithm is equivalent to our proposed pricing strategy for the most basic case, as will be discussed in Sec. 3.2.1. Nevertheless, our proof of optimality is different from that given by Zhou et al. [31], and leads to an intuitive interpretation on the choice of exponential pricing functions. More importantly, the total weight of items is assumed to be unbounded in the previous work, which is impractical in real-world applications. In this work, we develop a more general pricing strategy that achieves better competitive ratios for bounded total weight, and prove the optimality of the proposed strategy.

3 CLOUD RESOURCE PRICING: THE BASIC CASE

We start by designing pricing functions for a basic, yet fundamental version of the online resource allocation problem, following the posted pricing framework as described in Algorithm 1. By analyzing the basic case of allocating a single-type, non-recycled resource, we develop necessary techniques and theoretical results, for the online pricing and scheduling of more realistic cloud resource allocation problems.

3.1 The Basic Resource Allocation Problem

Consider a cloud provider whose data center is for now assumed to provision a single type of resource, to be allocated to a large number of cloud users. The users in a set $\mathcal{U}$ come in an arbitrary sequence. Upon arrival, a user decides immediately whether to rent cloud resources, by comparing the valuation of its job with the overall price of required resources for executing the job. Let d_i denote the amount of resource demanded by a user $i \in \mathcal{U}$, and v_i be the value of successfully finishing i's job. In practice, v_i is often influenced by multiple factors, such as the purpose and priority of the job, and what is gained from the job's completion. Without loss of generality, we normalize user resource demands, assuming the total amount of resource in the cloud is 1, so that d_i can be considered as the proportion of the entire resource pool demanded by user i. Let p be the unit price of the resource posted by the cloud provider, which may vary over time. A user i accepts the price and rents resource at quantity d_i, if and only if $v_i \geq d_i p$, where p is the current unit resource price at the time of user arrival. Effectively, v_i

ALGORITHM 1: Online pricing and resource allocation

```
Input: $d_i, v_i, \forall i \in \mathcal{U}$
Output: $x_i, \forall i \in \mathcal{U}$
$\rho = 0$ ;                                        // Initialize the resource utilization
for $i \in \mathcal{U}$ do
    /* Upon the arrival of each user $i$                                            */
    if $v_i \geq d_i P(\rho)$ and $\rho + d_i \leq 1$ then
        /* User $i$ accepts the posted price                                        */
        $x_i = 1$;
        $\rho = \rho + d_i$ ;                                // Allocate resource to user $i$
    else
        /* User $i$ rejects the posted price                                        */
        $x_i = 0$;
```

simply serves as a threshold for a price to be acceptable to user i. In this section, we assume that each unit of the resource, once allocated, will not be returned to the resource pool.

The utility of the cloud provider is the total payment received. The utility of a served user is the valuation of its job minus its payment. The utility of an unserved user is zero. Since payments cancel themselves, the social welfare of the entire cloud system is equivalent to the total valuation of served jobs, assuming no operational cost of the cloud.

Let x_i indicate whether user i rents resource (at quantity d_i) or not upon its arrival. The social welfare maximization problem can be formulated as an integer linear program (ILP):

$$\text{maximize} \quad \sum_{i \in \mathcal{U}} v_i x_i \tag{1}$$

s.t.:

$$\sum_{i \in \mathcal{U}} d_i x_i \leq 1 \tag{1a}$$

$$x_i \in \{0, 1\}, \forall i \in \mathcal{U} \tag{1b}$$

This is a 0-1 knapsack problem, and can be solved to optimum using dynamic programming in the offline setting. However, for the online problem we investigate, the columns of the coefficient matrix of constraint (1a), corresponding to different user arrivals, are revealed one-by-one, while the value of x_i is to be determined immediately when a user comes to the cloud. We apply an online resource allocation algorithm, as shown in Algorithm 1, to decide resource allocation given resource prices.

The performance of the posted pricing mechanisms in the online resource allocation algorithm clearly depends on the pricing function. We do not assume that users reveal their job valuations to the cloud provider. Consequently, the pricing strategy depends only on the demand-supply relationship of cloud resources. To evaluate the quality of a resource allocation solution, we resort to the standard notion of *competitive ratio*, defined as the ratio between the optimal objective value of the offline problem (1) and that of the online solution. The smaller (closer to 1) the competitive ratio is, the better the online resource allocation solution. We will focus on the *worst-case* competitive ratio, as opposed to the average-case competitive ratio. We first make the following two mild assumptions:

ASSUMPTION 1. *The variability of users' valuations is constrained,* i.e., $\underline{p} \leq v_i/d_i \leq \overline{p}, \forall i \in \mathcal{U}$*, where* $\underline{p}$ *and* $\overline{p}$ *are lower bound and upper bound of the per-unit-resource job valuation of all users, respectively.*

ASSUMPTION 2. *The resource demand of each user is much smaller than the total resource capacity,* i.e., $d_i \ll 1, \forall i \in \mathcal{U}$.

Assumption 2 is standard in the literature of online resource allocation [29, 30] and online knapsack problems [6, 7, 11, 31], and is reasonable in large-scale data centers. We make this assumption to facilitate theoretical analysis, such that techniques from calculus (differentiation) can be applied, and rare, extreme cases can be eliminated. For example, if a high-valued bid demanding almost all the resource from a cloud provider is rejected, because a small fraction of the resource is occupied by other users, then the worst-case competitive ratio can be arbitrarily high. Such an assumption.

Nonetheless, it is possible to relax Assumption 2 to specifying an upper bound on d_i instead, without significantly affecting our theoretical result. Specifically, differentiation and integration can be replaced with differences and summation, to derive similar results. We will relax this assumption in empirical studies later in the paper.

3.2 Pricing Function Design

We design pricing functions that adjust resource prices based on realtime demand-supply. It is helpful to have some prior knowledge about the total resource demand. In practice, unlimited total resource demand is rare; an estimated upper bound on the overall resource demand can often be obtained. This is reflected through the following definition.

Definition 3.1. Suppose the total resource demand of all users is upper bounded by $1 + \beta$ times the total resource supply, *i.e.*, $\sum_{i \in \mathcal{U}} d_i \leq 1 + \beta$, with $\beta > -1$. We refer to β as the *scarcity level* of the resource.

It is possible to have a known lower bound on the overall resource demand as well, but our algorithm design and analysis do not rely on such a lower bound.

We next present the optimal pricing function for $\beta \longrightarrow \infty$, and then derive the optimal pricing functions for finite β, based on the insight we gain from the analysis of the first case. We then further show in Sec. 3.3 that the results can be extended to the case with linear operational costs of cloud resources.

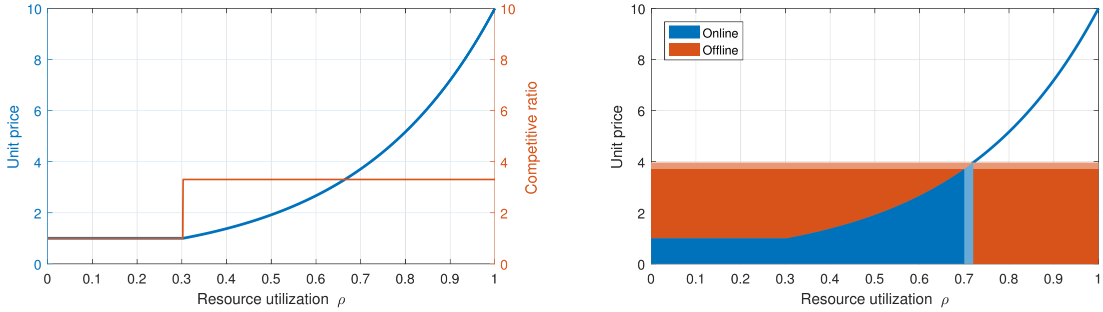
3.2.1 Pricing Function for Large Total Demand. We begin with the case that the total demand for resource is much larger than the capacity of the cloud resource pool. We propose an optimal pricing function for the case that $\beta \longrightarrow \infty$, and then show the same pricing function is in fact optimal as long as $\beta \geq 1$ (*i.e.*, the overall resource demand is at least twice of the resource capacity).

Definition 3.2. In Algorithm 1, oblivious of true valuations of users, a pricing function is *optimal* if it achieves the smallest possible worst-case competitive ratio in social welfare under Assumptions 1 and 2.

Let ρ be the resource utilization level, *i.e.*, the amount of the resource already allocated. Note that ρ is a function of time, but this dependency is omitted for notational simplicity. The unit price of the resource at the respective resource utilization level is denoted by $P_1(\rho)$, designed as follows:

$$P_1(\rho) = \begin{cases} \underline{p}, & \rho \in [0, 1/(\log \gamma + 1)] \\ \underline{p} e^{(\log \gamma + 1)\rho - 1}, & \rho \in (1/(\log \gamma + 1), 1), \\ +\infty, & \rho = 1 \end{cases} \tag{2}$$

where $\gamma = \overline{p}/\underline{p}$. An illustration of the pricing function for $\underline{p} = 1, \overline{p} = 10$ is given in Fig. 1 (blue lines in both subfigures). Intuitively, when ρ is quite small, it is desirable to keep the price at the lowest level ($\underline{p}$), to allow all potential users to rent the resource. As ρ increases, the amount of satisfied demand increases, as well as the obtained social welfare, and hence it is reasonable to raise price to filter out users with low valuations. When $\rho = 1$, cloud resource is exhausted, so we use an infinitely high price to reject all subsequent users. Note that even



(a) The pricing function and competitive ratio.

(b) The worst-case $V_{ol}(\rho^*)$ and $V_{opt}(\rho^*)$ for $\rho^* = 0.7$ visualized by AUCs. Each AUC indicates the total value obtained by an online or offline solution.

Fig. 1. An illustration of pricing function (2) for $\underline{p} = 1, \overline{p} = 10$.

if we need the lower bound and upper bound of the per-unit-resource valuation in (2), when applying this pricing function in online resource allocation, we can use estimates of the bounds, which can be further calibrated over time as more users arrive and more price taking decisions are learned.

We next prove the worst-case competitive ratio of Algorithm 1 when using the pricing function in (2), as well as the optimality of the pricing function when $\beta \to \infty$ (this default condition omitted in all lemmas, claims and theorems before Theorem 3.8), and then generalize the conclusion to the case $\beta \geq 1$ in Theorem 3.8.

Definition 3.3. $\rho^* \in [0, 1]$ denotes the *final* utilization level of the resource after all users have decided whether to rent the cloud resource to execute their jobs.

The following lemma implies that when the final resource utilization level is low, the total demand of potential users also tends to be low, thus it is possible to satisfy all user demands online.

Lemma 3.4. *If* $\rho^* \in [0, 1/(\log\gamma + 1)]$, *the worst-case competitive ratio achieved by Algorithm 1 using the pricing function in* (2) *is* $\alpha_{1,1} = 1$.

Proof. According to the pricing function in (2), for $\rho \in [0, 1/(\log\gamma + 1)]$, the unit price is a constant, $\underline{p}$, which by Assumption 1 is acceptable to any potential user, thus $\rho^* \in [0, 1/(\log\gamma + 1)]$ implies that the total demand of all users is exactly ρ^*. The social welfare achieved by the pricing function in (2) is the total value of all users, which is also the maximum possible social welfare achieved by solving the offline problem (1). Therefore, the worst-case competitive ratio is 1. □

For a final utilization level ρ^*, let $V_{ol}(\rho^*)$ be the total value obtained by an online solution, and $V_{opt}(\rho^*)$ be that obtained by an optimal offline solution. Thus, in any worst case, the ratio $V_{opt}(\rho^*)/V_{ol}(\rho^*)$ is maximized.

Lemma 3.5. *If* $\rho^* \in (1/(\log\gamma + 1), 1]$, *the worst-case competitive ratio achieved by Algorithm 1 using the pricing function in* (2) *is* $\alpha_{1,2} = \log\gamma + 1$.

Proof. For any $\rho^* \in (1/(\log\gamma + 1), 1]$, the worst case of the online solution is that the valuations of satisfied users are the same as the prices they accept. By Assumption 2, the minimum total value of an online solution is

$$V_{ol}(\rho^*) = \int_0^{\rho^*} P_1(\rho)\, d\rho = \frac{\underline{p}}{\log\gamma + 1} e^{(\log\gamma+1)\rho^* - 1}, \tag{3}$$

as shown by the blue *area under the curve (AUC)* in Fig. 1b. At the same time, any unsatisfied user has a unit value smaller than $P_1(\rho^*)$, because otherwise ρ^* cannot be the final resource utilization. Hence in the worst case, there can be a set of unsatisfied users with a total demand of 1 (*i.e.*, $\sum_{i \in \mathcal{U}_{opt}} d_i = 1, \forall r \in \mathcal{R}$, where $\mathcal{U}_{opt}$ is the set of users chosen by the optimal offline solution), and each with a unit value of $P_1(\rho^*) - \epsilon_i$, where ϵ_i is an arbitrarily small positive number, such that the optimal offline solution is to satisfy their demands with all available resource. This yields the maximum optimal offline value given Eq. (3):

$$\begin{aligned} V_{opt}(\rho^*) &= \sum_{i \in \mathcal{U}_{opt}} d_i \left(p(\rho^*) - \epsilon_i\right) = p(\rho^*) - \epsilon \\ &= \underline{p} e^{(\log \gamma + 1)\rho^* - 1} - \epsilon, \end{aligned} \tag{4}$$

as shown by the red AUC (partially covered by the blue one) in Fig. 1b. Here, $\epsilon = \sum_{i \in \mathcal{U}_{opt}} \epsilon_i$, and hence can also be arbitrarily small. Note that, there can be a case which leads to a larger optimal offline total value, by increasing the online value corresponding to $\rho \in [0, \rho^*]$ (*i.e.*, the blue AUC in Fig. 1b) until it is large enough and becomes part of the optimal offline value. However, the online value will increase more than the optimal offline value does in this case, making it impossible to be a worst case. Therefore, the worst-case competitive ratio $\alpha_{1,2} = \sup_{\epsilon > 0} \frac{V_{opt}(\rho^*)}{V_{ol}(\rho^*)} = \log \gamma + 1, \forall \rho^* \in (1/(\log \gamma + 1), 1]$. □

An illustration of the worst-case competitive ratio at different final resource utilization levels is shown in Fig. 1a (red line).

Theorem 3.6. *The worst-case competitive ratio of Algorithm 1 using the pricing function in* (2) *is*

$$\alpha_1 = \log \gamma + 1. \tag{5}$$

Proof. The worst-case competitive ratio of the pricing function in (2) is the maximum possible competitive ratio for all $\rho^* \in [0, 1]$. Hence following Lemma 3.4 and 3.5, $\alpha_1 = \max \left\{\alpha_{1,1}, \alpha_{1,2}\right\} = \log \gamma + 1$. □

We next show the optimality of the pricing function based on the observation that, to achieve a finite worst-case competitive ratio, any pricing function should contain a constant ($\underline{p}$) part at the beginning of the function.

Claim 3.1. *If a pricing function $P(\rho)$ achieves a finite worst-case competitive ratio of α, then $P(\rho) = \underline{p}, \forall \rho \in [0, 1/\alpha]$.*

Proof. If the claim does not hold and $P(0) > \underline{p}$, there can be a case where $\rho^* = 0$, such that the online total value $V'_{ol}(\rho^*) = 0$, while the optimal offline total value $V'_{opt}(\rho^*) = P(0) - \epsilon > 0$, where ϵ is an arbitrarily small positive number. Thus the worst-case competitive ratio $\alpha \geq \sup_{\epsilon > 0} \frac{V'_{opt}(\rho^*)}{V'_{ol}(\rho^*)} = +\infty$, which contradicts the assumption that α is finite.

If the claim does not hold and $P(0) = \underline{p}$, there must be a $\rho_0 \in (0, 1/\alpha]$ such that $P(\rho_0) > P(\rho), \forall \rho \in [0, \rho_0)$. There can be a case where $\rho^* = \rho_0$, such that the online total value

$$V'_{ol}(\rho^*) = \int_0^{\rho_0} P(\rho)\, d\rho < \rho_0 P(\rho_0),$$

while the optimal offline total value $V'_{opt}(\rho^*) = P(\rho_0) - \epsilon$. Thus the worst-case competitive ratio $\alpha \geq \sup_{\epsilon > 0} \frac{V'_{opt}(\rho^*)}{V'_{ol}(\rho^*)} > 1/\rho_0$, which contradicts $\rho_0 \leq 1/\alpha$. □

Theorem 3.7. *the pricing function in* (2) *is optimal according to Definition 3.2,* i.e., *using it Algorithm 1 achieves the smallest worst-case competitive ratio.*

Proof. We prove this theorem by way of contradiction. Assume that there exists a pricing function, $P_1'(\rho)$, which achieves a worst-case competitive ratio $\alpha_1' < \alpha_1$. According to Claim 3.1 and Theorem 3.6, we have $P_1'(\rho) = \underline{p}, \forall \rho \in [0, 1/\alpha_1']$, and hence

$$\int_0^{1/\alpha_1'} P_1'(\rho)\, d\rho < \int_0^{1/\alpha_1'} P_1(\rho)\, d\rho,$$

where $P_1(\rho)$ is the pricing function in (2).

If there exists some $\rho \in (1/\alpha_1', 1)$ such that $P_1'(\rho) \geq P_1(\rho)$ we find the smallest one, and denote it by ρ_1. Then there can be a case where $\rho^* = \rho_1$, such that the online total value

$$V_{ol}'(\rho^*) = \int_0^{\rho_1} P_1'(\rho)\, d\rho < \int_0^{\rho_1} P_1(\rho)\, d\rho = V_{ol}(\rho^*),$$

while the optimal offline total value $V_{opt}'(\rho^*) = P_1'(\rho_1) - \epsilon \geq P_1(\rho_1) - \epsilon = V_{opt}(\rho^*)$, where ϵ is an arbitrarily small positive number. Thus the worst-case competitive ratio $\alpha_1' \geq \sup_{\epsilon>0} \frac{V_{opt}'(\rho^*)}{V_{ol}'(\rho^*)} > \sup_{\epsilon>0} \frac{V_{opt}(\rho^*)}{V_{ol}(\rho^*)} = \alpha_1$, contradicting the assumption $\alpha_1' < \alpha_1$. Therefore, $P_1'(\rho) < P_1(\rho), \forall \rho \in (1/\alpha_1', 1)$.

For $\rho^* = 1$, since $P_1'(1) \leq \overline{p}$ (a unit price higher than $\overline{p}$ will have all potential users rejected) is finite, we now have

$$V_{ol}'(\rho^*) = \int_0^{1} P_1'(\rho)\, d\rho < \int_0^{1} P_1(\rho)\, d\rho = V_{ol}(\rho^*).$$

However, as cloud resource is exhausted, subsequent users will not be served, regardless of their valuations. There can be a case where the optimal offline total value $V_{opt}'(\rho^*) = \overline{p} = V_{opt}(\rho^*)$. Thus the worst-case competitive ratio $\alpha_1' \geq \frac{V_{opt}'(\rho^*)}{V_{ol}'(\rho^*)} > \frac{V_{opt}(\rho^*)}{V_{ol}(\rho^*)} = \alpha_1$, contradicting the assumption that $\alpha_1' < \alpha_1$. □

We next generalize the optimality result for all $\beta \geq 1$.

Theorem 3.8. *For $\beta \geq 1$, the pricing function in* (2) *is optimal according to Definition 3.2, and the corresponding worst-case competitive ratio is α_1.*

Proof. For any possible input set of users, we can prune the users that can be satisfied by neither the online solution or the optimal offline solution, without affecting the online or offline social welfare, given a certain pricing function. Clearly, the resulting set of users has a total demand no greater than 2, which can also happen given any $\beta \geq 1$. Consequently, all the discussions above can be generalized to $\beta \geq 1$. □

The following property (which holds for all $\beta \geq 1$) is useful for guiding the design of pricing functions in more realistic cloud computing scenarios.

Property 1. *For the pricing function in* (2), *and any $\rho^* \in (1/\alpha_1, 1]$,* i.e., *the monotonically increasing part of $P_1(\rho)$, we have*

$$\sup_{\epsilon>0} V_{opt}(\rho^*) = \alpha_1 V_{ol}(\rho^*), \tag{6}$$

and hence

$$\frac{d \sup_{\epsilon>0} V_{opt}(\rho^*)}{d\rho^*} = \alpha_1 \frac{dV_{ol}(\rho^*)}{d\rho^*}, \tag{7}$$

and a constant (w.r.t. ρ^*) worst-case competitive ratio, α_1.

Proof. This a corollary that follows from Eq. (3), (4), Lemma 3.5 and Theorem 3.5. □

Property 1 is illustrated in Fig. 1b, where the light red area corresponds to $\frac{dV_{opt}(\rho^*)}{d\rho^*}\Big|_{\rho^*=0.7}$, and the light blue area corresponds to $\frac{dV_{ol}(\rho^*)}{d\rho^*}\Big|_{\rho^*=0.7}$. Intuitively, this property implies the best trade-off between the worst-case competitive ratios corresponding to different ρ^* values. That is, any changes to the pricing function in (2) that may decrease the competitive ratio for some ρ^*, will increase the competitive ratio for some other ρ^*, and thus can only lead to a worse competitive ratio over all possible values of ρ^*.

3.2.2 Pricing Function for Small Total Demand. In the case that $\beta \in (-1, 0]$, the total resource demand is no larger than the total resource supply. The optimal strategy is simply serving all user demands by setting a unit resource price below the smallest per-unit-resource valuation of cloud users.

Theorem 3.9. *For $\beta \in (-1, 0]$, pricing function*

$$P_4(\rho) = \underline{p} \tag{8}$$

is optimal according to Definition 3.2, and the corresponding worst-case competitive ratio achieved by Algorithm 1 is 1.

The proof is straightforward and hence omitted.

3.2.3 Pricing Function for Total Demand Up to Twice of Supply. In the case that $\beta \in (0, 1)$, we first derive pricing functions that have Property 1, and then prove the optimality of the functions. In the following derivation, we assume that all pricing functions are continuous and non-decreasing, for the solution existence of our differential equations. However, the assumptions are not required by the proof of optimality. The following claim will be useful in the derivation.

Claim 3.2. *For any $\beta > -1$, if a pricing function $P(\rho)$ leads to a finite worst-case competitive ratio of α, then $P(\rho) = \underline{p}, \forall \rho \in [0, 1/\alpha]$.*

Proof. For $\beta > 0$, the proof is similar to that of Claim 3.1 and is omitted. For $\beta \in (-1, 0]$, the claim follows immediately from Theorem 3.9. □

Our derivation of the pricing function is further divided into two cases.

Case 1: $\beta \in (\beta_0, 1)$ where $\beta_0 \in (0, 1)$, such that $\beta > 1/\alpha_2$ and α_2 is the worst-case competitive ratio achieved using the optimal pricing function for $\beta \in (\beta_0, 1)$. According to Claim 3.2, the pricing function $P_2(\rho) = \underline{p}, \forall \rho \in [0, 1/\alpha_2]$. When $\rho^* \in (1/\alpha_2, 1)$, as discussed for Eq. (3), the minimum total value of an online solution is

$$V_{ol}(\rho^*) = \int_0^{\rho^*} P_2(\rho)\, d\rho, \tag{9}$$

and hence

$$\frac{dV_{ol}(\rho^*)}{d\rho^*} = \frac{d\left(\int_0^{\rho^*} P_2(\rho)\, d\rho\right)}{d\rho^*} = P_2(\rho^*), \tag{10}$$

which is illustrated by the light blue area in Fig. 2a. Since $P_2(\rho)$ is non-decreasing, when $\rho^* \in (1/\alpha_2, \beta]$, we still have $V_{opt}(\rho^*) = P_2(\rho^*) - \epsilon$ as discussed for Eq. (4), where ϵ is an arbitrarily small positive value. Thus

$$\frac{d \sup_{\epsilon>0} V_{opt}(\rho^*)}{d\rho^*} = \frac{dP_2(\rho^*)}{d\rho^*}. \tag{11}$$

It follows from Eq. (7), (10) and (11) that

$$\frac{dP_2(\rho)}{d\rho} - \alpha_2 P_2(\rho) = 0. \tag{12}$$

Solving the differential equation above gives $P_2(\rho) = Ce^{\alpha_2 \rho}$, where C is a constant to be determined. Since we assumed the continuity of $P_2(\rho)$, we let $\lim_{\rho \to 1/\alpha_2 +} P_2(\rho) = P_2(1/\alpha_2) = \underline{p}$, and then we obtain $C = \underline{p}/e$, and $P_2(\rho) = \underline{p}e^{\alpha_2 \rho - 1}, \forall \rho \in (1/\alpha_2, \beta]$.

When $\rho^* \in (\beta, 1)$, having a set of users with a unit value of $P_2(\rho^*) - \epsilon$ to consume all resource is no longer possible in the worst case. Instead, there can be a set of unsatisfied users with a total demand of $1 + \beta - \rho^*$, and with a unit value of $P_2(\rho^*) - \epsilon$, such that the optimal offline solution yields the maximum optimal offline total value given Eq. (9):

$$V_{opt}(\rho^*) = (1 + \beta - \rho^*)(P_2(\rho^*) - \epsilon) + \int_{\beta}^{\rho^*} P_2(\rho)\, d\rho, \tag{13}$$

as shown by the red and yellow AUCs (partially covered by the blue one) in Fig. 2a. We have

$$\frac{d \sup_{\epsilon > 0} V_{opt}(\rho^*)}{d\rho^*} = (1 + \beta - \rho^*) \frac{dP_2(\rho^*)}{d\rho^*}, \tag{14}$$

which is illustrated by the light red areas in Fig. 2a. Note that, there can be a case which leads to a larger optimal offline total value, by increasing the value corresponding to $\rho \in [\beta, \rho^*]$ (*i.e.*, the yellow AUC in Fig. 2a). Suppose the increased optimal offline total value is $V_{opt}(\rho^*) + \Delta$ ($\Delta > 0$), the online total value will also be increased to $V_{ol}(\rho^*) + \Delta$. However, since the competitive ratio now changes to $\sup_{\epsilon > 0} \frac{V_{opt}(\rho^*) + \Delta}{V_{ol}(\rho^*) + \Delta} < \sup_{\epsilon > 0} \frac{V_{opt}(\rho^*)}{V_{ol}(\rho^*)}$, it cannot be the worst case.

It follows from Eq. (7), (10) and (14) that

$$(1 + \beta - \rho) \frac{dP_2(\rho)}{d\rho} - \alpha_2 P_2(\rho) = 0. \tag{15}$$

Solving the differential equation above gives $P_2(\rho) = C(1 + \beta - \rho)^{-\alpha_2}$, where C is a constant to be determined. Again, due to the continuity of $P_2(\rho)$, we let $\lim_{\rho \to \beta +} P_2(\rho) = P_2(\beta) = \underline{p}e^{\alpha_2 \beta - 1}$. Then we obtain $C = \underline{p}e^{\alpha_2 \beta - 1}$, and $P_2(\rho) = \underline{p}e^{\alpha_2 \beta - 1}(1 + \beta - \rho)^{-\alpha_2}, \forall \rho \in (\beta, 1]$. To have a constant competitive ratio at $\rho^* = 1-$ and $\rho^* = 1$, as suggested by Property 1, we let $P_2(1) = \underline{p}e^{\alpha_2 \beta - 1}\beta^{-\alpha_2} = \bar{p} = \gamma \underline{p}$, which leads to

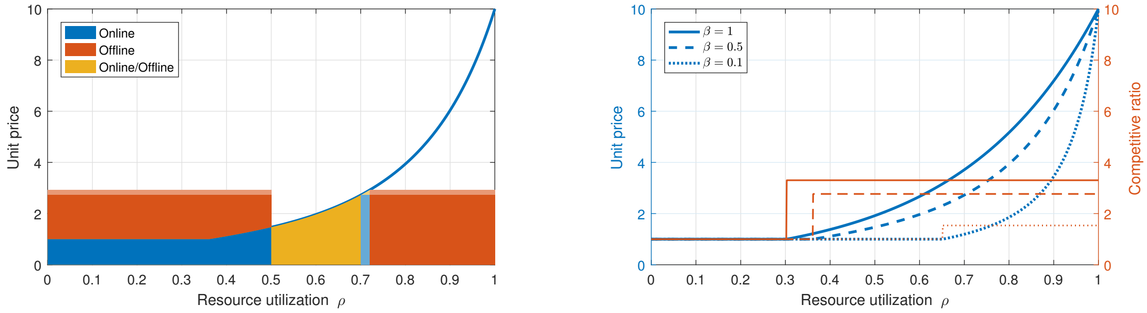
$$\alpha_2 = \frac{\log \gamma + 1}{\beta - \log \beta}. \tag{16}$$

To obtain the value of β_0, let $\beta = \beta_0 = 1/\alpha_2$. By Eq. (16), we obtain

$$\beta_0 = \frac{W(\log \gamma)}{\log \gamma}. \tag{17}$$

Here, $W(\cdot)$ is the Lamber W-function (*a.k.a.* the omega function or the product logarithm), which is the inverse function of $f(W) = We^W$. Therefore, for $\beta \in (\beta_0, 1)$, the pricing function is

$$P_2(\rho) = \begin{cases} \underline{p}, & \rho \in [0, 1/\alpha_2] \\ \underline{p}e^{\alpha_2 \rho - 1}, & \rho \in (1/\alpha_2, \beta] \\ \underline{p}e^{\alpha_2 \beta - 1}(1 + \beta - \rho)^{-\alpha_2}, & \rho \in (\beta, 1) \\ +\infty, & \rho = 1 \end{cases}. \tag{18}$$



(a) The worst-case $V_{ol}(\rho^*)$ and $V_{opt}(\rho^*)$ for $\beta = 0.5$, $\rho^* = 0.7$ visualized by AUCs.

(b) Pricing functions and competitive ratios for different values of β.

Fig. 2. Pricing functions and competitive ratios for $\beta \in (0, 1)$, $\underline{p} = 1, \overline{p} = 10$.

An example of $P_2(\rho)$ is shown in Fig. 2b by the dashed line corresponding to $\beta = 0.5$, where $\beta_0 = 0.399$. In practice, β can be estimated or optimized against competitive ratios.

Theorem 3.10. *For $\beta \in (\beta_0, 1)$, the pricing function in* (18) *is optimal according to Definition 3.2, and the corresponding worst-case competitive ratio is α_2.*

Proof. The proof of the worst-case competitive ratio α_2 follows that of Theorem 3.6, and is omitted.

Suppose there exists a pricing function, $P'_2(\rho)$, which achieves a worst-case competitive ratio $\alpha'_2 < \alpha_2$. According to Claim 3.2 and the proof of Theorem 3.7, we have

$$\int_0^\beta P'_2(\rho)\,d\rho < \int_0^\beta P_2(\rho)\,d\rho,$$

where $P_2(\rho)$ is the pricing function in (18).

If there exists some $\rho \in (\beta, 1)$ such that $P'_2(\rho) \geq P_2(\rho)$, we identify the smallest one and denote it by ρ_1. Then there can be a case where $\rho^* = \rho_1$, such that the online total value

$$V'_{ol}(\rho^*) = \int_0^{\rho_1} P'_2(\rho)\,d\rho < \int_0^{\rho_1} P_2(\rho)\,d\rho - \Delta = V_{ol}(\rho^*) - \Delta,$$

where $\Delta = \int_\beta^{\rho_1} \left[P_2(\rho) - P'_2(\rho)\right] d\rho$; while the optimal offline total value $V'_{opt}(\rho^*) \geq V_{opt}(\rho^*) - \Delta$ according to Eq. (13). Thus the worst-case competitive ratio $\alpha'_2 \geq \sup_{\epsilon>0} \frac{V'_{opt}(\rho^*)}{V'_{ol}(\rho^*)} > \sup_{\epsilon>0} \frac{V_{opt}(\rho^*)-\Delta}{V_{ol}(\rho^*)-\Delta} > \alpha_2$, contradicting the assumption $\alpha'_2 < \alpha_2$. Therefore, $P'_2(\rho) < P_2(\rho), \forall \rho \in (\beta, 1)$.

For $\rho^* = 1$, since $P'_2(1) \leq \overline{p}$ (a unit price higher than $\overline{p}$ will reject all potential users) is finite, we now have

$$V'_{ol}(\rho^*) = \int_0^1 P'_2(\rho)\,d\rho < \int_0^1 P_2(\rho)\,d\rho - \Delta = V_{ol}(\rho^*) - \Delta,$$

where $\Delta = \int_\beta^1 \left[P_2(\rho) - P'_2(\rho)\right] d\rho$. However, as the resource is exhausted, subsequent users will not be satisfied regardless of their valuations. There can be a case where the optimal offline total value $V'_{opt}(\rho^*) = V_{opt}(\rho^*) - \Delta$

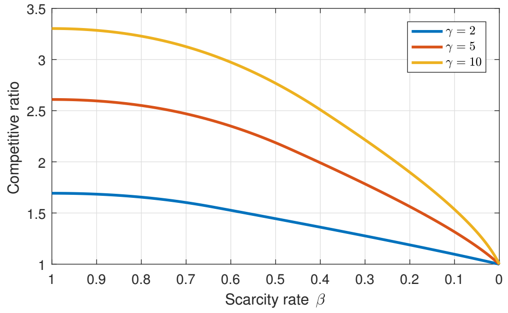


Fig. 3. Competitive ratios for different values of β and γ.

according to Eq. (13). Thus the worst-case competitive ratio $\alpha_2' \geq \frac{V_{opt}'(\rho^*)}{V_{ol}'(\rho^*)} > \frac{V_{opt}(\rho^*)-\Delta}{V_{ol}(\rho^*)-\Delta} > \alpha_2$, contradicting the assumption $\alpha_2' < \alpha_2$. □

Case 2: $\beta \in (0, \beta_0]$. From the definition of β_0, we have $\beta \leq 1/\alpha_3$, where α_3 is the worst-case competitive ratio of the optimal pricing function in this case. According to Claim 3.2, the pricing function $P_3(\rho) = \underline{p}, \forall \rho \in [0, 1/\alpha_3]$. When $\rho^* \in (1/\alpha_3, 1)$, $V_{ol}(\rho^*)$ follows Eq. (10) with $P_2(\rho^*)$ replaced by $P_3(\rho^*)$; $V_{opt}(\rho^*)$ follows Eq. (13), (14) with $P_2(\rho^*)$ replaced by $P_3(\rho^*)$. Then, following Eq. (15), we have $P_3(\rho) = C(1+\beta-\rho)^{-\alpha_3}$. As discussed for Eq. (16), we let $\lim_{\rho \to 1/\alpha_3+} P_3(\rho) = P_3(1/\alpha_3) = \underline{p}$, $P_3(1) = \overline{p} = \gamma \underline{p}$. Solving the resulting equations:

$$\begin{aligned} C\left(1+\beta-\frac{1}{\alpha_3}\right)^{-\alpha_3} &= \underline{p}, \\ C\beta^{-\alpha_3} &= \gamma \underline{p}, \end{aligned}$$

we get

$$\alpha_3 = \frac{\log \gamma}{(1+\beta)\log\gamma - W\left(\beta\gamma^{1+\beta}\log\gamma\right)}, \tag{19}$$

and the pricing function for $\beta \in (0, \beta_0]$ is:

$$P_3(\rho) = \begin{cases} \underline{p}, & \rho \in [0, 1/\alpha_3] \\ \underline{p}\gamma\beta^{\alpha_3}(1+\beta-\rho)^{-\alpha_3}, & \rho \in (1/\alpha_3, 1) \\ +\infty, & \rho = 1 \end{cases}. \tag{20}$$

An example of $P_3(\rho)$ is shown in Fig. 2b by the dashed line corresponding to $\beta = 0.2$.

Theorem 3.11. *For $\beta \in (0, \beta_0]$, the pricing function in* (20) *is optimal according to Definition 3.2, and the corresponding worst-case competitive ratio is α_3.*

Proof. The proof is similar to that of Theorem 3.10 and is omitted. □

For better illustrating how $\beta \in (0,1)$ affects the competitive ratio as dictated by Theorems 3.8, 3.10 and 3.11, we plot the competitive ratio as a function of β in Fig. 3. As shown in the figure, for a certain value of γ, the competitive ratio decreases with the decrease of β, and reaches the minimum value 1 when β drops to 0.

Putting Eq. (2), (18) and (20) together, we have obtained a 2-dimensional piecewise pricing function, $P(\rho; \beta)$. An illustration of the pricing function is given in Fig. 4.

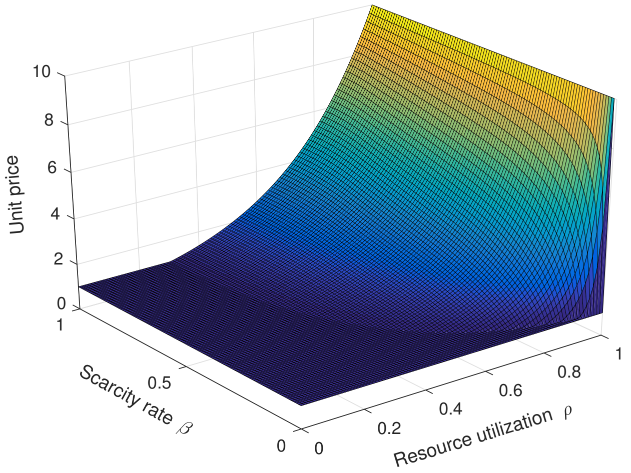


Fig. 4. 2-D pricing function $P(\rho;\beta)$ for $\rho \in [0,1]$, $\beta \in [0,1]$.

3.3 Linear Operational Cost

Resource provisioning in real-world cloud computing systems often incurs an operational cost. If such cost is proportional to the amount of resources provisioned, then we have a linear operational cost [29]. We can extend the proposed pricing strategy to accommodate such linear operational cost by making two modifications. First, we replace Assumption 1 by:

ASSUMPTION 3. *The variability of users' valuations is constrained,* i.e., $\underline{p} + c \le v_i/d_i \le \overline{p} + c$.

Here, $c \ge 0$ is the operational cost of using a unit of resource. Second, we replace the pricing functions (2), (18) and (20), by $P_1'(\rho) = P_1(\rho) + c$, $P_2'(\rho) = P_2(\rho) + c$ and $P_3'(\rho) = P_3(\rho) + c$. Then all discussions about the proposed pricing strategy remain valid, including the proof of optimality. In the rest of this paper, we ignore operational cost for simplicity.

4 PRICING MULTIPLE RESOURCE TYPES WITH RESOURCE RECYCLING

In this section, we extend our resource allocation problem in (1) to one with multiple types of resources (Sec. 4.1), and then further investigate the practical case where resource occupation spans multiple time slots (Sec. 4.2). We show that, by carefully designing the pricing and scheduling strategy, the worst-case competitive ratio in social welfare will not be influenced by the number of resource types, or by the number of requested time slots.

4.1 Pricing Function for Multiple Types of Resources

Now we consider a cloud system that provisions multiple types of resources in a set $\mathcal{R}$, as exemplified by CPU, GPU, RAM, and disk storage. Let $d_{i,r}$ be user i's demand for resource r, $\forall r \in \mathcal{R}$. Again, we assume the total amount of each type of resource is 1, so that $d_{i,r}$ is the proportion of the overall supply of resource r demanded by i.

The offline social welfare maximization problem is:

$$\text{maximize} \quad \sum_{i \in \mathcal{U}} v_i x_i \tag{21}$$

s.t.:

$$\sum_{i \in \mathcal{U}} d_{i,r} x_i \leq 1, \forall r \in \mathcal{R} \tag{21a}$$

$$x_i \in \{0, 1\}, \forall i \in \mathcal{U} \tag{21b}$$

The online resource allocation algorithm we apply to determine x_i immediately after user i arrives at the cloud, is the same as Alg. 1, except that d_i and the pricing function will be redefined.

Given the optimal pricing functions (2) (for $\beta \geq 1$), (18) (for $\beta \in (\beta_0, 1)$), (20) (for $\beta \in (0, \beta_0]$) and (8) (for $\beta \in (-1, 0]$) in case of a single resource type, we can simply price each type of resource independently as p_r, using these pricing functions, and sum them up by $\sum_{r \in \mathcal{R}} d_{i,r} p_r$ to form a total price; a user i accepts the prices and rents resources at quantities $d_{i,r}$'s, if and only if v_i is no smaller than the total price. Before doing so, we need to redefine $\underline{p}$ and $\overline{p}$. One way is to define them for each type of resource independently, as $\underline{p}_r = \inf_i \frac{v_i}{d_{i,r}}$, and $\overline{p}_r = \sup_i \frac{v_i}{d_{i,r}}$, as done by Zhang et al. [29]. However, a drawback of this definition is that $\overline{p}_r$ can be infinite, as we do not assume that every user demands all types of resources. A remedy is to define the same $\underline{p}$ and $\overline{p}$ for all types of resources, as $\underline{p} = \inf_i \frac{v_i}{d_i}$, and $\overline{p} = \sup_i \frac{v_i}{d_i}$, where $d_i = \sum_{r \in \mathcal{R}} d_{i,r}$. In this way, Assumption 1 or 3 remains intact. The definitions of $\underline{p}$ and $\overline{p}$ are a simple extension of Assumption 1 to the multi-resource case. Compared to the former definition, they do not make any (implicit) assumptions on the ratio of different resources each user demands, and thus are more practical. Moreover, summing up the demand for different types of resources is reasonable when each $d_{i,r}$ is normalized by the total supply of the corresponding resource, such that their values are all in the range of $[0, 1]$. Then given the resource utilization ρ_r and scarcity level β_r of each type of resource $r \in \mathcal{R}$, we define an average unit price for any resource for user i as

$$\mathscr{P}_i\left(\boldsymbol{\rho}\right) = \frac{1}{d_i} \sum_{r \in \mathcal{R}} d_{i,r} P\left(\rho_r; \beta_r\right) \tag{22}$$

where $\boldsymbol{\rho}$ denotes the vector of $\rho_r, \forall r \in \mathcal{R}$, and $P\left(\rho_r; \beta_r\right)$ is defined by Eq. (2), (18), (20) and (8). Therefore, $d_i \mathscr{P}_i\left(\boldsymbol{\rho}\right)$ is the total price for user i. Note that we omit β_r in $\mathscr{P}_i\left(\boldsymbol{\rho}\right)$ for notation simplicity, but different β_r will lead to different $\mathscr{P}_i\left(\boldsymbol{\rho}\right)$.

While it is traightforward to adapt the pricing strategy for a single resource type to multiple resource types, the resulting worst-case competitive ratio will be different. Specifically, we denote the final resource utilization level of resource r by ${\rho_r}^*, \forall r \in \mathcal{R}$, according to Definition 3.3, and analyze competitive ratios in three cases: (i) ${\rho_r}^* \in [0, 1/\alpha_r], \forall r \in \mathcal{R}$; (ii) there exists an $r \in \mathcal{R}$ such that ${\rho_r}^* \in (1/\alpha_r, 1)$, but no $r \in \mathcal{R}$ such that ${\rho_r}^* = 1$; (iii) there exists an $r \in \mathcal{R}$ such that ${\rho_r}^* = 1$. Here, α_r is defined by Eq. (5), (16) or (19) for $\beta = \beta_r$. We denote the three cases by $\boldsymbol{\rho}^* \in \Omega_1$, $\boldsymbol{\rho}^* \in \Omega_2$ and $\boldsymbol{\rho}^* \in \Omega_3$, respectively, and observe that $\Omega_1 \cup \Omega_2 \cup \Omega_3$ covers all possible values of $\boldsymbol{\rho}^*$. Without loss of generality, here we assume not all $\beta_r \leq 0$; otherwise the worst-case competitive ratio would be 1.

Lemma 4.1. *For $\boldsymbol{\rho}^* \in \Omega_1$, the worst-case competitive ratio achieved by Alg. 1 using pricing function (22) for multiple types of resources is $\alpha^1 = 1$.*

Proof. For $\boldsymbol{\rho}^* \in \Omega_1$, according to the pricing function in (22), $\mathscr{P}_i\left(\boldsymbol{\rho}^*\right) = \underline{p}$, which by Assumption 1 is acceptable to any potential users, thus the total demand of all users for resource r is exactly ${\rho_r}^*$. The social welfare achieved by the pricing function in (22) is the total value of all users, which is also the maximum possible social welfare achieved by solving the offline problem (21). Therefore, the worst-case competitive ratio $\alpha^1 = 1$. □

For $\boldsymbol{\rho}^* \in \Omega_2$, we first present the following claim, which states that worst cases happen when all users demand only one specific type of resource, driving the average unit price to slightly beyond $\underline{p}$.

Claim 4.1. *Let $\rho_r{}^* \in [0, 1/\alpha_r]$ for $r \in \mathcal{R}_1$, $\rho_r{}^* \in (1/\alpha_r, 1)$ for $r \in \mathcal{R}_2$, where $\mathcal{R}_1 \cup \mathcal{R}_2 = \mathcal{R}$. For $\boldsymbol{\rho}^* \in \Omega_2$, there exists a worst case that happens when $\rho_r{}^* = 0$ for $r \in \mathcal{R}_1$, and $\rho_r{}^* = 1/\alpha_r + \epsilon$ for $r \in \mathcal{R}_2$, where $|\mathcal{R}_2| = 1$. Here, ϵ is an arbitrarily small number.*

The proof can be found in the appendix.

Lemma 4.2. *For $\boldsymbol{\rho}^* \in \Omega_2$, the corresponding worst-case competitive ratio $\alpha^2 = \alpha_{\bar{r}} \sum_{r \in \mathcal{R}} \min\{1, 1 + \beta_r\}$, where $\alpha_{\bar{r}}$ is defined by Eq.* (5), (16) *or* (19) *for $\beta = \beta_{\bar{r}}$, and $\bar{r} = \arg\max_{r \in \mathcal{R}} \alpha_r$.*

Proof. According to Claim 4.1, we let $\rho_r{}^* = 0$ for $r \in \mathcal{R}_1$, and $\rho_r{}^* = 1/\alpha_r + \epsilon$ for $r \in \mathcal{R}_2$, and let $|\mathcal{R}_2| = 1$. Then from Eq. (31) and (32), $\mathcal{R}_2 = \{\bar{r}\}$ maximizes $\alpha\left(\boldsymbol{\rho}^*\right)$, and thus is a worst case for $\boldsymbol{\rho}^* \in \Omega_2$. The corresponding competitive ratio

$$\alpha^2 = \sup_{\epsilon > 0} \frac{\sum_{r \in \mathcal{R}} \underline{p} \min\{1, 1 + \beta_r\}}{\underline{p} \rho_{\bar{r}}{}^*} = \alpha_{\bar{r}} \sum_{r \in \mathcal{R}} \min\{1, 1 + \beta_r\}. \tag{23}$$

□

For $\boldsymbol{\rho}^* \in \Omega_3$, the following claim states that worst cases happen when all users satisfied by an online solution demand only one specific type of resource until it is exhausted.

Claim 4.2. *Let $\rho_r{}^* \in [0, 1)$ for $r \in \mathcal{R}_3$, $\rho_r{}^* = 1$ for $r \in \mathcal{R}_4$, where $\mathcal{R}_3 \cup \mathcal{R}_4 = \mathcal{R}$. For $\boldsymbol{\rho}^* \in \Omega_3$, there exists a worst case that happens when $\rho_r{}^* = 0$ for $r \in \mathcal{R}_3$, and $\rho_r{}^* = 1$ for $r \in \mathcal{R}_4$, where $|\mathcal{R}_4| = 1$.*

The proof can be found in the appendix.

Lemma 4.3. *For $\boldsymbol{\rho}^* \in \Omega_3$, the corresponding worst-case competitive ratio $\alpha^3 \geq \alpha_{\bar{r}} \sum_{r \in \mathcal{R}} \min\{1, 1 + \beta_r\}$.*

Proof. According to Claim 4.2, we let $\rho_r{}^* = 0$ for $r \in \mathcal{R}_3$, and $\rho_r{}^* = 1$ for $r \in \mathcal{R}_4$, and let $|\mathcal{R}_4| = 1$. Then from Eq. (33) and (34), we have the worst-cast competitive ratio for $\boldsymbol{\rho}^* \in \Omega_3$

$$\alpha^3 = \max_{r' \in \mathcal{R}'} \frac{\sum_{r \in \mathcal{R} \setminus \{r'\}} \bar{p} \min\{1, 1 + \beta_r\} + \alpha_{r'} \int_0^1 P(\rho; \beta_{r'})\, d\rho}{\int_0^1 P(\rho; \beta_{r'})\, d\rho}, \tag{24}$$

where $R' = \{r | \beta_r > 0\}$. Since it is assumed that $\beta_{\bar{r}} > 0$, we have $\min\{1, 1 + \beta_{\bar{r}}\} = 1$, and hence

$$\frac{\alpha_{\bar{r}} \int_0^1 P(\rho; \beta_{\bar{r}})\, d\rho}{\int_0^1 P(\rho; \beta_{\bar{r}})\, d\rho} = \frac{\bar{p} \min\{1, 1 + \beta_{\bar{r}}\}}{\bar{p}/\alpha_{\bar{r}}}.$$

And since $\alpha_{\bar{r}} \int_0^1 P(\rho; \beta_{\bar{r}})\, d\rho \leq \bar{p} \min\{1, 1 + \beta_{\bar{r}}\}$, we have

$$\begin{aligned} \alpha^3 &\geq \frac{\sum_{r \in \mathcal{R} \setminus \{\bar{r}\}} \bar{p} \min\{1, 1 + \beta_r\} + \alpha_{\bar{r}} \int_0^1 P(\rho; \beta_{\bar{r}})\, d\rho}{\int_0^1 P(\rho; \beta_{\bar{r}})\, d\rho} \\ &\geq \frac{\sum_{r \in \mathcal{R}} \bar{p} \min\{1, 1 + \beta_r\}}{\bar{p}/\alpha_{\bar{r}}} = \alpha_{\bar{r}} \sum_{r \in \mathcal{R}} \min\{1, 1 + \beta_r\}. \end{aligned}$$

□

By Lemma 4.1, 4.2 and 4.3, we have the following theorem:

Theorem 4.4. *The worst-case competitive ratio achieved by Alg. 1 using the pricing function in* (15) *for multiple types of resources is given by Eq.* (24).

As shown by Lemma 4.3, the worst-case competitive ratio for multiple resource types increases roughly linearly with the number of resource types. However, from Claim 4.1, 4.2, and the analysis above, it is clear that the worst cases happen in rather extreme scenarios, where all satisfied users demand only one type of resource, unrealistic in practical cloud computing systems. In fact, the supply of and the demand for resources in a cloud computing system are often balanced to some extent, since otherwise the supply would be adjusted to better meet the demand of users and to improve the system efficiency. Hence, we make the following realistic assumption:

assumption 4. *All types of resources share a common scarcity level, $\beta_{\mathcal{R}} > 0$, and hence a common $\alpha_{\mathcal{R}}$ as defined by Eq.* (5), (16) *or* (19) *for $\beta = \beta_{\mathcal{R}}$; and the final utilization vector, $\boldsymbol{\rho}^*$, follows*

$$\frac{\min_{r\in\mathcal{R}} {\rho_r}^*}{\max_{r\in\mathcal{R}} {\rho_r}^*} \geq \eta. \tag{25}$$

Assumption 4 leads to an improved competitive ratio.

Theorem 4.5. *Under Assumption 4, the worst-case competitive ratio with the pricing function in* (15) *is upper bounded by a constant with respect to $|\mathcal{R}|$.*

Proof. It can be shown that Claim 4.1 and 4.2 are still valid under Assumption 4. For $\boldsymbol{\rho}^* \in \Omega_2$, any worst case gives $V_{ol}\left(\boldsymbol{\rho}^*\right) = [1 + (|\mathcal{R}| - 1)\,\eta]\,\underline{p}/\alpha_{\mathcal{R}}$ and $V_{opt}\left(\boldsymbol{\rho}^*\right) = |\mathcal{R}|\,\underline{p}$, and hence the corresponding competitive ratio $\alpha^2 = \frac{|\mathcal{R}|}{1+(|\mathcal{R}|-1)\eta}\alpha_{\mathcal{R}}$. Since $\eta \leq 1$, we have $\alpha^2 \leq \alpha_{\mathcal{R}}/\eta$. For $\boldsymbol{\rho}^* \in \Omega_3$, as $\epsilon \longrightarrow 0$, any worst case gives

$$V_{ol}\left(\boldsymbol{\rho}^*\right) = \int_0^1 P\left(\rho;\beta_{\mathcal{R}}\right) d\rho + \left(|\mathcal{R}| - 1\right) \int_0^\eta P\left(\rho;\beta_{\mathcal{R}}\right) d\rho,$$

and

$$V_{opt}\left(\boldsymbol{\rho}^*\right) = \alpha_{\mathcal{R}} V_{ol}\left(\boldsymbol{\rho}^*\right) + \left(|\mathcal{R}| - 1\right)\left(1 + \beta_{\mathcal{R}} - \eta\right)\left(\overline{p} - P\left(\eta;\beta_{\mathcal{R}}\right)\right),$$

and hence the corresponding competitive ratio

$$\alpha^3 = \alpha_{\mathcal{R}} + \frac{\left(1 + \beta_{\mathcal{R}} - \eta\right)\left(\overline{p} - P\left(\eta;\beta_{\mathcal{R}}\right)\right)}{\int_0^1 P\left(\rho;\beta_{\mathcal{R}}\right) d\rho/\left(|\mathcal{R}| - 1\right) + \int_0^\eta P\left(\rho;\beta_{\mathcal{R}}\right) d\rho}.$$

Let

$$\theta = \frac{\left(1 + \beta_{\mathcal{R}} - \eta\right)\left(\overline{p} - P\left(\eta;\beta_{\mathcal{R}}\right)\right)}{\int_0^\eta P\left(\rho;\beta_{\mathcal{R}}\right) d\rho},$$

we have $\alpha^3 < \alpha_{\mathcal{R}} + \theta$. Therefore, the worst-cast competitive ratio under Assumption 4 is upper bounded by $\max\{\alpha_{\mathcal{R}}/\eta, \alpha_{\mathcal{R}} + \theta\}$. □

Theorem 4.5 justifies the pricing function in (22) by showing that, a direct extension of the optimal pricing functions for the single resource type case can achieve a reasonably good (degraded by a constant factor w.r.t. $|\mathcal{R}|$) competitive ratio in scenarios with multiple resource types.

4.2 Pricing Function for Multiple Time Slots

In real-world cloud systems, a user job runs over its specified resource bundle in the cloud, across one or more time slots. Once the job is completed, the resources that it occupies are then released back to the cloud pool.

Therefore, cloud resources can be reused over time. Let $\mathcal{T}$ denote the set of all time slots that the system spans, and $\mathcal{T}_i$ be the set of time slots when user i requires to use resources. $y_i(t)$ is an indication function as follows:

$$y_i(t) = \begin{cases} 1, & t \in \mathcal{T}_i \\ 0, & \text{otherwise} \end{cases}. \tag{26}$$

The offline social welfare maximization problem becomes:

$$\text{maximize} \quad \sum_{i \in \mathcal{U}} v_i x_i \tag{27}$$

s.t.:

$$\sum_{i \in \mathcal{U}} d_{i,r} x_i y_i(t) \le 1, \forall r \in \mathcal{R}, t \in \mathcal{T} \qquad (27a)$$

$$x_i \in \{0,1\}, \forall i \in \mathcal{U} \qquad (27b)$$

Since $y_i(t)$ is input (not a variable) in this optimization problem, problem (27) is still an ILP. The online resource allocation algorithm we apply to determine x_i upon the arrival of user i is still the same as Alg. 1, except that d_i and the pricing function will be redefined, and $y_i(t)$ needs to be further determined.

In fact, problem (21) and problem (27) are equivalent if we consider resource r in different time slots to be of different resource types. More specifically, let $d_{i,r(t)} = d_{i,r} y_i(t)$, where $r(t) \in \mathcal{R}(t)$, and $t \in \mathcal{T}$. Then problem (27) will have exactly the same form as problem (27). Therefore, according to Lemma 4.3 and Theorem 4.4, the worst-case competitive ratio will increase roughly linearly with the number of time slots, $|\mathcal{T}|$, if no other assumptions are made. If the number of slots required by each user is upper bounded, then the worst-case competitive ratio will increase roughly linearly with the maximum number of slots required by each user, which is also undesirable. Intuitively, this issue is caused by the fact that, if one of the time slots required by a user is unavailable (e.g., no available resources), then the demand of the user cannot be satisfied as a whole, even if other required slots are all available.

To address the aforementioned problem, we propose a strategy that satisfies users' demand in an elastic manner. Specifically, assuming we are allowed to satisfy user i with any $|\mathcal{T}_i|$ slots in a larger set of time slots, $\mathcal{T}_i' \supseteq \mathcal{T}_i$, we can significantly improve the competitive ratio by choosing $|\mathcal{T}_i|$ slots from $\mathcal{T}_i'$ that yield the lowest total price. Concretely, the corresponding online resource scheduling strategy is that, we try to satisfy each user i with $|\mathcal{T}_i|$ time slots chosen from $\mathcal{T}_i'$, and $\left|\mathcal{T}_i'\right| = \lceil \lambda |\mathcal{T}_i| \rceil$, where λ is a constant factor. Here $\mathcal{T}_i'$ can be interpreted as the allowed (loosened) time interval for completing the user's job. The overall price to user i is computed as the minimum possible total price of $|\mathcal{T}_i|$ time slots selected from $\mathcal{T}_i'$.

From the user perspective, the price each user receives is determined upon its arrival in the system, and does not change afterwards. A user i accepts the price and leases resource at quantities $d_{i,r}$'s in the chosen $|\mathcal{T}_i|$ time slots, if and only if v_i is no smaller than the overall price. Once a user accepts the price, its job is guaranteed to be completed within $\lambda |\mathcal{T}_i|$. If the provider tells that a job cannot be completed within $\lambda |\mathcal{T}_i|$, the job will receive an infinitely high price according to the pricing function upon arrival (*i.e.*, the user will reject the price and the job will not be executed).

In fact, similar non-consecutive execution schemes have been implemented on Amazon EC2 Spot Instance [2], and have been discussed in the literature [30]. Here, we further justify the use of non-consecutive execution schemes from a theoretical point of view.

Without loss of generality, we assume both $\mathcal{T}_i$ and $\mathcal{T}_i'$ are consecutive time slots; and if $\mathcal{T}_i = [\tau_i, \tau_i + |\mathcal{T}_i| - 1]$, we let $\mathcal{T}_i' = \left[\tau_i, \tau_i + \left|\mathcal{T}_i'\right| - 1\right]$. To formulate the offline version of the modified social welfare maximization problem, we can add the following constraints to problem (27):

$$\sum_{t\in\mathcal{T}_i'} y_i(t) = |\mathcal{T}_i|, \forall i \in \mathcal{U} \qquad (27c)$$

$$y_i(t) \in \{0,1\}, \forall i \in \mathcal{U}, t \in \mathcal{T} \qquad (27d)$$

Note that $y_i(t)$ now follows Eq. (27c) and (27d), instead of Eq. (26), and $y_i(t)$ becomes a variable. Therefore, the new problem is no longer an ILP.

We reuse the notation $\mathscr{P}_i(\cdot)$ to denote the pricing function for user i; and we reuse the symbols, d_i and $\boldsymbol{\rho}$, to taken into account different time slots, *i.e.*, $d_i = |\mathcal{T}_i| \sum_{r\in\mathcal{R}} d_{i,r}$, and $\boldsymbol{\rho}$ denotes the vector of $\rho_r(t), \forall r \in \mathcal{R}, t \in \mathcal{T}$. The definitions of $\underline{p}$ and $\overline{p}$ remain the same, *i.e.*, $\underline{p} = \inf_i \frac{v_i}{d_i}$ and $\overline{p} = \sup_i \frac{v_i}{d_i}$. Then under Assumption 4, our pricing strategy for online resource allocation can be described by the following pricing function:

$$\mathscr{P}_i(\boldsymbol{\rho}) = \frac{1}{d_i} \min_{\boldsymbol{y}_i \in \mathcal{Y}_i} \left[\sum_{t\in\mathcal{T}_i'} \sum_{r\in\mathcal{R}} d_{i,r} y_i(t) P(\rho_r(t); \beta_{\mathcal{R}}) \right], \qquad (28)$$

where $\mathcal{Y}_i$ is defined by Eq. (27c) and (27d), and $P(\rho_r(t); \beta_{\mathcal{R}})$ is defined by Eq. (2), (18) and (20). Obviously, $d_i\mathscr{P}_i(\boldsymbol{\rho})$ is the total price for user i.

In general, $\mathscr{P}_i(\boldsymbol{\rho})$ sets different unit prices for different time slots, according to the scheduled resource utilization levels. Note that, the overall price that each user receives for its resource demand over the requested resource usage duration is determined when the user comes to the system and requests resources, and does not change over the course.

Given an arbitrary set of time slots $\mathcal{T}$, and the corresponding time horizon $|\mathcal{T}|$, any $\mathcal{T}_i \nsubseteq \mathcal{T}$ can be ignored since it cannot be satisfied anyway. Furthermore, we ignore the marginal effect of any $\mathcal{T}_i' \nsubseteq \mathcal{T}$, since $|\mathcal{T}|$ is usually significantly larger than $|\mathcal{T}_i'|$. Thus, we assume $\mathcal{T}_i, \mathcal{T}_i' \subseteq \mathcal{T}, \forall i \in \mathcal{U}$. As we did to analyze competitive ratios for multiple resource types, we divide possible values of final resource utilization levels into three cases: (i) $\rho_r{}^*(t) \in [0, 1/\alpha_{\mathcal{R}}], \forall r \in \mathcal{R}, t \in \mathcal{T}$; (ii) there exists an $r \in \mathcal{R}$ and a $t \in \mathcal{T}$ such that $\rho_r{}^*(t) \in (1/\alpha_R, 1)$, but no $r \in \mathcal{R}$ or $t \in \mathcal{T}$ such that $\rho_r{}^*(t) = 1$; (iii) there exists an $r \in \mathcal{R}$ and a $t \in \mathcal{T}$ such that $\rho_r{}^*(t) = 1$. Here, α_R is defined by Eq. (5), (16) or (19) for $\beta = \beta_R$. We denote the three cases by $\boldsymbol{\rho}^* \in \Pi_1$, $\boldsymbol{\rho}^* \in \Pi_2$ and $\boldsymbol{\rho}^* \in \Pi_3$, respectively.

Lemma 4.6. *For $\boldsymbol{\rho}^* \in \Pi_1$, the worst-case competitive ratio achieved by our online resource scheduling strategy using pricing function (28) is $\alpha^1 = 1$.*

Proof. The proof is similar to that of Lemma 4.1 and is omitted. □

Lemma 4.7. *For $\boldsymbol{\rho}^* \in \Pi_2$, the corresponding worst-case competitive ratio $\alpha^2 < \frac{\alpha_{\mathcal{R}}}{(\lambda-1)\eta} + 1$, where η is defined in Assumption 4.*

Proof. Let $\mathcal{T}_1 = \{t|\rho_r{}^*(t) \in [0, 1/\alpha_{\mathcal{R}}], \forall r \in \mathcal{R}\}$, and $\mathcal{T}_2 = \{t|\rho_r{}^*(t) \in (1/\alpha_{\mathcal{R}}, 1), \exists r \in \mathcal{R}\}$, and $\mathcal{T}_1 \cup \mathcal{T}_2 = \mathcal{T}$. For $\boldsymbol{\rho}^* \in \Pi_2$, following the proof of Lemma 4.1, there exists a worst case that happens when $\rho_r{}^*(t) = 0$ for all $r \in \mathcal{R}$ and $t \in \mathcal{T}_1$; while for $t \in \mathcal{T}_2$, $\rho_r{}^*(t) = 1/\alpha_r + \epsilon$ for some $r' \in \mathcal{R}$, and $\rho_r{}^*(t) = \eta(1/\alpha_r + \epsilon)$ for $r \in \mathcal{R}\backslash\{r'\}$. Here, ϵ is an arbitrarily small number. Following the proof of Theorem 4.5, as $\epsilon \longrightarrow 0$, we have

$$V_{ol}(\boldsymbol{\rho}^*) = |\mathcal{T}_2| [1 + (|\mathcal{R}| - 1)\eta] \underline{p}/\alpha_{\mathcal{R}},$$

as the minimum total value of the online solution. For any $\mathcal{T}_i$, since $|\mathcal{T}_i'| = \lceil \lambda |\mathcal{T}_i| \rceil$, the demand will be satisfied regardless of the user's valuation, unless $|\mathcal{T}_i' \cap \mathcal{T}_2| > \lceil \frac{\lambda-1}{\lambda} |\mathcal{T}_i'| \rceil$. In other words, if the demand of user i is not satisfied by the online solution, there must be at least $\lceil \frac{\lambda-1}{\lambda} |\mathcal{T}_i'| \rceil$ time slots in $\mathcal{T}_i'$ that also belong to $\mathcal{T}_2$; or equivalently, for any $\mathcal{S} \subseteq \mathcal{T}$, and any $\mathcal{T}_i' \subseteq \mathcal{S}$ that is not satisfied by the online solution, $|\mathcal{T}_i'| < \lfloor \frac{\lambda}{\lambda-1} |\mathcal{S} \cap \mathcal{T}_2| \rfloor$, and hence $|\mathcal{T}_i| < \lfloor \frac{1}{\lambda-1} |\mathcal{S} \cap \mathcal{T}_2| \rfloor$. Let $\mathcal{T}_2'$ be the union of all sets of consecutive time slots that contain $\mathcal{T}_2$, and have a cardinality of $\lfloor \frac{\lambda}{\lambda-1} |\mathcal{T}_2| \rfloor - 1$. When $|\mathcal{T}_i| < \lfloor \frac{1}{\lambda-1} |\mathcal{S} \cap \mathcal{T}_2| \rfloor$, since at least one type of resource in at least one required time slot has a unit price above $\underline{p}$, there can be a set of users in a worst case, demanding all resources in $\left|\mathcal{T}_2'\right|$ time slots, with $\mathscr{P}_i(\boldsymbol{\rho}) = \underline{p}$, where $\left|\mathcal{T}_2'\right| < 2\lfloor \frac{1}{\lambda-1} |\mathcal{T}_2| \rfloor + |\mathcal{T}_2|$. Thus we have the maximum optimal offline total value

$$V_{opt}(\boldsymbol{\rho}^*) < \left|\mathcal{T}_2'\right| |\mathcal{R}| \underline{p} < \left(2\lfloor \frac{1}{\lambda-1} |\mathcal{T}_2| \rfloor + |\mathcal{T}_2| \right) |\mathcal{R}| \underline{p}.$$

Therefore, for $\boldsymbol{\rho}^* \in \Pi_2$, the worst-cast competitive ratio

$$\alpha^2 < \frac{\frac{\lambda+1}{\lambda-1} |\mathcal{R}| |\mathcal{T}_2| \underline{p}}{|\mathcal{T}_2| \left[1 + (|\mathcal{R}| - 1)\eta\right] \underline{p}/\alpha_{\mathcal{R}}} = \frac{\lambda+1}{\lambda-1} \frac{|\mathcal{R}|}{1 + (|\mathcal{R}| - 1)\eta} \alpha_{\mathcal{R}} \le \frac{(\lambda+1)\alpha_{\mathcal{R}}}{(\lambda-1)\eta}. \tag{29}$$

□

Lemma 4.8. *For $\boldsymbol{\rho}^* \in \Pi_3$, the corresponding worst-case competitive ratio $\alpha^3 < \frac{\lambda+1}{(\lambda-1)\eta'}$, where $\eta' = \int_0^\eta P(\rho; \beta_{\mathcal{R}})\, d\rho / \overline{p}$.*

Proof. Let $\mathcal{T}_3 = \{t | \rho_r{}^*(t) \in [0, 1), \forall r \in \mathcal{R}\}$, and $\mathcal{T}_4 = \{t | \rho_r{}^*(t) = 1, \exists r \in \mathcal{R}\}$, and $\mathcal{T}_3 \cup \mathcal{T}_4 = \mathcal{T}$. For $\boldsymbol{\rho}^* \in \Pi_2$, following the proof of Lemma 4.2, there exists a worst case that happens when $\rho_r{}^*(t) = 0$ for all $r \in \mathcal{R}$ and $t \in \mathcal{T}_3$; while for $t \in \mathcal{T}_4$, $\rho_r{}^*(t) = 1$ for some $r' \in \mathcal{R}$, and $\rho_r{}^*(t) = \eta$ for $r \in \mathcal{R} \backslash \{r'\}$. Following the proof of Theorem 4.5, we have

$$V_{ol}(\boldsymbol{\rho}^*) = |\mathcal{T}_4| \left[\int_0^1 P(\rho; \beta_{\mathcal{R}})\, d\rho + (|\mathcal{R}| - 1) \int_0^\eta P(\rho; \beta_{\mathcal{R}})\, d\rho \right],$$

as the minimum total value of the online solution. For any $\mathcal{T}_i$, since $\left|\mathcal{T}_i'\right| = \lceil \lambda |\mathcal{T}_i| \rceil$, the demand will be satisfied regardless of the user's valuation, unless $\left|\mathcal{T}_i' \cap \mathcal{T}_4\right| > \lceil \frac{\lambda-1}{\lambda} \left|\mathcal{T}_i'\right| \rceil$. In other words, if the demand of user i is not satisfied by the online solution, there must be at least $\lceil \frac{\lambda-1}{\lambda} \left|\mathcal{T}_i'\right| \rceil$ time slots in $\mathcal{T}_i'$ that also belong to $\mathcal{T}_4$; or equivalently, for any $\mathcal{S} \subseteq \mathcal{T}$, and any $\mathcal{T}_i' \subseteq \mathcal{S}$ that is not satisfied by the online solution, $\left|\mathcal{T}_i'\right| < \lfloor \frac{\lambda}{\lambda-1} |\mathcal{S} \cap \mathcal{T}_4| \rfloor$, and hence $|\mathcal{T}_i| < \lfloor \frac{1}{\lambda-1} |\mathcal{S} \cap \mathcal{T}_4| \rfloor$. Let $\mathcal{T}_4'$ be the union of all sets of consecutive time slots that contain $\mathcal{T}_4$, and have a cardinality of $\lfloor \frac{\lambda}{\lambda-1} |\mathcal{T}_4| \rfloor - 1$. When $|\mathcal{T}_i| < \lfloor \frac{1}{\lambda-1} |\mathcal{S} \cap \mathcal{T}_4| \rfloor$, since at least one type of resource in at least one required time slot is fully occupied, there can be a set of users in a worst case, demanding all resources in $\left|\mathcal{T}_4'\right|$ time slots, with $\mathscr{P}_i(\boldsymbol{\rho}) = \overline{p}$, where $\left|\mathcal{T}_4'\right| < 2\lfloor \frac{1}{\lambda-1} |\mathcal{T}_4| \rfloor + |\mathcal{T}_4|$. Thus we have the maximum optimal offline total value

$$V_{opt}(\boldsymbol{\rho}^*) < \left|\mathcal{T}_4'\right| |\mathcal{R}| \underline{p} < \left(2\lfloor \frac{1}{\lambda-1} |\mathcal{T}_4| \rfloor + |\mathcal{T}_4| \right) |\mathcal{R}| \overline{p}.$$

Therefore, for $\boldsymbol{\rho}^* \in \Pi_2$, the worst-cast competitive ratio

$$\alpha^3 < \frac{\frac{\lambda+1}{\lambda-1} |\mathcal{R}| |\mathcal{T}_4| \overline{p}}{|\mathcal{R}| |\mathcal{T}_4| \int_0^\eta P(\rho; \beta_{\mathcal{R}})\, d\rho} = \frac{\lambda+1}{(\lambda-1)\eta'}. \tag{30}$$

□

Theorem 4.9. *The worst-cast competitive ratio achieved by our online resource scheduling strategy using pricing function (28) is upper bounded by $\frac{\lambda+1}{\lambda-1} \max\{\alpha_{\mathcal{R}}/\eta, 1/\eta'\}$, which is a constant with respect to both $|\mathcal{R}|$ and $|\mathcal{T}|$. Here, α_R is defined by Eq.* (5), (16) *or* (19) *for $\beta = \beta_R$, and η is defined as in Assumption 4.*

Proof. The theorem follows immediately from Lemmas 4.6, 4.7 and 4.8. □

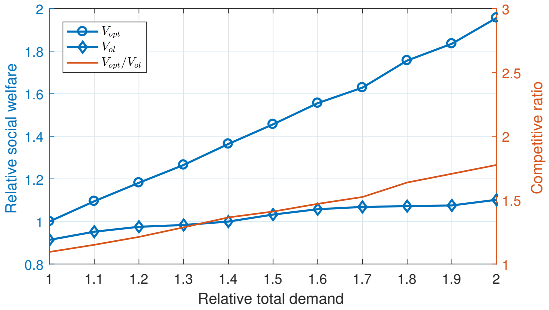


Fig. 5. Online/offline social welfares and competitive ratios given different total demands.

5 EMPIRICAL STUDIES

In this section, we evaluate the proposed pricing and scheduling strategies through simulation studies. To simulate realistic cloud computing scenarios, we relax the assumptions made previously for theoretical analysis. We use a Poisson process to model the arrival of users, with arrival rate between 20 and 50 per time slot. Each user requests 5 time slots on average and 5 different types of resources at most, unless otherwise specified. Each user demands 1 to 3 percent of each type of resource on average,[1] with different standard deviation for different resource types ranging from 0.2% to 2%. We set $\lambda = 1.2$ by default. The time horizon of simulations is 1000 time slots, which is large enough compared to the demand of each user. The statistics of the random input variables are stationary in all cases except the last one (shown in Fig. 8). The optimal offline total values are obtained by solving problem (27) with constraints (27c) and (27d).

By relaxing the assumptions, we can now optimize the parameters in our pricing functions, *e.g.*, β, $\underline{p}$ and $\overline{p}$, to maximize average social welfare. Specifically, we employ pattern search for the optimization: we repeat each experiment for multiple iterations; in the first iteration, we fix the parameters to random estimates; then we add a perturbation (decays with iterations) to each parameter and run the experiment again; a perturbation is retained from one iteration to the next if the total value is improved. In practice, similar probing of parameter values can be done through online learning techniques such as reinforcement learning.

Our theoretical analysis suggests that, under mild assumptions, the worst-case competitive ratio is mainly influenced by the total demand level (Fig. 3), but not by the number of resource types (Theorem 4.5), nor by the number of requested time slots (Theorem 4.9). We now investigate the impact of the three factors on social welfare and competitive ratio, as well as the robustness of the theoretical results, when the assumptions are relaxed.

To quantify different demand levels, we define the relative total demand as the ratio between the total demand of all potential users and the total resource supply Fig. 5 shows that, the optimal offline total value, V_{opt}, increases almost linearly with a slope of 1, as the total demand increases. At the same time, the online total value, V_{ol}, increases with a smaller slope. Consequently, the competitive ratio increases noticeably from 1.09 to 1.78. Although the results depict average system performance (rather than worst-case competitive ratios), it coincides with our worst-case analysis on the scarcity level, β, where larger β leads to a larger competitive ratio (see Fig. 3).

Next, we vary the number of resource types, $|\mathcal{R}|$, from 1 to 10 to see how it affects the competitive ratio. As shown in Fig. 6, due to the increase in total demand and supply, both V_{opt} and V_{ol} increase linearly with $|\mathcal{R}|$, while V_{opt} increases slightly faster than V_{ol}. Consequently, the competitive ratio only increases mildly (from 1.34

[1] These percentages are quite high as compared to practice. We set such percentages to evaluate performance of our pricing functions in the case that Assumption 2 is not true.

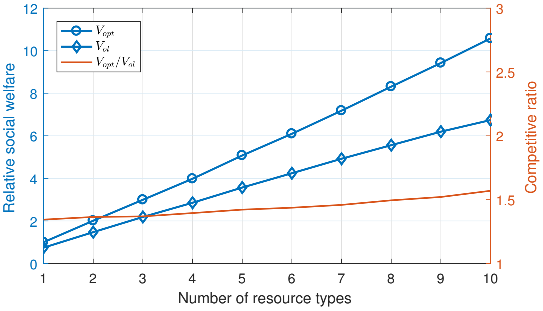
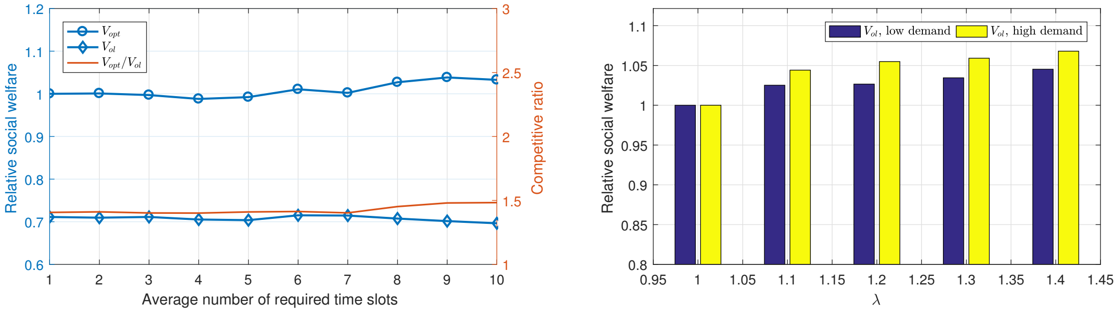


Fig. 6. Online/offline social welfares and competitive ratios given different numbers of resource types.



(a) Online/offline social welfares and competitive ratios given different numbers of required time slots.

(b) Online social welfare at different values of λ.

Fig. 7. Performance of the elastic scheduling strategy discussed in Sec. 4.2.

to 1.57) as $|\mathcal{R}|$ increases. The results may indicate that Assumption 4 is slightly violated in practice, since larger $|\mathcal{R}|$ can increase the chance of unbalanced resource utilization.

Similarly, it is also interesting to see how the number of time slots required by each user affects the competitive ratio. Different from the case of varying $|\mathcal{R}|$, only the total demand will increase with the average number of required time slots. Thus we adjust the demand of each user accordingly to eliminate the effect of increasing relative total demand (Fig. 5). As we can see in Fig. 7a, V_{opt} and V_{ol} remain almost the same as the average number of required time slots increases, and so does the competitive ratio (varying slightly from 1.41 to 1.48). To further verify the proposed strategies, we vary the value of λ from 1 to 1.4 as shown in Fig. 7b. We test the performance for two levels of total demands, 1.5 and 3. In this case, V_{opt} stays almost the same as λ changes and is omitted from the figure. Clearly, V_{ol} increases more from $\lambda = 1$ to $\lambda = 1.2$ than from $\lambda = 1.2$ to $\lambda = 1.4$, indicating $\lambda = 1.2$ is a good trade-off between the availability and timeliness of service.

The simulations conducted so far are based on stationary arrival processes of users. In practice, however, the arrival rate may change over time (*e.g.*, fluctuating periodically). To capture this characteristic, we vary the arrival rate according to a sine curve with a period of 100 time slots. In Fig. 8, as we increase the amplitude of the sine curve from 0 to 1 (normalized by the average arrival rate), both V_{opt} and V_{ol} decrease significantly, while the competitive ratio remains at around 1.5. The reason behind the results is that, when the arrival rate is very low,

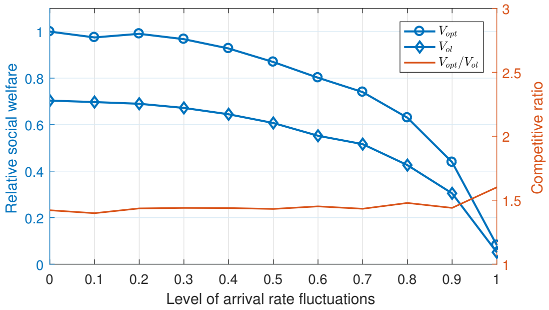


Fig. 8. Online/offline social welfare and competitive ratios given different levels of arrival rate fluctuations.

the resource utilization ratios stay low, and almost all demands can be satisfied; when the arrival rate is very high, a high proportion of the demands cannot be satisfied by either the optimal offline solution or the online solution.

6 CONCLUDING REMARKS

We studied online posted pricing strategies in a number of cloud resource allocation scenarios. We started by investigating the basic case of a single type of cloud resource without recycling, and proved optimality of a set of exponential pricing functions in terms of social welfare, which compute unit resource prices based on realtime demand-supply. Exploiting the insights acquired, we further derived pricing functions in more practical scenarios with multiple resource types and limited resource occupation durations, and proved tight competitive ratio bounds of these functions, without assumptions on user arrival process or valuation distribution. Empirical studies further reveal good performance of our pricing functions under realistic settings. Though set up in a cloud computing environment, our models and algorithms are also applicable to posted pricing in other related online resource allocation problems.

A PROOF OF CLAIM 4.1

Proof. In the worst case of online solution, the valuations of satisfied users are the same as the prices they accept. Thus by Assumption 2, we have

$$V_{ol}\left(\boldsymbol{\rho}^*\right) = \sum_{r\in\mathcal{R}} \int_0^{\rho_r{}^*} P\left(\rho;\beta_r\right)d\rho = \sum_{r\in\mathcal{R}_1} \rho_r{}^*\underline{p} + \sum_{r\in\mathcal{R}_2} \int_0^{\rho_r{}^*} P\left(\rho;\beta_r\right)d\rho, \tag{31}$$

as the minimum total value of the online solution. On the other hand, any unsatisfied user i has an average unit value smaller than $\mathscr{P}_i\left(\boldsymbol{\rho}^*\right)$, otherwise $\boldsymbol{\rho}^*$ cannot be the final resource utilization. We can decompose each user's value as $v_i = \sum_{r\in\mathcal{R}} d_{i,r}U_{i,r}\left(\boldsymbol{\rho}\right)$, and

$$U_{i,r}\left(\boldsymbol{\rho}\right) = \frac{v_i}{d_i\mathscr{P}_i\left(\boldsymbol{\rho}\right)}P\left(\rho_r;\beta_r\right),$$

such that a user i's average unite value $v_i/d_i < \mathscr{P}_i\left(\boldsymbol{\rho}^*\right)$ if and only if $U_{i,r}\left(\boldsymbol{\rho}^*\right) < P\left(\rho_r;\beta_r\right)$, for any $r\in\mathcal{R}$. Here, $U_{i,r}\left(\boldsymbol{\rho}^*\right)$ can be seen as user i's unit value of resource r given a certain $\boldsymbol{\rho}^*$.

For $r\in\mathcal{R}_1$, in the worst case, there can be a set of unsatisfied users with a total demand of $\min\left\{1, 1+\beta_r\right\}$ for each type of resource, and with a unit value $U_{i,r}\left(\boldsymbol{\rho}^*\right) = \underline{p} - \epsilon_r$. Note that $U_{i,r}\left(\boldsymbol{\rho}^*\right) < \underline{p}$ does not contradict with Assumption. 1, since a small enough ϵ_r can ensure $v_i/d_i \geq \underline{p}$. For $r\in\mathcal{R}_2$, the discussion on Eq. (3), (9) for a single resource type is still valid if we consider $U_{i,r}\left(\boldsymbol{\rho}^*\right)$ as unit value of resource; and according to Eq. (6), we have

$$V_{opt}\left({\rho_r}^*\right) = \alpha_r V_{ol}\left({\rho_r}^*\right) - \epsilon_r = \alpha_r \int_0^{{\rho_r}^*} P\left(\rho;\beta_r\right) d\rho) - \epsilon_r.$$

This yields the maximum optimal offline total value given Eq. (31):

$$V_{opt}\left(\boldsymbol{\rho}^*\right) = \sum_{r\in\mathcal{R}_1} \underline{p}\min\left\{1, 1+\beta_r\right\} + \sum_{r\in\mathcal{R}_2} \alpha_r \int_0^{{\rho_r}^*} P\left(\rho;\beta_r\right) d\rho - \epsilon. \tag{32}$$

For $r \in \mathcal{R}_1$, ${\rho_r}^*$ only affects the first term of Eq. (31), while the first term of Eq. (32) is a constant with respect to ${\rho_r}^*$. Thus in any worst case, the first term of Eq. (31) should be minimized, and hence ${\rho_r}^* = 0, \forall r \in \mathcal{R}_1$. For $r \in \mathcal{R}_2$, let $V_{ol}\left({\rho_r}^*\right) = \int_0^{{\rho_r}^*} P\left(\rho;\beta_r\right) d\rho$, we have

$$\alpha\left(\boldsymbol{\rho}^*\right) = \frac{\sup_{\epsilon>0} V_{opt}\left(\boldsymbol{\rho}^*\right)}{V_{ol}\left(\boldsymbol{\rho}^*\right)} \geq \frac{\sum_{r\in\mathcal{R}_2} \alpha_r V_{ol}\left({\rho_r}^*\right)}{\sum_{r\in\mathcal{R}_2} V_{ol}\left({\rho_r}^*\right)} \geq \frac{\alpha_{\underline{r}} \sum_{r\in\mathcal{R}_2} V_{ol}\left({\rho_r}^*\right)}{\sum_{r\in\mathcal{R}_2} V_{ol}\left({\rho_r}^*\right)} = \alpha_{\underline{r}},$$

where $\underline{r} = \arg\min_{r\in\mathcal{R}_2} \alpha_r$. When $|\mathcal{R}_2| \geq 2$, we can iteratively move $\underline{r}$ from $\mathcal{R}_2$ to $\mathcal{R}_1$, and set ${\rho_{\underline{r}}}^* = 0$ without decreasing $\alpha\left(\boldsymbol{\rho}^*\right)$, until $|\mathcal{R}_2| = 1$, since

$$\frac{\sup_{\epsilon>0}\left(V_{opt}\left(\boldsymbol{\rho}^*\right) - \alpha_{\underline{r}} V_{ol}\left({\rho_r}^*\right) - \epsilon + \underline{p}\min\left\{1, 1+\beta_{\underline{r}}\right\}\right)}{V_{ol}\left(\boldsymbol{\rho}^*\right) - V_{ol}\left({\rho_r}^*\right)} \geq \frac{\sup_{\epsilon>0} V_{opt}\left(\boldsymbol{\rho}^*\right)}{V_{ol}\left(\boldsymbol{\rho}^*\right)}.$$

Similarly, for the only $r \in \mathcal{R}_2$, we can decrease ${\rho_r}^*$ to $1/\alpha_r + \epsilon$ without decreasing $\alpha\left(\boldsymbol{\rho}^*\right)$. Therefore, for $\boldsymbol{\rho}^* \in \Omega_2$, there exists a worst case that happens when ${\rho_r}^* = 0$ for $r \in \mathcal{R}_1$, and ${\rho_r}^* = 1/\alpha_r + \epsilon$ for $r \in \mathcal{R}_2$, where $|\mathcal{R}_2| = 1$. □

B PROOF OF CLAIM 4.2

Proof. The worst case of online solution is that the valuations of satisfied users are the same as the prices they accept. Thus by Assumption 2, we have

$$V_{ol}\left(\boldsymbol{\rho}^*\right) = \sum_{r\in\mathcal{R}} \int_0^{{\rho_r}^*} P\left(\rho;\beta_r\right) d\rho = \sum_{r\in\mathcal{R}_3} \int_0^{{\rho_r}^*} P\left(\rho;\beta_r\right) d\rho + \sum_{r\in\mathcal{R}_4} \int_0^{1} P\left(\rho;\beta_r\right) d\rho, \tag{33}$$

as the minimum total value of the online solution. On the other hand, since there is at least one type of resource fully occupied, i.e., $|\mathcal{R}_4| \geq 1$, there can be a case where all subsequent users demand a small amount of resource $r \in \mathcal{R}_4$, making it impossible to satisfy their demands regardless of their valuations. Hence the maximum optimal offline total value

$$V_{opt}\left(\boldsymbol{\rho}^*\right) = \sum_{r\in\mathcal{R}_3} \int_{\rho_r^1}^{\rho_r^2} P\left(\rho;\beta_r\right) d\rho + \sum_{r\in\mathcal{R}_3} \bar{p}\min\left\{1, 1+\beta_r - {\rho_r}^*\right\} + \sum_{r\in\mathcal{R}_4} \alpha_r \int_0^{1} P\left(\rho;\beta_r\right) d\rho, \tag{34}$$

where $\rho_r^1 = \max\left\{0, \beta_r\right\}$ and $\rho_r^2 = \max\left\{\beta_r, {\rho_r}^*\right\}$.

For $r \in \mathcal{R}_3$, Eq. (33) stays the same or increases as any ${\rho_r}^*$ increases, while Eq. (34) stays the same or decreases. Thus there exists a worst case where ${\rho_r}^* = 0, \forall r \in \mathcal{R}_3$. Let $\underline{r} = \arg\min_{r\in\mathcal{R}_4} \alpha_r$. Due to the same reason as discussed for Eq. (31) and Eq. (32), when $|\mathcal{R}_4| \geq 2$, we can iteratively move $\underline{r}$ from $\mathcal{R}_4$ to $\mathcal{R}_3$, and set ${\rho_{\underline{r}}}^* = 0$ without decreasing the competitive ratio, until $|\mathcal{R}_4| = 1$. □

REFERENCES

[1] 2017. Amazon EC2 Spot Instances Pricing. https://aws.amazon.com/ec2/spot/pricing/. (2017).

[2] 2017. Spot Instance Interruptions. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-interruptions.html. (2017).

[3] Orna Agmon Ben-Yehuda, Muli Ben-Yehuda, Assaf Schuster, and Dan Tsafrir. 2013. Deconstructing amazon ec2 spot instance pricing. *ACM Transactions on Economics and Computation* 1, 3 (2013), 16.

[4] May Al-Roomi, Shaikha Al-Ebrahim, Sabika Buqrais, and Imtiaz Ahmad. 2013. Cloud computing pricing models: a survey. *International Journal of Grid and Distributed Computing* 6, 5 (2013), 93–106.

[5] Bo An, Victor Lesser, David Irwin, and Michael Zink. 2010. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 981–988.

[6] Niv Buchbinder and Joseph Naor. 2005. Online primal-dual algorithms for covering and packing problems. In *European Symposium on Algorithms*. Springer, 689–701.

[7] Niv Buchbinder and Joseph Naor. 2006. Improved bounds for online routing and packing via a primal-dual approach. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE.

[8] Niv Buchbinder and Joseph Naor. 2009. The design of competitive online algorithms via a primal: dual approach. *Foundations and Trends® in Theoretical Computer Science* 3, 2–3 (2009), 93–263.

[9] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems* 25, 6 (2009), 599–616.

[10] Yang Cai, Constantinos Daskalakis, and S Matthew Weinberg. 2013. Reducing revenue to welfare maximization: Approximation algorithms and other generalizations. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 578–595.

[11] Deeparnab Chakrabarty, Yunhong Zhou, and Rajan Lukose. 2008. Online knapsack problems. In *Workshop on internet and network economics (WINE)*.

[12] Saurabh Kumar Garg, Steve Versteeg, and Rajkumar Buyya. 2013. A framework for ranking of cloud computing services. *Future Generation Computer Systems* 29, 4 (2013), 1012–1023.

[13] Sijia Gu, Zongpeng Li, Chuan Wu, and Chuanhe Huang. 2016. An Efficient Auction Mechanism for Service Chains in The NFV Market. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE.

[14] Hao Li, Jianhui Liu, and Guo Tang. 2011. A pricing algorithm for cloud computing resources. In *Network Computing and Information Security (NCIS), 2011 International Conference on*, Vol. 1. IEEE, 69–73.

[15] Wei-Yu Lin, Guan-Yu Lin, and Hung-Yu Wei. 2010. Dynamic auction mechanism for cloud resource allocation. In *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*. IEEE, 591–592.

[16] RT Ma, Dah Ming Chiu, John CS Lui, Vishal Misra, and Dan Rubenstein. 2010. On resource management for cloud users: A generalized kelly mechanism approach. *Electrical Engineering, Tech. Rep* (2010).

[17] Sunilkumar S Manvi and Gopal Krishna Shyam. 2014. Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. *Journal of Network and Computer Applications* 41 (2014), 424–440.

[18] Ishai Menache, Asuman Ozdaglar, and Nahum Shimkin. 2011. Socially optimal pricing of cloud computing resources. In *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 322–331.

[19] Marian Mihailescu and Yong Meng Teo. 2010. Dynamic resource pricing on federated clouds. In *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE Computer Society, 513–517.

[20] Mahyar Movahed Nejad, Lena Mashayekhy, and Daniel Grosu. 2015. Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds. *IEEE transactions on parallel and distributed systems* 26, 2 (2015), 594–603.

[21] Weijie Shi, Chuan Wu, and Zongpeng Li. 2014. RSMOA: A revenue and social welfare maximizing online auction for dynamic cloud resource provisioning. In *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*. IEEE, 41–50.

[22] Weijie Shi, Chuan Wu, and Zongpeng Li. 2016. An online mechanism for dynamic virtual cluster provisioning in geo-distributed clouds. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE.

[23] Weijie Shi, Linquan Zhang, Chuan Wu, Zongpeng Li, and Francis Lau. 2014. An online auction framework for dynamic resource provisioning in cloud computing. *ACM SIGMETRICS Performance Evaluation Review* 42, 1 (2014), 71–83.

[24] Adel Nadjaran Toosi, Rodrigo N Calheiros, and Rajkumar Buyya. 2014. Interconnected cloud computing environments: Challenges, taxonomy, and survey. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 7.

[25] Wei Wang, Ben Liang, and Baochun Li. 2013. Revenue maximization with dynamic auctions in IaaS cloud markets. In *Quality of Service (IWQoS), 2013 IEEE/ACM 21st International Symposium on*. IEEE, 1–6.

[26] Hong Xu and Baochun Li. 2013. Dynamic cloud pricing for revenue maximization. *IEEE Transactions on Cloud Computing* 1, 2 (2013), 158–171.

[27] Sharrukh Zaman and Daniel Grosu. 2013. Combinatorial auction-based allocation of virtual machine instances in clouds. *J. Parallel and Distrib. Comput.* 73, 4 (2013), 495–508.
[28] Qi Zhang, Quanyan Zhu, Mohamed Faten Zhani, Raouf Boutaba, and Joseph L Hellerstein. 2013. Dynamic service placement in geographically distributed clouds. *IEEE Journal on Selected Areas in Communications* 31, 12 (2013), 762–772.
[29] Xiaoxi Zhang, Zhiyi Huang, Chuan Wu, Zongpeng Li, and Francis Lau. 2015. Online auctions in IaaS clouds: welfare and profit maximization with server costs. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 43. ACM, 3–15.
[30] Ruiting Zhou, Zongpeng Li, Chuan Wu, and Zhiyi Huang. 2016. An Efficient Cloud Market Mechanism for Computing Jobs With Soft Deadlines. *IEEE/ACM Transactions on Networking* (2016).
[31] Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. 2008. Budget constrained bidding in keyword auctions and online knapsack problems. In *International Workshop on Internet and Network Economics*. Springer, 566–576.

Received March 2017; revised April 2017; accepted May 2017