Online Combinatorial Auctions for Resource Allocation with Supply Costs and Capacity Limits

Xiaoqi Tan, *Member, IEEE,* Alberto Leon-Garcia, *Life Fellow, IEEE,* Yuan Wu, *Senior Member, IEEE,* and Danny H.K. Tsang, *Fellow, IEEE*

***Abstract*—We study a general online combinatorial auction problem in algorithmic mechanism design. A provider allocates multiple types of capacity-limited resources to customers that arrive in a sequential and arbitrary manner. Each customer has a private valuation function on bundles of resources that she can purchase (e.g., a combination of different resources such as CPU and RAM in cloud computing). The provider charges payment from customers who purchase a bundle of resources and incurs an increasing supply cost with respect to the totality of resources allocated. The goal is to maximize the social welfare, namely, the total valuation of customers for their purchased bundles, minus the total supply cost of the provider for all the resources that have been allocated. We adopt the competitive analysis framework and provide posted-price mechanisms with optimal competitive ratios. Our pricing mechanism is *optimal* in the sense that no other online algorithms can achieve a better competitive ratio. We validate the theoretic results via empirical studies of online resource allocation in cloud computing. Our numerical results demonstrate that the proposed pricing mechanism is competitive and robust against system uncertainties and outperforms existing benchmarks.**

***Index Terms*—Combinatorial Auctions, Posted Price, Resource Allocation, Mechanism Design, Online Algorithms**

I. Introduction

Many auction problems involve allocation of distinct types of resources concurrently. For example, customers in auction-based cloud computing platforms can bid on virtual machines or containers with a package of resources such as CPU and RAM. In these problems, customers often have preferences for bundles or combinations of different items, instead of a single one [1]. For this reason, pricing and allocating resources to customers with combinatorial preferences or valuations, termed as combinatorial auctions (CAs) [2] [3], play a critical role in enhancing economic efficiency. This is also considered a hard-core problem in algorithmic mechanism design [2].

In this paper, we study an online version of CAs for resource allocation with supply costs and capacity limits. A single provider allocates multiple types of capacity-limited resources to customers that arrive in a sequential and arbitrary manner. Each customer has a valuation function on possible bundles of resources that she wants to purchase. The provider charges payment from customers who purchase a bundle of resources and incurs an increasing marginal supply cost (i.e., the derivative of the supply cost function) per unit of consumed resource. The goal is to maximize the social welfare, namely, the total valuation of customers for their purchased bundles, minus the supply cost of the provider for all the resources allocated.

When online CAs are subject to increasing supply costs and capacity limits, a fundamental challenge is how to properly price the resources in the absence of future information. Specifically, if the resources are sold too cheaply (i.e., too aggressive), then an excessive portion of them may be purchased by earlier customers with low valuations. This will increase the total cost for the provider and thus the price, which consequently prevents later customers from purchasing the resources even if their valuations are higher than the earlier ones. On the other hand, if the price is set too high (i.e., too conservative), then the provider may lose customers, leading to poor performance as well. This paper tackles this challenge by proposing pricing mechanisms that achieve an optimal balance between aggressiveness and conservativeness without future information, leading to the best-achievable competitive ratios under arbitrary increasing marginal cost functions.

Our results are applicable to a variety of resource allocation problems in the emerging paradigms of networking and computing systems. For example, for auction-based resource allocation in infrastructure-as-a-service clouds, providers can charge their users with a certain payment mechanism while also paying a considerable amount of energy costs to maintain the computing servers [4]. Another example is 5G network slicing, one of the key elements of 5G communications [5]. The ultimate goal of network slicing is to dynamically package different types of network resources (e.g., the base stations and the spectrum channels) for different customers. Here, the network operator needs to consider the cost for providing these resources. In this regard, the model studied in this paper offers a promising option to address such resource allocation problems in 5G network slicing.

A. Related Work

Online CAs without supply costs, which is essentially an online set-packing problem [1], has been widely studied, including online auctions [6], [7], online matching [8] [9], AdWords problems [10], [11], online covering and packing problems [12], [13], and online knapsack problems [14]. Among them, the authors of [6] studied an online CA problem and proposed an $O(\log(v_{\max}/v_{\min}))$-competitive online

X. Tan and A. Leon-Garcia are with the Edward S. Rogers Sr. Dept. of Electrical and Computer Engineering, University of Toronto, Canada. Email: {xiaoqi.tan, alberto.leongarcia}@utoronto.ca.

Y. Wu is with the State Key Lab of Internet of Things for Smart City, University of Macau, Macau, China. Email: yuanwu@um.edu.mo.

D.H.K. Tsang is with the Dept. of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. Email: eetsang@ust.hk.

algorithm when there are $\Omega(\log(v_{\max}/v_{\min}))$ copies of each item and each customer's valuation is assumed to be within the interval of $[v_{\min}, v_{\max}]$. Similar results have also been reported for online knapsack problems [14]. By assuming that the weight of each item is much smaller than the capacity of the knapsack, and that the value-to-weight ratio of each item is bounded within the interval of $[L, U]$, the authors of [14] proposed an algorithm which is $(1 + \ln(U/L))$-competitive.

One of the common assumptions made in the above papers is that the resources can be allocated without incurring an increasing supply cost for the provider. Although this assumption is reasonable for the allocation of digital goods [11], it may not hold for most paradigms of network resource managements, where the production cost or the operational cost is an increasing function of the allocated resources. Motivated by this, Blum et al. [15] pioneered the study of online CAs with an increasing production cost. In this setting, the provider can produce any number of copies of the items being sold (i.e., without capacity limit), but needs to pay an increasing marginal production cost per copy. Blum et al. proposed a pricing scheme called *twice-the-index* for several reasonable marginal production cost functions such as linear, lower-degree polynomial and logarithmic functions. For each of these functions, a constant competitive ratio was derived. Huang et al. [16] later studied a similar problem and achieved a tighter competitive ratio with a unified pricing framework. In particular, for power cost functions, they proved that the optimal competitive ratio can be achieved when there is no capacity limit. In contrast to [15] and [16], in this work we prove that in the capacity-limited case, direct application of the pricing schemes designed in [15] and [16] is suboptimal, and a tighter (and optimal) competitive ratio can be achieved by our newly proposed pricing schemes.

B. Major Contributions

We develop an optimal posted-price mechanism (PPM), dubbed PPM$_\phi$, for online CAs with supply costs and capacity limits. PPM$_\phi$ is optimal in the sense that no other online algorithms can achieve a tighter/better competitive ratio. One of the key elements in PPM$_\phi$ is a strategically-designed pricing function ϕ that determines the selling price based on the current resource utilization levels only. In the general case where the supply cost function is convex and differentiable, we prove that the necessary and sufficient conditions for PPM$_\phi$ to be competitive are related to the existence of an increasing pricing function ϕ for a group of first-order two-point boundary value problems (BVPs) in the field of ordinary differential equations (ODEs) [17], [18]. We derive structural results based on these BVPs that lead to a fundamental characterization of the optimal competitive ratios and the optimal pricing functions. To validate our structural results, we perform a case study when the supply cost function is a power function (e.g., $f(y) = ay^s$), which is an important case widely exploited [15], [16], [19], [20], and show that both the optimal competitive ratios and the corresponding optimal pricing functions can be characterized in analytical forms with some low-complexity numerical computations. Our optimal analytical results for the power cost function improve or generalize the results in several previous studies, e.g., [15], [16], [21], [22]. Moreover, our structural results can also be extended to general settings of online resource allocation with heterogeneous supply costs and multiple time slots.

II. The Basic Resource Allocation Model

This section presents the basic model, the technical assumptions and the definition of competitive ratios for online resource allocation with supply costs and capacity limits.

A. The Basic Model

We consider a single provider who allocates a set $\mathcal{K} = \{1, \cdots, K\}$ of K types of resources to its customers. Each type of resource $k \in \mathcal{K}$ is associated with a cost function $f_k(y)$, where $f_k(y)$ denotes the total supply cost of providing y units of resource k. For example, if resource k represents the computing cycles in cloud/fog/edge computing [23], then the supply cost $f_k(y)$ can represent the electricity cost of maintaining the computing servers. In the following we will also frequently use the derivative of f_k, i.e., the marginal cost function f_k'. *For simplicity of exposition, we assume that the cost functions are identical for all types of resources, and thus we drop the index k and simply use f to denote the supply cost function of all resource types*. Our results are applicable to general cases with heterogeneous cost functions, and we will provide our general results in Section V.

We consider an online setting where customers arrive one at a time in some arbitrary manner. In particular, for a set of customers $\mathcal{N} = \{1, 2, \cdots, N\}$, we denote the arrival time of customer n by t_n. Meanwhile, we assume without loss of generality that $t_1 \leq t_2 \leq \cdots \leq t_N$, where ties are broken arbitrarily if multiple customers arrive simultaneously. Each customer n wants to get a bundle of resources $b \in \mathcal{B}$ based on their own preferences, where $\mathcal{B}$ denotes the set of all the possible bundles (including the empty bundle $\emptyset$). A bundle b of resources is denoted by the vector $(r_1^b, \cdots, r_K^b)$, where r_k^b denotes the number of units for resource $k \in \mathcal{K}$. We consider the case of limited-supply, and normalize the capacity limit to be 1 for each resource type. Therefore, r_k^b is also normalized to be the proportion of the capacity limit accordingly. Each customer n has a private valuation function $v_n : \mathcal{B} \to \mathbb{R}$, where $v_n(b)$ denotes the valuation of customer n for getting bundle $b \in \mathcal{B}$. For simplicity of notations, we denote the valuation by $v_n^b = v_n(b)$ if customer n gets bundle $b \in \mathcal{B}$. In the following we may use v_n^b and $v_n(b)$ interchangeably. We do not make any assumption regarding the valuation functions (except that $v_n(\emptyset) = 0$, i.e., valuation of the empty bundle is zero).

In the standard setting of online CAs, the provider does not have any information about the customers. Upon the arrival of each customer $n \in \mathcal{N}$, the customer reports a valuation function $\hat{v}_n$ to the provider. The valuation function $\hat{v}_n$ may or may not be the true valuation of customer n (i.e., customers may strategically manipulate their bids). The provider collects the valuation function $\hat{v}_n$ from customer n and decides an irrevocable decision about whether to accept this customer or not. The provider will wait for the next customer $n + 1$ if

customer n is rejected (or customer n gets an empty bundle $\emptyset$). Otherwise, the provider needs to determine the payment π_n to be collected from customer n based on the known information (including current valuation function $\hat{v}_n$ and all the previous valuation functions before customer n), and then allocates a bundle $b_n \in \mathcal{B}$ of resources to customer n. The resulting *payment rule* (i.e., the determination of $\{\pi_n\}_{\forall n}$) and the *allocation rule* (i.e., the determination of $\{b_n\}_{\forall n}$) constitute an *online mechanism*. An important economic objective in mechanism design is *incentive compatibility*. Specifically, a mechanism is incentive compatible or truthful if each customer maximizes its own quasilinear utility, i.e., $v_n(b_n) - \pi_n$, by reporting the true valuation function, namely, $\hat{v}_n = v_n$.

The objective is to design the payment rule to incentivize customers to truthfully report their valuation functions, and the allocation rule to maximize the social welfare $\sum_{n\in\mathcal{N}} v_n(b_n) - \sum_{k\in\mathcal{K}} f(y_k)$.

B. Assumptions

We make the following assumptions throughout the paper.

Assumption 1. *The cost function $f(y)$ is differentiable and strictly-convex in $y \in [0,1]$ and $f(0) = 0$.*

If we denote the set of all differentiable and strictly-convex cost functions with $f(0) = 0$ by $\mathcal{F}$, then *Assumption 1 states that we only focus on the cases when $f \in \mathcal{F}$*. In the following we will frequently use the minimum and maximum marginal costs defined as follows:

$$\underline{c} \triangleq f'(0), \overline{c} \triangleq f'(1). \tag{1}$$

Intuitively, if f is known to the provider, then $\underline{c}$ and $\overline{c}$ are known to the provider as well. Note that a given cost function $f \in \mathcal{F}$ always has a strictly-increasing marginal cost f'.

Assumption 2. *For each resource type $k \in \mathcal{K}$, the number of units in each bundle $b \in \mathcal{B}$ is much smaller than the total capacity limit, i.e., $r_k^b \ll 1$.*

Assumption 2 states that allocating a bundle of resources to a single customer does not substantially influence the overall system and market (i.e., each customer's demand is very small), and thus allows us to focus on the online nature of the problem with mathematical convenience. In large-scale systems (e.g., when N is large), Assumption 2 naturally holds.

Assumption 3. *The per-unit-valuation (PUV) of all customers, defined as v_n^b/r_k^b, is upper bounded by $\overline{p}$, namely,*

$$\max_{n\in\mathcal{N}, b\in\mathcal{B}, k\in\mathcal{K}, r_k^b\neq 0} \left\{v_n^b/r_k^b\right\} \le \overline{p}. \tag{2}$$

We will refer to $\overline{p}$ as the *upper bound* hereinafter. Since r_k^b is finite, Assumption 3 states that the outputs of the value function $v_n(\cdot)$ are upper bounded, and thus it helps to eliminate those irrational cases with extremely-high valuations. Alternatively, $\overline{p}$ can be interpreted as the maximum price customers are willing to pay for purchasing a single unit of resource. Throughout the paper we also assume $\overline{p} > \underline{c}$ in order to ensure that the problem setup is interesting. Otherwise, no resources will be allocated.

C. Competitive Analysis

We categorize all the parameters defined previously into the following two groups:

1) The *Setup* $\mathcal{S}$: all the parameters known at the beginning, including the cost function $f \in \mathcal{F}$, the upper bound $\overline{p}$, the set of resource types $\mathcal{K}$, and the set of bundles $\mathcal{B}$.
2) The *Arrival Instance* $\mathcal{A}$: all the parameters revealed over time, including the set of customers $\mathcal{N}$, their arrival times $\{t_n\}_{\forall n\in\mathcal{N}}$, and the valuation functions $\{v_n(\cdot)\}_{\forall n\in\mathcal{N}}$.

An arrival instance $\mathcal{A}$ consists of all the information in the customer side that is not known to the provider a priori. In the offline setting when we assume a complete knowledge of $\mathcal{A}$, the optimal social welfare $W_{\mathsf{opt}}(\mathcal{A})$ can be obtained by solving the following mixed-integer program:

$$W_{\mathsf{opt}}(\mathcal{A}) = \underset{\mathbf{x},\mathbf{y}}{\text{maximize}} \quad \sum_{n\in\mathcal{N}}\sum_{b\in\mathcal{B}} v_n^b x_n^b - \sum_{k\in\mathcal{K}} f(y_k), \tag{3a}$$

$$\text{subject to} \quad \sum_{n\in\mathcal{N}}\sum_{b\in\mathcal{B}} r_k^b x_n^b = y_k, \forall k, \tag{3b}$$

$$\sum_{b\in\mathcal{B}} x_n^b \le 1, \forall n, \tag{3c}$$

$$0 \le y_k \le 1, \forall k, \tag{3d}$$

$$x_n^b \in \{0,1\}, \forall n, b, \tag{3e}$$

where $x_n^b \in \{0,1\}$ is a binary variable that represents the status of bundle b for customer n, and y_k denotes the total resource consumption of resource type k in the end. In particular, $x_n^b = 1$ means that bundle b is allocated to customer n, and $x_n^b = 0$ otherwise. It is possible that $x_n^b = 0$ for all $b \in \mathcal{B}$, meaning that customer n will leave without making any purchase. Constraint (3c) indicates that at most one bundle will be allocated to each customer. Constraint (3d) denotes the normalized capacity limit for resource type $k \in \mathcal{K}$.

In the online setting when customers are revealed one-by-one in a sequential manner, the social welfare performance, denoted by $W_{\mathsf{online}}(\mathcal{A})$, can be quantified via the standard competitive analysis framework [24]. Specifically, an online mechanism is α-competitive if

$$W_{\mathsf{online}}(\mathcal{A}) \ge \frac{1}{\alpha} W_{\mathsf{opt}}(\mathcal{A}) \tag{4}$$

holds for all possible arrival instances $\mathcal{A}$, where $\alpha \ge 1$. Our target is to design an online mechanism such that W_{online} is as close to W_{opt} as possible, i.e., α is as close to 1 as possible.

III. PPM and Structural Results

In this section, we introduce our proposed online mechanism PPM_ϕ, and present the necessary and sufficient conditions for PPM_ϕ to be α-competitive. Based on these conditions, we derive structural results to characterize the minimum value of α.

A. PPM_ϕ: An Overview of How It Works

We focus on the setting of posted-price [25] and propose PPM_ϕ in Algorithm 1. In posted-price, the provider cannot ask the customers to submit their valuation functions, and thus cannot run Vickrey–Clarke–Groves auctions [2]. Instead, the

provider posts prices upon arrival of each customer $n \in \mathcal{N}$, and lets customer n make her own decision on whether to purchase or not based on the posted prices. In this regard, posted-price is privacy-preserving since it does not require customers to reveal their private valuation functions. Meanwhile, by virtue of posted-price, our proposed PPM$_\phi$ is incentive compatible since false reports naturally vanish [25].

In Algorithm 1, at each round when there is a new arrival of customer $n \in \mathcal{N}$, the provider offers her the prices $\{p_k^{(n)}\}_{\forall k}$ by Eq. (6), where ϕ is referred to as the **pricing function** and $y_k^{(n-1)}$ denotes the utilization of resource type $k \in \mathcal{K}$ upon the arrival of customer n, i.e., after processing customer $n-1$. Note that when $n = 1$, the posted price for the first customer is given by $p_k^{(1)} = \phi(y_k^0)$, where $y_k^{(0)}$ is the resource utilization before processing the first customer, and thus is initialized to be zero. Based on the offered prices $\{p_k^{(n)}\}_{\forall k}$, customer n selects the utility-maximizing bundle by solving the problem in Eq. (7) and calculates the potential payment in Eq. (8). If the maximum utility of customer n, i.e., $v_n^{b_*} - \pi_n$, is less than zero (i.e., negative utility), or the capacity limit constraint (3d) is violated, then customer n will leave without purchasing anything[1] and the provider will wait for the next customer $n+1$. Otherwise, customer n will choose bundle b_*. The provider will charge this customer the payment π_n and update the total resource utilization level y_k in Eq. (9). The process repeats upon arrival of customer $n+1$.

The above processes show that the solutions found by PPM$_\phi$, namely, $\{x_n^b\}_{\forall n,b}$ and $\{y_k^{(n)}\}_{\forall n,k}$, are always feasible to Problem (3). Another observation is that the pricing function ϕ plays a critical role in PPM$_\phi$. Indeed, it is ϕ that determines the posted prices in line 3, and then influences each customer's decision-making in line 4-line 8, which ultimately influences the social welfare achieved by PPM$_\phi$, i.e.,

$$W_{\mathsf{online}}(\mathcal{A}) = \sum_{n\in\mathcal{N}} v_n^{b_*} x_n^{b_*} - \sum_{k\in\mathcal{K}} f\big(y_k^{(N)}\big). \tag{5}$$

In Eq. (5), $y_k^{(N)}$ denotes the final resource utilization of resource type $k \in \mathcal{K}$, and $x_n^{b_*}$ denotes the status of the utility-maximized bundle b_* for customer n, i.e., $x_n^{b_*} = 1$ denotes that customer n obtains bundle b_*, and $x_n^{b_*} = 0$ otherwise. Note that both $\{x_n^{b_*}\}_{\forall n}$ and $\{y_k^{(N)}\}_{\forall k}$ depend on the pricing function ϕ, and thus the final competitive ratio of PPM$_\phi$ depends on ϕ as well.

Algorithm 1: PPM with Pricing Function ϕ (PPM$_\phi$)

1: **Input:** Setup $\mathcal{S}$ and ϕ, and initialize $y_k^{(0)} = 0, \forall k$.
2: **while** a new customer n arrives **do**
3: Offer resource $k \in \mathcal{K}$ at price $p_k^{(n)}$ as follows:

$$p_k^{(n)} = \phi(y_k^{(n-1)}). \tag{6}$$

4: Customer chooses the utility-maximizing bundle b_* by solving the following problem:

$$b_* = \arg\max_{b\in\mathcal{B}} \; v_n^b - \sum\nolimits_{k\in\mathcal{K}} p_k^{(n)} r_k^b, \tag{7}$$

where r_k^b denotes the units of resource k in bundle b, and then calculates the potential payment

$$\pi_n = \sum\nolimits_{k\in\mathcal{K}} p_k^{(n)} r_k^{b_*}. \tag{8}$$

5: **if** $v_n^{b_*} - \pi_n < 0$ or $y_k^{(n-1)} + r_k^{b_*} > 1$ holds for any $k \in \mathcal{K}$ **then**
6: Customer n leaves without purchasing anything (i.e., $x_n^b = 0$ for all $b \in \mathcal{B}$).
7: **else**
8: Customer n chooses bundle b_* and pays π_n to the provider (i.e., $x_n^{b_*} = 1$ and $x_n^b = 0$, $\forall b \in \mathcal{B}\backslash\{b_*\}$).
9: Provider updates the resource consumption by

$$y_k^{(n)} = y_k^{(n-1)} + r_k^{b_*}, \forall k \in \mathcal{K}. \tag{9}$$

10: **end if**
11: **end while**

B. Conditions for PPM$_\phi$ to Be α-Competitive

An important result in this paper is the development of the following Theorem 1, which characterizes the sufficient and necessary conditions for the pricing function ϕ such that PPM$_\phi$ can be α-competitive.

Theorem 1. *Given a setup $\mathcal{S}$ with $\overline{p} \in (\underline{c}, +\infty)$, we have:*

- Low-Uncertainty Case (LUC): $\overline{p} \in (\underline{c}, \overline{c}]$.
  - **Sufficiency**. *For any given $\alpha \geq 1$, if $\phi(y)$ is a solution to the following first-order BVP:*

$$\mathsf{L}(\alpha)\begin{cases}\phi'(y) = \alpha \cdot \frac{\phi(y) - f'(y)}{f'^{-1}(\phi(y))}, y \in (0, v),\\ \phi(0) = \underline{c}, \phi(v) \geq \overline{p},\end{cases}$$

where $v \triangleq f'^{-1}(\overline{p})$ and f'^{-1} denotes the inverse of f', then PPM$_\phi$ is α-competitive.
  - **Necessity**. *If there exists an α-competitive online algorithm, then there must exist a strictly-increasing function $\phi(y)$ that satisfies $\mathsf{L}(\alpha)$.*
- High-Uncertainty Case (HUC): $\overline{p} \in (\overline{c}, +\infty)$.
  - **Sufficiency**. *For any given $\alpha \geq 1$, if $\phi(y)$ is a solution to the following two first-order BVPs simultaneously:*

$$\mathsf{H}_1(u,\alpha)\begin{cases}\phi'(y) = \alpha \cdot \frac{\phi(y) - f'(y)}{f'^{-1}(\phi(y))}, y \in (0, u),\\ \phi(0) = \underline{c}, \phi(u) = \overline{c}.\end{cases}$$

$$\mathsf{H}_2(u,\alpha)\begin{cases}\phi'(y) = \alpha \cdot (\phi(y) - f'(y)), y \in (u, 1),\\ \phi(u) = \overline{c}, \phi(1) \geq \overline{p},\end{cases}$$

where $u \in (0,1)$ is the resource utilization level such that $\phi(u) = \overline{c}$, then PPM$_\phi$ is α-competitive.
  - **Necessity**. *If there exists an α-competitive online algorithm, then there must exist a resource utilization level $u \in (0,1)$ and a strictly-increasing function $\phi(y)$ such that $\phi(y)$ satisfies $\{\mathsf{H}_1(u,\alpha), \mathsf{H}_2(u,\alpha)\}$.*

[1] We assume that customers are rational and will not purchase any bundle if they suffer from negative utilities. This is known as the individual rationality in economics and is a common design objective in mechanism design [2].

Proof. The terms LUC and HUC arise from the fact that $\overline{p}$ indicates the uncertainty level of the PUVs in the arrival instance $\mathcal{A}$, namely, a larger $\overline{p}$ implies a wider range of the PUV distribution (note that the PUVs are randomly distributed within $[0, \overline{p}]$ based on Assumption 3), and vice versa. We emphasize that the division into cases LUC and HUC is not artificial, but arise from a principled online primal-dual analysis of Problem (3). The detailed proof is given in [26]. □

Theorem 1 consists of the conditions that are both sufficient and necessary. The sufficiency in Theorem 1 argues that PPM_ϕ is α-competitive as long as the pricing function ϕ is a strictly-increasing solution to the corresponding BVPs in LUC and HUC. Hence, the discussion is within the domain of PPMs. The necessity of Theorem 1 argues that if there exists any α-competitive online algorithm, then there must exist a strictly-increasing solution to the corresponding BVPs. Therefore, the necessity of Theorem 1 is not restricted to PPMs only, and thus is more general.

(**Intuition of Theorem 1**) In Fig. 1, we illustrate two pricing functions for both LUC and HUC. Fig. 1(a) illustrates a special case in LUC when $\phi(v) = \overline{p}$, where $v = f'^{-1}(\overline{p})$ denotes the maximum-possible resource utilization level for PPM_ϕ. Here we use the pricing function illustrated in Fig. 1(a) to briefly explain the intuition behind the BVP of $\mathsf{L}(\alpha)$. The rationality of the two BVPs in HUC follows the same principle. Note that the ODE of $\mathsf{L}(\alpha)$ in Theorem 1 can be reorganized as

$$\phi(y) - f'(y) = \frac{1}{\alpha}\phi'(y)f'^{-1}\left(\phi(y)\right), y \in (0, v). \tag{10}$$

The left-hand-side of Eq. (10) is illustrated by the grey area in Fig. 1(a). Since $f(0) = 0$, $\phi(0) = \underline{c}$, and $\phi(v) = \overline{p}$, integrating both sides of Eq. (10) for $y \in [0, v]$ leads to

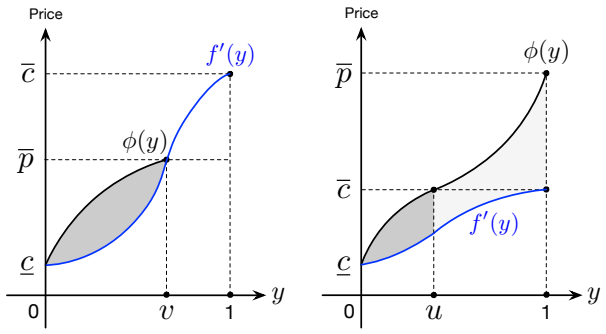
$$\begin{aligned}\int_0^v \phi(y)dy - f(v) &= \frac{1}{\alpha}\int_0^v \phi'(y)f'^{-1}\left(\phi(y)\right)dy \\ &= \frac{1}{\alpha}\int_{\underline{c}}^{\overline{p}} f'^{-1}\left(\phi\right)d\phi.\end{aligned} \tag{11}$$

Notice that the last integration in Eq. (11) is over the inverse of the marginal cost function, which can be solved in analytical form so Eq. (11) is equivalently written as

$$\alpha = \frac{\overline{p}v - f(v)}{\int_0^v \phi(y)dy - f(v)}. \tag{12}$$

Next we show that Eq. (12) essentially captures the worst-case ratio between the optimal offline social welfare and the social welfare achieved by PPM_ϕ under a special arrival instance.

Suppose we have an arrival instance $\mathcal{A}_v$ given as follows: *for all $y \in [0, v]$, there is a continuum of customers, indexed by $y \in [0, v]$, whose valuations are given by $v_y = \phi(y)\Delta y$, where Δy denotes the units of resources that are purchased by customer y and is infinitesimally small. For $y \in (v, 2v]$, there is another continuum of customers whose valuations are given by $v_y = \overline{p}\Delta y$.* Given the arrival instance $\mathcal{A}_v$, PPM_ϕ will accept all the customers indexed by $y \in [0, v]$. Thus, the social welfare achieved by PPM_ϕ is the denominator of the right-hand-side of Eq. (12), namely, the total valuation of all the accepted customers less the supply cost $f(v)$. The optimal offline social welfare, however, is to reject all the customers indexed by $y \in [0, v]$ but only accept the second continuum of customers indexed by $y \in (v, 2v]$. Therefore, the optimal offline social welfare in hindsight is given by $\overline{p}v - f(v)$, which is exactly the numerator of the right-hand-side of Eq. (12). Therefore, a pricing function $\phi(y)$ that satisfies $\mathsf{L}(\alpha)$ leads to the quotient in Eq. (12), which captures the worst-case ratio α between the social welfare achieved by the optimal offline algorithm and PPM_ϕ. Based on the competitive ratio definition in Eq. (4), we can see that PPM_ϕ is α-competitive.



(a) LUC: $\overline{p} \in (\underline{c}, \overline{c}]$ (b) HUC: $\overline{p} \in (\overline{c}, +\infty)$

Fig. 1. Illustration of the pricing function ϕ in LUC and HUC.

(**Dividing Threshold**) Note that for the case of LUC in Fig. 1(a), the capacity limit 1 will never be reached. Otherwise, the system may suffer from negative social welfare (i.e., added valuations are smaller than the increased supply costs). In contrast, Fig. 1(b) illustrates a pricing function in HUC with $\phi(0) = \underline{c}$ and $\phi(1) = \overline{p}$. In this case, the capacity limit 1 can be reached as long as we have enough customers. In particular, there exists a threshold $u \in (0, 1)$ such that $\phi(u) = f'(1) = \overline{c}$. In the following we refer to $u \in (0, 1)$ as the *dividing threshold* of pricing function ϕ. The formal definition is given as follows.

Definition 1. *Given a continuous pricing function ϕ with $\phi(0) < \overline{c}$ and $\phi(1) > \overline{c}$, the dividing threshold of ϕ is the resource utilization level $u \in (0, 1)$ so that $\phi(u) = f'(1) = \overline{c}$.*

In HUC, for any dividing threshold $u \in (0, 1)$, the whole interval of $[0, 1]$ is divided into segments $[0, u]$ and $[u, 1]$. When the lower and upper bounds of ϕ are fixed, e.g., $\phi(0) = \underline{c}$ and $\phi(1) = \overline{p}$ in Fig. 2(b), the dividing threshold u has a strong impact on the curvature of ϕ. A smaller dividing threshold u indicates a steeper pricing curve in $[0, u]$, and thus will perform better for arrival instances with high-PUVs. In contrast, a larger dividing threshold u indicates a less steep pricing curve within $[0, u]$ and thus will perform better for arrival instances with low-PUVs. When there is no future information, we need to find a balance between these two so that the resulting online mechanism PPM_ϕ has a stable performance regardless of arrival instances. Theorem 1 captures this intuition by explicitly discriminating the pricing function design in $[0, u]$ and $[u, 1]$ with two different BVPs in HUC. The next subsection shows that if the dividing threshold u is strategically chosen, the competitive ratio of PPM_ϕ can be minimized.

C. Structural Analysis for Optimal Design

Recall that our objective is to design online mechanisms to achieve the value of α which is as small as possible. To quantify how small α can possibly be, we define the **optimal competitive ratio** in the following Definition 2.

Definition 2. *Given a setup $\mathcal{S}$, the competitive ratio α is optimal if no other online algorithms can achieve a smaller competitive ratio under Assumption 1-Assumption 3.*

Based on the necessity in Theorem 1, to find the optimal competitive ratio for a given setup $\mathcal{S}$, it suffices to find the minimum α so that there exist strictly-increasing solutions to the BVPs in Theorem 1. Hence, we give Proposition 2 below.

Proposition 2. *Given a setup $\mathcal{S}$, if $\alpha_*(\mathcal{S})$ is defined as follows:*

$$\alpha_*(\mathcal{S}) \triangleq \inf \left\{ \alpha \;\middle|\; \begin{array}{l} \text{there exists a strictly-increasing} \\ \text{function } \phi \text{ so that i) if } \overline{p} \in (\underline{c}, \overline{c}], \\ \phi \text{ is a solution to } \mathsf{L}(\alpha), \text{ or ii) if} \\ \overline{p} > \overline{c}, \phi \text{ is a solution to } \mathsf{H}_1(u,\alpha) \\ \text{and } \mathsf{H}_2(u,\alpha)\} \text{ with a feasible} \\ \text{dividing threshold } u \in (0,1). \end{array} \right\},$$

then $\alpha_*(\mathcal{S})$ is the optimal competitive ratio achievable by all online algorithms.

Proposition 2 directly follows the necessity of Theorem 1. Based on Proposition 2, we have the following corollary.

Corollary 3. *Given a setup $\mathcal{S}$, there exists no $(\alpha_*(\mathcal{S}) - \epsilon)$-competitive online algorithm, $\forall \epsilon > 0$.*

Based on Proposition 2, to obtain the optimal competitive ratio $\alpha_*(\mathcal{S})$, we just need to characterize the existence conditions of strictly-increasing solutions to the BVPs in Theorem 1. Note that in LUC, for a given setup $\mathcal{S}$, $\mathsf{L}(\alpha)$ is not indexed by any other parameters except the competitive ratio parameter α, and thus, $\alpha_*(\mathcal{S})$ is the minimum α so that there exists a strictly-increasing solution to $\mathsf{L}(\alpha)$. However, in HUC, both the two BVPs are indexed by the dividing threshold u, which is a design variable that can be flexibly chosen within $(0,1)$. As a result, the minimum α to guarantee the existence of strictly-increasing solutions to $\{\mathsf{H}_1(u,\alpha), \mathsf{H}_2(u,\alpha)\}$ will depend on u. To characterize this dependency, we define the lower bound of α for each given $u \in (0,1)$ as follows.

Definition 3 (Lower Bound of α in HUC). *Given a setup $\mathcal{S}$ with $\overline{p} \in (\overline{c}, +\infty)$, the lower bound of α for any given $u \in (0,1)$, denoted by $\underline{\alpha}(u)$, is defined as follows:*

$$\underline{\alpha}(u) \triangleq \inf \left\{ \alpha \;\middle|\; \begin{array}{l} \text{There exists a strictly-increasing} \\ \text{pricing function } \phi(y) \text{ that is a} \\ \text{solution to } \{\mathsf{H}_1(u,\alpha), \mathsf{H}_2(u,\alpha)\}. \end{array} \right\}.$$

Based on Definition 3, the optimal competitive ratio can be calculated as follows:

$$\alpha_*(\mathcal{S}) = \underline{\alpha}(u_*), \text{ where } u_* = \arg\min_{u \in (0,1)} \underline{\alpha}(u), \tag{13}$$

where u_* denotes the optimal dividing threshold.

Algorithm 2 summarizes the above structural results and provides a principled way to characterize the optimal competitive ratio and the corresponding optimal pricing function for any given setup $\mathcal{S}$. The key steps in Algorithm 2 are line 3 and line 6, in which we need to characterize the conditions for the existence of strictly-increasing solutions to the BVPs in Theorem 1. We emphasize that characterizing such existence conditions heavily depends on the cost function f. The next section will demonstrate how such conditions can be derived in analytical forms when f is a power function.

Algorithm 2: Principles of Optimal Design

1: **Input**: the setup $\mathcal{S}$ with $\overline{p} \in (\underline{c}, +\infty)$.
2: **if** $\overline{p} \in (\underline{c}, \overline{c}]$ **then**
3: Get the minimum α, denoted by $\alpha_*(\mathcal{S})$, so that there exists a strictly-increasing solution to $\mathsf{L}(\alpha)$.
4: Solve $\mathsf{L}(\alpha_*(\mathcal{S}))$ and get the optimal pricing function ϕ so that PPM_ϕ is $\alpha_*(\mathcal{S})$-competitive.
5: **else**
6: Get the lower bound $\underline{\alpha}(u)$ based on Definition 3.
7: Obtain $\alpha_*(\mathcal{S})$ and $u_* \in (0,1)$ based on Eq. (13).
8: Solve $\{\mathsf{H}_1(u_*, \underline{\alpha}(u_*)), \mathsf{H}_2(u_*, \underline{\alpha}(u_*))\}$ and get the optimal pricing function ϕ so that PPM_ϕ is $\alpha_*(\mathcal{S})$-competitive or $\underline{\alpha}(u_*)$-competitive.
9: **end if**
10: **Output**: $\alpha_*(\mathcal{S})$ and optimal pricing functions.

IV. Case Study: $f(y) = ay^s$

We now perform a case study for $f(y) = ay^s$ (i.e., power function), and show how to use Algorithm 2 to obtain the minimum value of α, the optimal dividing threshold u_*, and the corresponding optimal pricing functions. At the end of this section, we will discuss some important structural properties about the optimal pricing functions.

A. Preliminaries: The BVPs in Both Cases

We consider $f(y) = ay^s$ with $a > 0$ and $s > 1$ so that the marginal cost $f'(y) = asy^{s-1}$ is strictly increasing. Such power cost functions are often used for modeling the costs that are diseconomies-of-scale (i.e., no volume discounts). For example, when $s \geq 2$, $f(y)$ is a classic power-rate curve, reflecting the power consumption of a general networking and computing device with the capability of speed-scaling [19], [20], e.g., CPU, edge router, and communication link. It is also common to use $s = 1 \sim 3$ to model the power consumption of data centers in cloud computing [4], [21].

When $f(y) = ay^s$, the minimum marginal cost is $\underline{c} = f'(0) = 0$ and the maximum marginal cost is $\overline{c} = f'(1) = as$. Based on Theorem 1, $\mathsf{L}(\alpha)$, $\mathsf{H}_1(u,\alpha)$, and $\mathsf{H}_2(u,\alpha)$ can be written as follows:

- LUC: $\overline{p} \in (\underline{c}, \overline{c}]$. $\mathsf{L}(\alpha)$ is given by

$$\begin{cases} \phi'(y) = \alpha \cdot \frac{\phi(y) - f'(y)}{(\phi(y)/\overline{c})^{\frac{1}{s-1}}}, y \in (0, v), \\ \phi(0) = 0, \phi(v) \geq \overline{p}, \end{cases} \tag{14}$$

where $v = f'^{-1}(\overline{p}) = (\overline{p}/\overline{c})^{\frac{1}{s-1}}$.

- HUC: $\overline{p} \in (\overline{c}, +\infty)$. $\{\mathsf{H}_1(u,\alpha), \mathsf{H}_2(u,\alpha)\}$ are given by

$$\begin{cases} \phi'(y) = \alpha \cdot \frac{\phi(y) - f'(y)}{(\phi(y)/\overline{c})^{\frac{1}{s-1}}}, y \in (0,u), \\ \phi(0) = 0, \phi(u) = \overline{c}, \end{cases} \tag{15a}$$

$$\begin{cases} \phi'(y) = \alpha \cdot \left(\phi(y) - \overline{c} y^{s-1}\right), y \in (u,1), \\ \phi(u) = \overline{c}, \phi(1) \geq \overline{p}, \end{cases} \tag{15b}$$

  where Problem (15a) corresponds to $\mathsf{H}_1(u,\alpha)$, and Problem (15b) corresponds to $\mathsf{H}_2(u,\alpha)$.

Following lines 3 and 6 in Algorithm 2, the next subsection will characterize the conditions for the existence of strictly-increasing solutions to the BVPs in Eq. (14) and Eq. (15).

B. Lower Bound of α in LUC and HUC

1) Lower Bound of α in LUC: We first focus on LUC and give the following Theorem 4.

Theorem 4. *Given a setup $\mathcal{S}$ with $f(y) = ay^s$ and $\overline{p} \in (\underline{c}, \overline{c}]$, there exist strictly-increasing solutions to Problem* (14) *if and only if $\alpha \geq \alpha_s^{\min}$, where $\alpha_s^{\min} = s^{\frac{s}{s-1}}$.*

Theorem 4 provides the lower bound of α so that there exists a strictly-increasing solution to Problem (14) above. Based on Proposition 2, we can conclude that the optimal competitive ratio $\alpha_*(\mathcal{S}) = \alpha_s^{\min}$. According to line 4 in Algorithm 2, the design of optimal pricing functions in LUC is equivalent to solving Problem (14) with $\alpha = \alpha_*(\mathcal{S}) = \alpha_s^{\min}$. In Section IV-D, we will discuss how to solve Problem (14) to get a set of infinitely-many optimal pricing functions.

2) Lower Bound of α in HUC: Theorem 5 below summarizes a necessary and sufficient condition for α such that we can guarantee the existence of a strictly-increasing solution to Problem (15a) and this solution is unique.

Theorem 5. *Given a setup $\mathcal{S}$ with $f(y) = ay^s$ and $\overline{p} > \overline{c}$, for any $u \in (0,1)$, there exists a unique strictly-increasing solution to Problem* (15a) *if and only if $\alpha \geq \underline{\alpha}_1(u)$, where $\underline{\alpha}_1(u)$ is given by*

$$\underline{\alpha}_1(u) = \begin{cases} \alpha_s(u) & \text{if } u \in (0, u_s), \\ \alpha_s^{\min} & \text{if } u \in [u_s, 1). \end{cases} \tag{16}$$

In Eq. (16)*, $\alpha_s(u)$ and u_s are given as follows:*

$$\alpha_s(u) = \frac{s-1}{u - u^s}, u_s = \left(\frac{1}{s}\right)^{\frac{1}{s-1}}. \tag{17}$$

Proof. The proof of the above two theorems is non-trivial since the right-hand-side of the ODE in Problem (15a) (also Problem (14)) has a singular boundary condition at $\phi(0) = 0$ [17]. The detailed proof is given in [26]. □

Theorem 5 provides a lower bound of α for each given dividing threshold u. Note that $\alpha_s(u_s) = \alpha_s^{\min}$. Thus, $\underline{\alpha}_1(u)$ is continuous in $u \in (0,1)$. Meanwhile, $\underline{\alpha}_1(u)$ is non-increasing in $u \in (0,1)$ and achieves its minimum $\alpha_s^{\min}$ when $u \in [u_s, 1)$. However, we cannot directly conclude that the optimal competitive ratio in HUC is also $\alpha_s^{\min}$. This is because it is unclear whether there exists any strictly-increasing solution to Problem (15b) when $u \in [u_s, 1)$ and $\alpha = \alpha_s^{\min}$. To answer this question, below we give Theorem 6.

Theorem 6. *Given a setup $\mathcal{S}$ with $f(y) = ay^s$ and $\overline{p} > \overline{c}$, for any $u \in (0,1)$, there exists a unique strictly-increasing solution to Problem* (15b) *if and only if $\alpha \geq \underline{\alpha}_2(u)$, where $\underline{\alpha}_2(u)$ is the unique root to the following equation*

$$\int_{u\underline{\alpha}_2(u)}^{\underline{\alpha}_2(u)} \eta^{s-1} e^{-\eta} d\eta = \frac{\left(\underline{\alpha}_2(u)\right)^{s-1}}{\exp(u\underline{\alpha}_2(u))} - \frac{\overline{p}\left(\underline{\alpha}_2(u)\right)^{s-1}}{\overline{c}\exp(\underline{\alpha}_2(u))}. \tag{18}$$

Meanwhile, $\underline{\alpha}_2(u)$ is strictly-increasing in $u \in (0,1)$.

Proof. The proof of the lower bound $\underline{\alpha}_2(u)$ is trivial since the ODE in Problem (15b) can be solved in analytical forms. The detailed proof is given in [26]. □

Based on Theorem 5 and Theorem 6, to guarantee the existence of strictly-increasing solutions to Problem (15a) and Problem (15b) simultaneously, α must be jointly lower bounded by $\underline{\alpha}_1(u)$ and $\underline{\alpha}_2(u)$ for all $u \in (0,1)$. Therefore, the lower bound of α is given by

$$\underline{\alpha}(u) = \max\left\{\underline{\alpha}_1(u), \underline{\alpha}_2(u)\right\}, \forall u \in (0,1), \tag{19}$$

which follows our definition of $\underline{\alpha}(u)$ in Definition 3. Note that if $\mathcal{R}(u,\alpha) \subset (0,1) \times [1, +\infty)$ is defined as follows:

$$\mathcal{R}(u,\alpha) \triangleq \{(u,\alpha) | \alpha \geq \underline{\alpha}(u), u \in (0,1)\}. \tag{20}$$

Then, for any given $(u,\alpha) \in \mathcal{R}(u,\alpha)$, the resulting BVPs $\{\mathsf{H}_1(u,\alpha), \mathsf{H}_2(u,\alpha)\}$ must have a strictly-increasing solution. For this reason, we will refer to $\mathcal{R}(u,\alpha)$ as the *achievable region* of (u,α).

Based on line 7 in Algorithm 2, to get the optimal competitive ratio $\alpha_*(\mathcal{S})$ in HUC, we need to find the optimal dividing threshold u_* by solving the following problem

$$u_* = \arg\min_{u \in (0,1)} \underline{\alpha}(u) = \arg\min_{u \in (0,1)} \max\left\{\underline{\alpha}_1(u), \underline{\alpha}_2(u)\right\},$$

where $\underline{\alpha}_1(u)$ is analytically given in Eq. (16), and $\underline{\alpha}_2(u)$ is the unique root to Eq. (18). The next section will show that the optimal dividing threshold u_* always exists. However, the uniqueness of u_* depends on the value of $\overline{p}$.

C. Optimal Competitive Ratios

To characterize the optimal dividing threshold u_*, we give the following Proposition 7 to show the unique existence of an intersection point between $\underline{\alpha}_1(u)$ and $\underline{\alpha}_2(u)$, which we refer to as the **critical dividing threshold** (CDT), denoted by u_{cdt}.

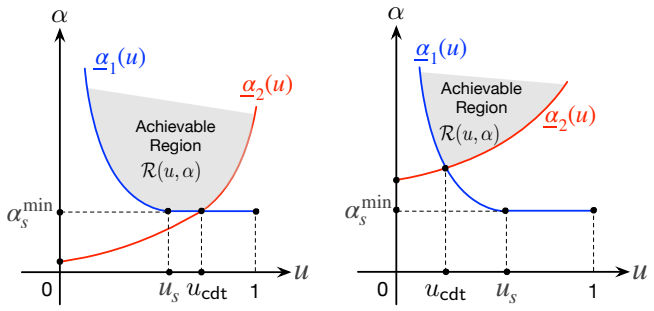
Proposition 7. *Given a setup $\mathcal{S}$ with $f(y) = ay^s$ and $\overline{p} \in (\overline{c}, +\infty)$, there exists a unique CDT $u_{\mathsf{cdt}} \in (0,1)$ such that $\underline{\alpha}_1(u_{\mathsf{cdt}}) = \underline{\alpha}_2(u_{\mathsf{cdt}})$. Specifically, if we define C_s by*

$$C_s \triangleq \overline{c} \cdot \left(\frac{1}{e^s} - \frac{1}{s^s} \cdot \int_s^{\alpha_s^{\min}} \eta^{s-1} e^{-\eta} d\eta\right) \cdot \exp(\alpha_s^{\min}), \tag{21}$$

then the unique CDT can be calculated as follows:

- HUC$_1$: $\overline{p} \in (\overline{c}, C_s]$. *In this case, the CDT is the unique root to the following equation in variable $u_{\mathsf{cdt}} \in [u_s, 1)$:*

$$\int_{u_{\mathsf{cdt}} \cdot \alpha_s^{\min}}^{\alpha_s^{\min}} \eta^{s-1} e^{-\eta} d\eta = \frac{s^s}{\exp(u_{\mathsf{cdt}} \cdot \alpha_s^{\min})} - \frac{\overline{p} s^s}{\overline{c}\exp(\alpha_s^{\min})}.$$



(a) HUC_1: $\overline{p} \in (\overline{c}, C_s]$ (b) HUC_2:$\overline{p} \in (C_s, +\infty)$

Fig. 2. Illustration of the two lower bounds $\underline{\alpha}_1(u)$, $\underline{\alpha}_2(u)$, and $\mathcal{R}(u,\alpha)$.

- HUC_2: $\overline{p} \in (C_s, +\infty)$. *In this case, the CDT is the unique root to the following equation in variable* $u_{\mathsf{cdt}} \in (0, u_s)$*:*

$$\int_{u_{\mathsf{cdt}}\cdot\alpha_s(u_{\mathsf{cdt}})}^{\alpha_s(u_{\mathsf{cdt}})} \eta^{s-1}e^{-\eta}d\eta = \frac{(\alpha_s(u_{\mathsf{cdt}}))^{s-1}}{\exp(u_{\mathsf{cdt}}\cdot\alpha_s(u_{\mathsf{cdt}}))} - \frac{\overline{p}\cdot(\alpha_s(u_{\mathsf{cdt}}))^{s-1}}{\overline{c}\cdot\exp(\alpha_s(u_{\mathsf{cdt}}))}.$$

Proof. This corollary follows the previous two theorems regarding the lower bound $\underline{\alpha}_1(u)$ and $\underline{\alpha}_2(u)$. The detailed proof is given in [26]. □

Fig. 2 illustrates $\underline{\alpha}_1(u)$ and $\underline{\alpha}_2(u)$ in two cases. As can be seen from Fig. 2(a), in HUC_1 (i.e., $\overline{p} \in (\overline{c}, C_s]$), the CDT $u_{\mathsf{cdt}} \in [u_s, 1)$, and the optimal competitive ratio $\alpha_*(\mathcal{S}) = \underline{\alpha}(u_{\mathsf{cdt}}) = \alpha_s^{\min}$. In this case, any dividing threshold $u \in [u_s, u_{\mathsf{cdt}}]$ and $\alpha = \alpha_s^{\min}$ will determine an optimal pricing function that satisfies Problem (15a) and Problem (15b). Therefore, the optimal dividing threshold u_* is not unique and can be any value within the interval $[u_s, u_{\mathsf{cdt}}]$. In comparison, as shown in Fig. 2(b), in HUC_2 (i.e., $\overline{p} \in (C_s, +\infty)$), the unique CDT u_{cdt} is within the interval $(0, u_s)$ and is the unique optimal dividing threshold (i.e., $u_* = u_{\mathsf{cdt}}$ and $\alpha_*(\mathcal{S}) = \alpha_s(u_{\mathsf{cdt}})$). In this case, the optimal pricing function is the unique solution to Problem (15a) and Problem (15b) with $u = u_{\mathsf{cdt}}$ and $\alpha = \alpha_s(u_{\mathsf{cdt}})$.

Corollary 8 summarizes the optimal competitive ratios in LUC and the two sub-cases in HUC.

Corollary 8. *Given a setup $\mathcal{S}$ with $f(y) = ay^s$, the optimal competitive ratio $\alpha_*(\mathcal{S})$ is given by*

$$\alpha_*(\mathcal{S}) = \begin{cases} s^{\frac{s}{s-1}} & \text{if } \overline{p} \in (\underline{c}, \overline{c}], \quad (\mathsf{LUC}) \\ s^{\frac{s}{s-1}} & \text{if } \overline{p} \in (\overline{c}, C_s], \quad (\mathsf{HUC}_1) \\ \frac{s-1}{u_{\mathsf{cdt}} - u_{\mathsf{cdt}}^s} & \text{if } \overline{p} \in (C_s, +\infty), (\mathsf{HUC}_2) \end{cases} \tag{22}$$

where C_s and u_{cdt} can be calculated based on Proposition 7.

The optimal competitive ratio in LUC directly follows Theorem 4, and the optimal competitive ratios in HUC_1 and HUC_2 follow Theorem 5, Theorem 6, and Proposition 7. Note that the first two cases of LUC and HUC_1 in Eq. (22) can be combined together. However, we keep the current three-case form so that it clearly distinguishes LUC and HUC.

D. Optimal Pricing Functions

Based on Corollary 8 and Algorithm 2: i) to get the optimal pricing function for LUC, we need to solve $\mathsf{L}(\alpha)$ with $\alpha = s^{\frac{s}{s-1}}$; ii) to get the optimal pricing function for HUC_1, we need to solve $\{\mathsf{H}_1(u,\alpha), \mathsf{H}_2(u,\alpha)\}$ with any $u \in [u_s, u_{\mathsf{cdt}}]$ and $\alpha = s^{\frac{s}{s-1}}$; iii) to get the optimal pricing function for HUC_2, we need to solve $\{\mathsf{H}_1(u,\alpha), \mathsf{H}_2(u,\alpha)\}$ with $u = u_{\mathsf{cdt}}$ and $\alpha = \frac{s-1}{u_{\mathsf{cdt}} - u_{\mathsf{cdt}}^s}$.

To help characterize the optimal pricing functions for the above three cases, we first focus on the following first-order initial value problem (IVP):

$$\begin{cases} \phi'_{\mathsf{ivp}}(y) = \alpha \cdot \left(\phi_{\mathsf{ivp}}(y) - \overline{c}y^{s-1}\right), y \in (u, 1), \\ \phi_{\mathsf{ivp}}(u) = \overline{c}. \end{cases} \tag{23}$$

Problem (23) is the same as Problem (15b) if we exclude the second boundary condition $\phi(1) \geq \overline{p}$. Based on the Picard-Lindelöf theorem [17], [18], the IVP in Eq. (23) always has a unique strictly-increasing solution for all $\alpha \in \mathbb{R}$. We solve Problem (23) with $\alpha = \underline{\alpha}_1(u)$, and denote the unique solution by $\phi_{\mathsf{ivp}}(y; u)$ as follows:

$$\phi_{\mathsf{ivp}}(y;u) = \overline{c}\cdot\frac{\exp\left(y\cdot\underline{\alpha}_1(u)\right)}{\left(\underline{\alpha}_1(u)\right)^{s-1}}\cdot\int_{y\underline{\alpha}_1(u)}^{\underline{\alpha}_1(u)u}\eta^{s-1}e^{-\eta}d\eta + \overline{c}\cdot\exp\left((y-u)\cdot\underline{\alpha}_1(u)\right), y \in [u, 1]. \tag{24}$$

Intuitively, if $\phi_{\mathsf{ivp}}(1;u) \geq \overline{p}$, then $\phi_{\mathsf{ivp}}(y;u)$ is also a solution to Problem (15b). Below in Lemma 9 we show that $\phi_{\mathsf{ivp}}(1;u) \geq \overline{p}$ holds as long as $u \in [u_s, u_{\mathsf{cdt}}]$.

Lemma 9. *Given $\overline{p} \in (\overline{c}, +\infty)$, for any $u \in [u_s, u_{\mathsf{cdt}}]$, $\phi_{\mathsf{ivp}}(y;u)$ is a solution to Problem* (15b) *with $\phi_{\mathsf{ivp}}(1;u) \geq \overline{p}$.*

We also give the following lemma to show the existence of a unique resource utilization level ρ_s such that $\phi_{\mathsf{ivp}}(\rho_s; u_s) = \overline{p}$.

Lemma 10. *If the value of ρ_s leads to $\phi_{\mathsf{ivp}}(\rho_s; u_s) = \overline{p}$, then ρ_s is the unique root to the following equation:*

$$\int_s^{\alpha_s^{\min}\rho_s}\eta^{s-1}e^{-\eta}d\eta = \frac{s^s}{\exp(s)} - \frac{\overline{p}s^s}{\overline{c}\cdot\exp(\alpha_s^{\min}\rho_s)}. \tag{25}$$

The proofs of the above two lemmas are given in [26]. Based on Eq. (24), Lemma 9, and Lemma 10 above, we next give Theorem 11 which summarizes the optimal pricing functions for all cases of LUC, HUC_1, and HUC_2.

Theorem 11. *Given a setup $\mathcal{S}$ with $f(y) = ay^s$, the optimal pricing functions for PPM$_\phi$ are determined as follows.*

- LUC: $\overline{p} \in (\underline{c}, \overline{c}]$. *Let us define $w \triangleq f'^{-1}(\overline{p}/s)$, then we have $0 < w < v \leq 1$, where $v = f'^{-1}(\overline{p})$. For any $m \in [w, v]$, PPM$_{\phi_m}$ achieves the optimal competitive ratio of $s^{\frac{s}{s-1}}$ if ϕ_m is given by:*

$$\phi_m(y) = \begin{cases} 0 & \text{if } y = 0, \\ \overline{c}\left(\varphi_{\mathsf{luc}}(y)\right)^{s-1} & \text{if } y \in (0, m], \end{cases} \tag{26}$$

where for each given $y \in (0, m]$, $\varphi_{\mathsf{luc}}(y)$ is the unique root to the following equation in variable $\varphi_{\mathsf{luc}} \in (0, 1]$:

$$\int_{1/m}^{\varphi_{\mathsf{luc}}/y}\frac{\eta^{s-1}}{\eta^s - \frac{\alpha_s^{\min}}{s-1}\eta^{s-1} + \frac{\alpha_s^{\min}}{s-1}}d\eta = \ln\left(\frac{m}{y}\right). \tag{27}$$

Meanwhile, when $m = w = f'^{-1}(\overline{p}/s)$*, the optimal pricing function* $\phi_w(y)$ *is given by*

$$\phi_w(y) = sf'(y), y \in [0, w]. \tag{28}$$

- HUC_1: $\overline{p} \in (\overline{c}, C_s]$. *In this case, the CDT* $u_{\mathsf{cdt}} \in [u_s, 1)$*, and for each* $u \in [u_s, u_{\mathsf{cdt}}]$*, PPM*$_{\phi_u}$ *achieves the optimal competitive ratio of* $s^{\frac{s}{s-1}}$ *if* ϕ_u *is given by:*

$$\phi_u(y) = \begin{cases} 0 & \text{if } y = 0, \\ \overline{c}\big(\varphi_{\mathsf{huc}}(y)\big)^{s-1} & \text{if } y \in (0, u), \\ \phi_{\mathsf{ivp}}\big(y; u\big) & \text{if } y \in [u, \rho], \end{cases} \tag{29}$$

where for any given $y \in (0, u)$*,* $\varphi_{\mathsf{huc}}(y)$ *is the unique root to the following equation in variable* $\varphi_{\mathsf{huc}} \in (0, 1)$*:*

$$\int_{1/u}^{\varphi_{\mathsf{huc}}/y} \frac{\eta^{s-1}}{\eta^s - \frac{\alpha_s^{\min}}{s-1}\eta^{s-1} + \frac{\alpha_s^{\min}}{s-1}} \, d\eta = \ln\left(\frac{u}{y}\right). \tag{30}$$

In Eq. (29)*,* $\rho \in [\rho_s, 1]$ *is the maximum resource utilization level that satisfies* $\phi_{\mathsf{ivp}}(\rho; u) = \overline{p}$*, where* ρ_s *is given by Lemma 10. In particular, if* $u = u_s$*, then* $\rho = \rho_s$*; if* $u = u_{\mathsf{cdt}}$*, then* $\rho = 1$*. Meanwhile, if* $u = u_s$*, the optimal pricing function* $\phi_{u_s}(y)$ *can be given analytically by*

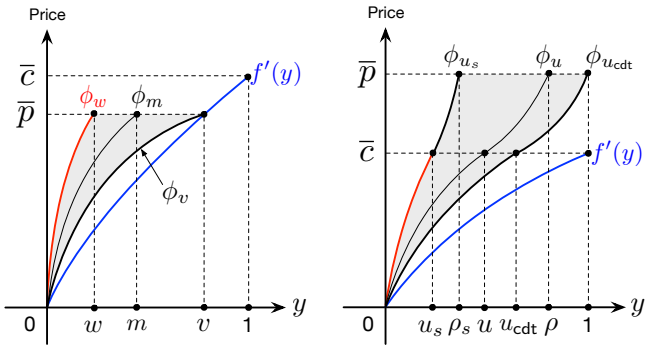
$$\phi_{u_s}(y) = \begin{cases} sf'(y) & \text{if } y \in [0, u_s), \\ \phi_{\mathsf{ivp}}\big(y; u_s\big) & \text{if } y \in [u_s, \rho_s]. \end{cases} \tag{31}$$

- HUC_2: $\overline{p} \in (C_s, +\infty)$. *In this case, the CDT* $u_{\mathsf{cdt}} \in (0, u_s)$*, and PPM*$_{\phi_{u_{\mathsf{cdt}}}}$ *achieves the optimal competitive ratio of* $\frac{s-1}{u_{\mathsf{cdt}} - u_{\mathsf{cdt}}^s}$ *if and only if* $\phi_{u_{\mathsf{cdt}}}$ *is given by:*

$$\phi_{u_{\mathsf{cdt}}}(y) = \begin{cases} f'\left(\frac{y}{u_{\mathsf{cdt}}}\right), & \text{if } y \in [0, u_{\mathsf{cdt}}], \\ \phi_{\mathsf{ivp}}(y; u_{\mathsf{cdt}}), & \text{if } y \in [u_{\mathsf{cdt}}, 1). \end{cases} \tag{32}$$

Proof. The optimal pricing functions in the above three cases are derived by solving the corresponding BVPs in Eq. (14) and Eq. (15). The details are given in [26]. □

For Theorem 11 we make the following two points. First, the optimal pricing functions in Eq. (26) and Eq. (29) have a separated case when $y = 0$. This is because Eq. (27) and Eq. (30) are not defined at $y = 0$. However, we can prove that both φ_{luc} and φ_{huc} approach 0 from the right when $y \to 0^+$, and thus both $\phi_m(y)$ and $\phi_u(y)$ are right-differentiable at $y = 0$, which is consistent with the ODEs in Eq. (14) and Eq. (15). Second, we emphasize that although many parameters in Theorem 11 are in analytical forms (e.g., $u_s, \alpha_s^{\min}$, and $\phi_{\mathsf{ivp}}\big(y; u\big)$, etc.), numerical computations of $u_{\mathsf{cdt}}, \varphi_{\mathsf{luc}}$, and φ_{huc} are still needed. In particular, the CDT u_{cdt} can be calculated offline, while the computations of φ_{luc} and φ_{huc} must be performed in real-time (i.e., "on-the-fly"). This should not be a concern for the online implementation of PPM$_\phi$ since these computations are light-weight (e.g., all the root-finding can be performed efficiently by bisection searching).



(a) LUC: $\overline{p} \in (\underline{c}, \overline{c}]$ (b) HUC_1: $\overline{p} \in (\overline{c}, C_s]$

Fig. 3. Illustration of the optimal pricing functions in LUC and HUC_1. The two red curves represent the same function $sf'(y)$ but with different domains.

E. Discussion of Structural Properties

Fig. 3 illustrates the optimal pricing functions for LUC and HUC_1. We do not illustrate the unique optimal pricing function for HUC_2 since it is similar to Fig. 1(b). We discuss several interesting structural properties revealed by Theorem 11.

(**Aggressiveness of Pricing Functions**) In both LUC and HUC_1, the optimal pricing functions are non-unique, while the optimal pricing function is unique in HUC_2. In particular, the optimal pricing functions for LUC and HUC_1 can be represented by two infinite sets of functions as follows:

$$\Omega_{\mathsf{luc}} = \{\phi_m\}_{\forall m \in [w, v]}, \Omega_{\mathsf{huc}_1} = \{\phi_u\}_{\forall u \in [u_s, u_{\mathsf{cdt}}]}, \tag{33}$$

where ϕ_m and ϕ_u are given by Eq. (26) and Eq. (29), respectively. Graphically, these two sets cover the grey area in Fig. 3. Specifically, as shown in Fig. 3(a), all the optimal pricing functions in Ω_{luc} are lower bounded by ϕ_v and upper bounded by ϕ_w. Similarly, in HUC_1, all the optimal pricing functions in Ω_{huc_1} are lower bounded by $\phi_{u_{\mathsf{cdt}}}$ and upper bounded by ϕ_{u_s}. In economics, if a pricing scheme 'A' sets the price cheaper than pricing scheme 'B', then we say pricing scheme 'A' is more aggressive than pricing scheme 'B' [27]. In this regard, $\phi_{u_{\mathsf{cdt}}}$ (ϕ_v) is the most aggressive optimal pricing function in HUC_1 (LUC), that is, ϕ_{u_s} (ϕ_w) is the most conservative optimal pricing function in HUC_1 (LUC). Interestingly, the pricing scheme proposed by [16] for the same setup of power cost functions is $\phi_w(y) = sf'(y)$ (i.e., the red curves in Fig. 3), which is only a special case of all the optimal pricing functions characterized in Ω_{luc} and Ω_{huc_1}. Moreover, in HUC_2, Theorem 11 shows that the pricing scheme ϕ_w is suboptimal when $\overline{p}$ is larger than C_s. Therefore, our optimal pricing functions in Theorem 11 generalize and improve the results in [16].

(**Pricing at Multiple-the-Index**) Note that the pricing function ϕ_w in LUC and the first segment of ϕ_{u_s} in HUC_1 can be written as $sf'(y) = f'\big(s^{\frac{1}{s-1}}y\big)$, which uses the marginal cost function f' to price the resource at $s^{\frac{1}{s-1}}$-multiple-the-index, and the multiplicative factor $s^{\frac{1}{s-1}} \in (e, 1)$ when $s > 1$. In HUC_2, the optimal pricing function $\phi_{u_{\mathsf{cdt}}}$ also prices the resources at $\frac{1}{u_{\mathsf{cdt}}}$-multiple-the-index of $f'(y)$ when $y \in [0, u_{\mathsf{cdt}}]$. The development of such pricing schemes is not entirely new

in algorithmic mechanism design. For example, for similar setups of online CAs with supply or production costs (but without capacity limits), the authors of [15] proposed a pricing scheme called "twice-the-index" (i.e., $\phi(y) = f'(2y)$), and the authors of [16] proposed a more general pricing scheme of $\phi(y) = f'(\beta y)$ with $\beta > 1$. However, to the best of our knowledge, our work here is the first to prove that such pricing schemes are optimal even if capacity limits are present, provided that the multiplicative factors are properly chosen.

V. Extensions: The General Model

In this section, we extend our previous results to more general settings of online resource allocation with heterogeneous cost functions and multiple time slots.

A. The General Model

We consider the same problem setup as in Section II-A, but make the following generalizations. First, the cost function for each resource type $k \in \mathcal{K}$ is denoted by f_k, which can be different among different resource types. Second, if customer $n \in \mathcal{N}$ chooses bundle $b \in \mathcal{B}$, let $r_k^b(t)$ denote the units of resource type k owned by customer n at time slot t, where $t \in \mathcal{T}_n$ and $\mathcal{T}_n$ is the duration that customer n wants to own the resources in bundle b. Suppose bundle b is denoted by the same vector $(r_1^b, \cdots, r_K^b)$ as before, then $r_k^b(t)$ is given by

$$r_k^b(t) = \begin{cases} r_k^b & \text{if } t \in \mathcal{T}_n, \\ 0 & \text{if } t \in \mathcal{T} \backslash \mathcal{T}_n, \end{cases} \tag{34}$$

where $\mathcal{T}$ denotes the total time horizon of interest. Based on the above generalizations, our extended model can account for *multi-period online resource allocation with heterogeneous cost functions*. In particular, the new offline social welfare maximization problem is given by:

$$\underset{\mathbf{x},\mathbf{y}}{\text{maximize}} \quad \sum_{n\in\mathcal{N}}\sum_{b\in\mathcal{B}} v_n^b x_n^b - \sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} f_k\big(y_k(t)\big) \tag{35a}$$

$$\text{subject to} \quad \sum_{n\in\mathcal{N}}\sum_{b\in\mathcal{B}} r_k^b(t) x_n^b = y_k(t), \forall k, t, \tag{35b}$$

$$\sum_{b\in\mathcal{B}} x_n^b \le 1, \forall n, \tag{35c}$$

$$0 \le y_k(t) \le 1, \forall k, t, \tag{35d}$$

$$x_n^b \in \{0,1\}, \forall n, b, \tag{35e}$$

where $y_k(t)$ is the utilization of resource type k at time t.

B. Generalization of Theorem 1

To generalize Theorem 1 to account for the above resource allocation model, we first need to redefine some key parameters as follows. We assume that $\max_{n\in\mathcal{N},b\in\mathcal{B},r_k^b\neq 0}\left\{\frac{v_n^b}{|\mathcal{T}_n|\cdot r_k^b}\right\} \le \overline{p}_k, \underline{c}_k \triangleq f_k'(0)$, and $\overline{c}_k \triangleq f_k'(1), \forall k \in \mathcal{K}$, where $\overline{p}_k$, $\underline{c}_k$ and $\overline{c}_k$ correspond to $\overline{p}$, $\underline{c}$ and $\overline{c}$ in Section II-B, respectively. Here, we have an upper bound $\overline{p}_k$, a minimum marginal cost $\underline{c}_k$, and a maximum marginal cost $\overline{c}_k$ for each $k \in \mathcal{K}$. In particular, $\overline{p}_k$ can be interpreted as the maximum price customers are willing to pay for purchasing a single unit of resource type k for each time slot.

Below we give a general version of Theorem 1. Specifically, we focus on the case of HUC only (i.e., $\overline{p}_k > \overline{c}_k$). The case of LUC (i.e., $\overline{p}_k \le \overline{c}_k$) is similar and is omitted for brevity.

Theorem 12. *For any $k \in \mathcal{K}$, if $f_k \in \mathcal{F}$ and the upper bound $\overline{p}_k \in (\overline{c}_k, +\infty)$, then we have:*

- Sufficiency. *For any given $\alpha_k \ge 1$, if $\phi_k(y)$ is a solution to the following two first-order BVPs simultaneously:*

$$\begin{cases} \phi_k'(y) = \alpha_k \cdot \frac{\phi_k(y) - f_k'(y)}{f_k'^{-1}(\phi_k(y))}, y \in (0, u_k), \\ \phi_k(0) = \underline{c}_k, \phi_k(u_k) = \overline{c}_k. \end{cases} \tag{36a}$$

$$\begin{cases} \phi_k'(y) = \alpha_k \cdot \left(\phi_k(y) - f_k'(y)\right), y \in (u_k, 1), \\ \phi_k(u_k) = \overline{c}_k, \phi_k(1) \ge \overline{p}_k, \end{cases} \tag{36b}$$

  where $u_k \in (0,1)$ is the dividing threshold of ϕ_k, then PPM$_{\boldsymbol{\phi}}$ is $\max_{k\in\mathcal{K}}\{\alpha_k\}$-competitive.
- Necessity. *If there is an α-competitive online algorithm, then for all $k \in \mathcal{K}$, there must exist a dividing threshold $u_k \in (0,1)$ and a strictly-increasing pricing function $\phi_k(y)$ such that $\phi_k(y)$ satisfies Problem* (36a) *and Problem* (36b) *with a feasible competitive ratio parameter $\alpha_k \in [1, \alpha]$.*

The proof of Theorem 12 is similar to that of Theorem 1, and the details are given in [26]. Based on the two BVPs in Theorem 12, for each resource type $k \in \mathcal{K}$, we can define the minimum competitive ratio parameter α_k^* in a similar way as Proposition 2. The final competitive ratio is then given by $\alpha_*(\mathcal{S}) = \max_{k\in\mathcal{K}}\{\alpha_k^*\}$. We can also define the lower bound of α_k according to Definition 3. The principles in Algorithm 2 can thus be applied for characterizing the competitive ratios and the corresponding pricing functions in the general case. Meanwhile, our analytical results for the setup with power cost functions also hold with some slight modifications. The details are omitted for brevity.

VI. Empirical Evaluation

In this section we evaluate the performance of our designed online mechanism via extensive empirical experiments of online job scheduling in cloud computing.

A. Simulation Setup

(**Supply Costs**) We consider two types of resources ($K = 2$), namely, CPU and RAM. We use the traces of one-month computing tasks in a Google cluster [28]. We assume each bundle $b \in \mathcal{B}$ is given by $(r_{\mathsf{cpu}}^b, r_{\mathsf{ram}}^b)$, where r_{cpu}^b and r_{ram}^b can be any value in $\{0.001, 0.003, 0.005\}$ units of the total normalized capacity 1. Therefore, in total we have $|\mathcal{B}| = 9$ bundles. We assume $T = 3600$ time slots and each time slot is 10 seconds. The cost functions for CPU and RAM are given by $f_{\mathsf{cpu}}(y) = a_{\mathsf{cpu}} y^{s_{\mathsf{cpu}}}$ and $f_{\mathsf{ram}}(y) = a_{\mathsf{ram}} y^{s_{\mathsf{ram}}}$, respectively. Following [19], [20], [21], we assume $s_{\mathsf{cpu}} = 3$ and $s_{\mathsf{ram}} = 1.2$. We set up the coefficients $(a_{\mathsf{cpu}}, a_{\mathsf{ram}}) = (0.223, 8.38 \times 10^{-6})$ by keeping the ratio of $a_{\mathsf{cpu}}/a_{\mathsf{ram}}$ based on [29], where the dominate power consumption is from CPU. This setup of cost functions follows the typical power consumption models of data centers [4]. The minimum marginal costs are zero and

the maximum marginal costs are given by $\overline{c}_{\mathsf{cpu}} \approx 0.67$ and $\overline{c}_{\mathsf{ram}} \approx 1.01 \times 10^{-5}$. Since $\overline{c}_{\mathsf{ram}}$ is much smaller than $\overline{c}_{\mathsf{cpu}}$, our simulation mainly focuses on the power costs of CPU consumptions. For simplicity, we write $\overline{c}_{\mathsf{cpu}} = 0.67$ hereinafter without the approximation sign.

(**Job Arrivals**) We consider the total number of jobs is $N = 4000$. The arrival time and duration of each job follow the job arrival and departure times in Google cluster trace [28]. For job n, the valuation v_n^b is given by $v_n^b = p|\mathcal{T}_n|r_{\mathsf{cpu}}^b$, where $|\mathcal{T}_n|$ denotes the duration of job n and p is a random variable constructed as follows:

1) Uniform-Exact Case (Case-UE). The sequences of p are uniformly distributed within $[0, \overline{p}]$ and the pricing functions are designed based on the exact value of $\overline{p}$.
2) Extreme-Exact Case (Case-EE). This extreme case evaluates the performance robustness of online mechanisms. For the first-half of the total jobs, the sequences of p are uniformly distributed within $[0, \frac{\overline{p}}{2}]$. While for the second-half, the sequences of p are uniformly distributed within $[\frac{\overline{p}}{2}, \overline{p}]$. Meanwhile, the pricing functions are designed based on the exact value of $\overline{p}$.
3) Uniform-Inexact Case (Case-UI). The sequences of p are uniformly distributed within $[0, \overline{p}]$. However, the pricing function is designed based on the estimated upper bound $\overline{p}_{\mathsf{estimate}} = \overline{p}(1+\delta)$, where $\delta \in [-0.8, 2.4]$, meaning that $\overline{p}$ can be underestimated (overestimated) for as much as 80% (240%). We use this case to evaluate the impact of underestimations/overestimations of $\overline{p}$ on the performances of different online mechanisms.
4) Extreme-Inexact Case (Case-EI). This is a mixture of the second and third case. Specifically, the sequences of p are generated in the same way as those in Case-EE, and $\overline{p}_{\mathsf{estimate}}$ follows the same setup as Case-UI.

(**Performance Metrics**) Given any arrival instance $\mathcal{A}$, we define the empirical ratio (ER) by

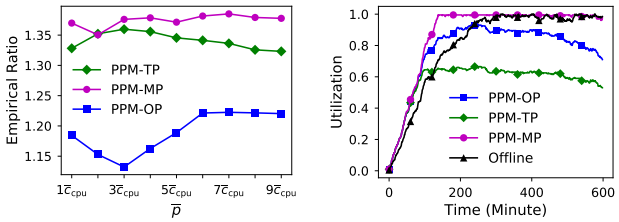
$$\mathrm{ER}(\mathcal{A}) \triangleq \frac{W_{\mathsf{opt}}(\mathcal{A})}{W_{\mathsf{online}}(\mathcal{A})},$$

where $W_{\mathsf{opt}}(\mathcal{A})$ is the optimal objective of Problem (3). For each sample of $\mathcal{A}$, we solve Problem (3) by Gurobi 8.1 via its Python API[2], and then evaluate ERs over 1000 samples of $\mathcal{A}$'s to get the average ER of each online mechanism.

(**Benchmarks**) We refer to our proposed PPM with optimal pricing as PPM-OP, and compare it with the offline benchmark and two existing PPMs as follows:

- PPM with Twice-the-index Pricing (PPM-TP). This PPM is first proposed in [15] and later extended for cloud resource allocation problems in [21]. By PPM-TP, when $y \in [0, 0.5]$, the pricing function is $\phi(y) = f'(2y)$; when $y \in (0.5, 1]$, the pricing function is exponential and the detailed expression is referred to [21].
- PPM with Myopic Pricing (PPM-MP). This PPM prices the resources based on the current marginal costs, i.e., $\phi(y) = f'(y)$, and thus is myopic in the sense that the resources will be allocated aggressively without reservation for potential high-PUV customers in the future.

[2]http://www.gurobi.com/index



(a) Empirical Ratios (b) Total Resource Utilization

Fig. 4. ERs and total resource utilizations of different online mechanisms in Case-UE. Each point in the left figure is an average of 1000 instances. The right figure is for one instance of $\overline{p} = 2\overline{c}_{\mathsf{cpu}} = 1.34$.

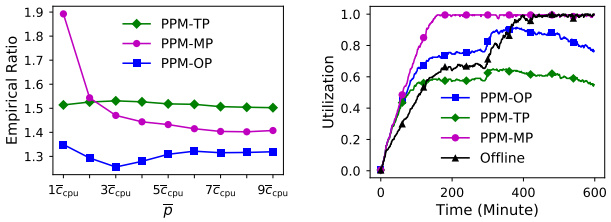
For any given resource utilization level $y \in (0, 1)$, PPM-TP always has the highest posted prices and PPM-MP always has the cheapest ones. Therefore, among the three online mechanisms, PPM-TP (PPM-MP) is the most conservative (aggressive) one[3].

B. Numerical Results

Fig. 4 compares the different online mechanisms in Case-UE. As shown in Fig. 4(a), $\overline{p}$ varies within $[\overline{c}_{\mathsf{cpu}}, 9\overline{c}_{\mathsf{cpu}}]$, where $\overline{c}_{\mathsf{cpu}} = 0.67$ and $9\overline{c}_{\mathsf{cpu}} = 6.03$. Note that based on Eq. (21), we have $C_s \approx 4.21 \approx 6.28\overline{c}_{\mathsf{cpu}}$, and thus the setup of $\overline{p} \in [\overline{c}_{\mathsf{cpu}}, 9\overline{c}_{\mathsf{cpu}}]$ in Fig. 4(a) covers all the cases of LUC, HUC_1, and HUC_2. We can see that the ERs of our proposed PPM-OP are roughly around $1.12 \sim 1.22$, which strictly outperforms both PPM-TP and PPM-MP. An interesting result revealed by Fig. 4(a) is that the ER performance of PPM-OP (PPM-TP) first improves (deteriorates) and then deteriorates (improves) when $\overline{p}$ increases within $[\overline{c}_{\mathsf{cpu}}, 9\overline{c}_{\mathsf{cpu}}]$. We argue that the ER behaviours of PPM-OP for $\overline{p} \in [\overline{c}_{\mathsf{cpu}}, 6\overline{c}_{\mathsf{cpu}}]$ are reasonable although the optimal competitive ratios are the same when $\overline{p} \in [\overline{c}_{\mathsf{cpu}}, 6\overline{c}_{\mathsf{cpu}}] \subset [\overline{c}_{\mathsf{cpu}}, C_s]$. The insight is that when $\overline{p}$ slightly increases from $\overline{c}_{\mathsf{cpu}}$ to $3\overline{c}_{\mathsf{cpu}}$, the uncertainty level of the arrival instances also slightly increases, and this is beneficial for the online posted-price control since whatever decisions made now may have remedies in the future. However, when $\overline{p} > 3\overline{c}_{\mathsf{cpu}}$, the ER performance of PPM-OP becomes worse whenever $\overline{p}$ increases. This is because the uncertainty level of the arrival instances is too high so that it becomes challenging to perform online posted-price control without future information. The differences of the three online mechanisms can also be seen by their total CPU resource utilizations in Fig. 4(b). PPM-MP is the most aggressive and thus the total capacity is quickly depleted (i.e., 100% utilization). PPM-TP is the most conservative and reserves over 40% capacity for future jobs. The total CPU resource utilization of PPM-OP (around 85% maximum utilization) stays between those of PPM-MP and PPM-TP, and achieves a better balance between aggressiveness and conservativeness.

Fig. 5 shows the ERs of online mechanisms in Case-EE. The first result revealed by Fig. 5(a) is intuitive, namely, the ERs of all the three online mechanisms are worse than the ERs

[3]Based on (28), the most conservative optimal pricing function is $\phi_w(y) = sf'(y)$, which is still more aggressive than $f'(2y) = 2^s f'(y)$ when $s > 1$.



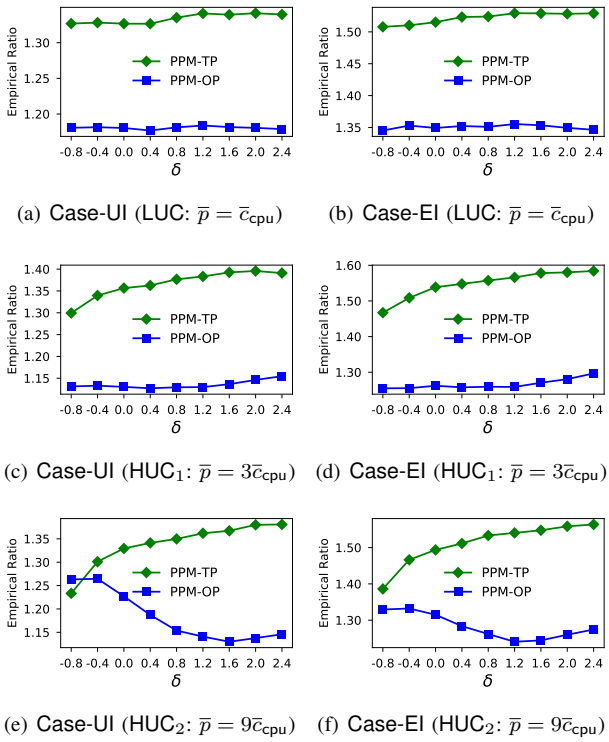
(a) Empirical Ratios (b) Total Resource Utilization

Fig. 5. ERs and total resource utilizations of different online mechanisms in Case-EE. Each point in the left figure is an average of 1000 instances. The right figure is for one instance of $\overline{p} = 2\overline{c}_{\mathsf{cpu}} = 1.34$.

in Case-UE. Second, our proposed PPM-OP achieves a very competitive performance even in this extreme case: the ERs of PPM-OP are always below 1.4, which outperforms PPM-TP by more than 15% in average. Third, Fig. 5(a) also shows that the greedy mechanism PPM-MP is significantly worse than both PPM-TP and PPM-OP when $\overline{p}$ is small, but outperforms PPM-TP when $\overline{p}$ is large. However, due to the greedy nature of PPM-MP, the ERs of PPM-MP are considerably less robust than those of PPM-OP and PPM-TP, as illustrated in Fig. 5(a). Fig. 5(b) shows the total CPU resource utilizations of different mechanisms when $\overline{p} = 2\overline{c}_{\mathsf{cpu}}$. Since in Case-EE the first-half (second-half) of the total jobs have low (high) PUVs, the total CPU resource utilization profile of the offline benchmark depicts two distinct levels within the duration of $t \in [0, 300]$ and $t \in [300, 600]$. We can see that PPM-MP completely fails to achieve such a two-level utilization profile by quickly reaching the capacity limit before $t = 200$ min; PPM-TP performs better than PPM-MP, but reserves too much available capacity for future jobs (too conservative). In comparison, PPM-OP shows the capability of distinguishing the two different intervals, and has a similar utilization profile to that of the offline benchmark.

We next demonstrate the impact of inexact estimations of $\overline{p}$ on the ER performances of PPM-OP and PPM-TP (note that the performance of PPM-MP is independent of $\overline{p}$). We perform an indepth comparison between PPM-OP and PPM-TP in both Case-UI and Case-EI with $\overline{p} = \overline{c}_{\mathsf{cpu}} \in (0, \overline{c}_{\mathsf{cpu}}]$ (i.e., LUC), $\overline{p} = 3\overline{c}_{\mathsf{cpu}} \in (\overline{c}_{\mathsf{cpu}}, C_s]$ (i.e., HUC_1), and $\overline{p} = 9\overline{c}_{\mathsf{cpu}} \in (C_s, +\infty)$ (i.e., HUC_2), where $C_s \approx 6.28\overline{c}_{\mathsf{cpu}}$. Hence, we have six cases in total, which correspond to the six sub-figures in Fig. 6. We note that the choices of $\overline{p} = 3\overline{c}_{\mathsf{cpu}}$ and $\overline{p} = 9\overline{c}_{\mathsf{cpu}}$ have no specific reasons other than making them in HUC_1 and HUC_2, respectively.

- Fig. 6(a) and Fig. 6(b) show that the ER performances of both PPM-OP and PPM-TP are insensitive to δ in LUC. The insensitivity of PPM-TP is reasonable since the first segment of the pricing function of PPM-TP, i.e., $\phi(y) = f'(2y)$, is independent of $\overline{p}$. Therefore, when $\overline{p} = \overline{c}_{\mathsf{cpu}}$, the highest resource utilization level will not significantly exceed 50% of the total capacity (since $\overline{p} \le \overline{c}_{\mathsf{cpu}} = f'(2 * 0.5)$). As a result, the first segment of the pricing function of PPM-TP is the major active part for most of the time slots. Meanwhile, it is also not surprising that PPM-OP is insensitive to δ in LUC since $\overline{p}_{\mathsf{estimation}}$ does not influence PPM-OP when $\overline{p}_{\mathsf{estimation}} \le C_s \approx 6.28\overline{c}_{\mathsf{cpu}}$.
- Fig. 6(c) and Fig. 6(d) show that the ER performance of PPM-TP always deteriorates with the increase of δ in HUC_1 (underestimation is always better than overestimation). The ER behaviors of PPM-TP are interesting but quite reasonable since an overestimation of $\overline{p}$ will make the second segment of the pricing function of PPM-TP over conservative, leading to a worse ER performance. Similar results have also been reported by [21]. Unlike PPM-TP, PPM-OP is insensitive to the estimation error δ when $\delta < C_s/\overline{p} - 1 \approx 1.1$, meaning that as long as the overestimation of $\overline{p}$ does not change the design of optimal pricing functions from HUC_1 to HUC_2, the ERs of PPM-OP will be the same. However, a larger estimation error $\delta > 1.1$ will slightly worsen the ER performance of PPM-OP as the optimal pricing function in HUC_2 is too conservative in HUC_1.
- Fig. 6(e) and Fig. 6(f) show that the ER performances of PPM-TP and PPM-OP have opposite behaviors w.r.t. the estimation error δ in HUC_2. Specifically, overestimations of $\overline{p}$ still increase the ERs of PPM-TP, similar to the results in HUC_1. In contrast, PPM-OP will benefit from overestimating $\overline{p}$ when δ is within a certain range (e.g., when $\delta \in (0, 1.6)$ in Fig. 6(e)), and then deteriorate when the estimation error δ is too large (e.g., when $\delta > 1.6$ in Fig. 6(e)). Note that the ER behaviors of PPM-OP are very counter-intuitive since an overestimation of $\overline{p}$ in HUC_2 will inevitably make the optimal pricing functions in PPM-OP more conservative, which intuitively should

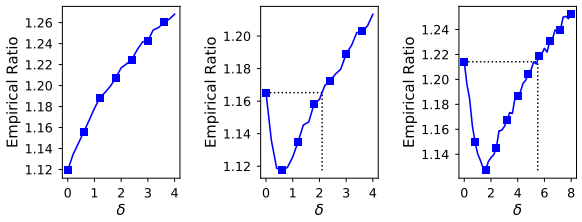


(a) Case-UI (LUC: $\overline{p} = \overline{c}_{\mathsf{cpu}}$) (b) Case-EI (LUC: $\overline{p} = \overline{c}_{\mathsf{cpu}}$)

(c) Case-UI (HUC_1: $\overline{p} = 3\overline{c}_{\mathsf{cpu}}$) (d) Case-EI ($\mathsf{HUC}_1$: $\overline{p} = 3\overline{c}_{\mathsf{cpu}}$)

(e) Case-UI (HUC_2: $\overline{p} = 9\overline{c}_{\mathsf{cpu}}$) (f) Case-EI ($\mathsf{HUC}_2$: $\overline{p} = 9\overline{c}_{\mathsf{cpu}}$)

Fig. 6. Comparison between PPM-OP and PPM-TP when the estimated upper bound $\overline{p}_{\mathsf{estimate}}$ is inexact, where $\overline{p}_{\mathsf{estimate}} = \overline{p}(1+\delta)$ and $\overline{p}$ denotes the real upper bound. Each point in the figure is an average of 1000 instances.



(a) $\mathbb{N}(\overline{c}_{\mathsf{cpu}}, 1, 0, \overline{p})$ (b) $\mathbb{N}(\overline{c}_{\mathsf{cpu}}, 4, 0, \overline{p})$ (c) $\mathbb{N}(\overline{c}_{\mathsf{cpu}}, 100, 0, \overline{p})$

Fig. 7. Impact of overestimations of $\overline{p}$ on the ER performance of PPM-OP. Each point in the figure is an average of 1000 instances.

lead to a worse ER performance. However, Fig. 6(e) and Fig. 6(f) show that, the ER performance of PPM-OP will deteriorate only if the overestimation of $\overline{p}$ exceeds some threshold (e.g., 1.6 in Fig. 6(e) and 1.2 in Fig. 6(f)).

The above illustrations indicate that underestimations of $\overline{p}$ should always be avoided when using our proposed PPM-OP. This is because a negative δ either has no impact on the ER performance of PPM-OP in LUC and HUC_1 (the first four sub-figures in Fig. 6), or makes it even worse in HUC_2 (the final two sub-figures in Fig. 6). Meanwhile, it is generally beneficial to slightly overestimate $\overline{p}$ when $\overline{p}$ is larger than C_s.

To further evaluate the impact of overestimations of $\overline{p}$ on the ER performance of PPM-OP, in particular, to quantify how much overestimation will lead to a worse ER performance than using the exact value of $\overline{p}$, we change the uniform distribution of p in Case-UI to a truncated normal distribution as follows:

$$p \sim \mathbb{N}(\mu, \sigma^2, 0, \overline{p}),$$

where $\mu, \sigma, 0$, and $\overline{p}$ denote the mean, the standard deviation, the lower bound, and the upper bound of random variable p, respectively. We set $\mu = \overline{c}_{\mathsf{cpu}}$ and $\overline{p} = 9\overline{c}_{\mathsf{cpu}}$, and assume similarly as Case-UI that the optimal pricing function is designed based on the estimated upper bound $\overline{p}_{\mathsf{estimate}} = \overline{p}(1+\delta)$, where $\sigma > 0$ since here we only consider overestimation. We plot the ER performances of PPM-OP with different variances in Fig. 7. It can be seen that when the variance is small, e.g., $\sigma = 1$ in Fig. 7(a), the ER performance of PPM-OP becomes worse w.r.t. the increase of $\delta > 0$. When the variance is higher, e.g., $\sigma = 2$ in Fig. 7(b) and $\sigma = 10$ in Fig. 7(c), the ER performance of PPM-OP first improves and then deteriorates w.r.t. the increase of $\delta > 0$, similar to the results in Fig. 6 when p is uniformly distributed. An interesting result revealed by Fig. 7 is that PPM-OP can tolerate a higher estimation error of $\overline{p}$ when the variance of p is higher. In other words, when the arrival instance is highly uncertain or volatile, it tends to be more beneficial for the provider to overestimate $\overline{p}$. This insight shows that when there exists no exact statistical model about future arrivals, the information uncertainty is not always a disadvantage. Instead, the provider can artificially amplify the estimation of $\overline{p}$ so as to benefit from the uncertainty of arrival instances. We argue that this is another advantage of our proposed PPM-OP as the prior theoretic analysis does not provide such a guarantee.

VII. Conclusion

We studied the online combinatorial auctions for resource allocation with supply costs and capacity limits. In the studied model, the provider charges payment from customers who purchase a bundle of resources and incurs an increasing supply cost with respect to the total resource allocated. We focused on maximizing the social welfare. Adopting the competitive analysis framework we provided an optimal online mechanism via posted-price. Our online mechanism is optimal in that no other online algorithms can achieve a better competitive ratio. Our theoretic results improve and generalize the results in prior work. Moreover, we validated our results via empirical studies of online resource allocation in cloud computing, and showed that our pricing mechanism is more competitive than existing benchmarks. We expect that the model and algorithms presented in this paper will find application in different paradigms of networking and computing systems. Meanwhile, leveraging techniques in artificial intelligence and machine learning to extend our model is an interesting future direction, e.g., posted-price via online learning.

References

[1] S. de Vries and R. Vohra, "Combinatorial auctions: A survey," *INFORMS J. Comput.*, vol. 15, no. 3, pp. 284–309, 2003.

[2] N. Nisan and A. Ronen, "Algorithmic mechanism design," *Games and Economic Behavior*, vol. 35, no. 1, pp. 166 – 196, 2001.

[3] D. Porter, S. Rassenti, A. Roopnarine, and V. Smith, "Combinatorial auction design," *Proceedings of the National Academy of Sciences*, vol. 100, no. 19, pp. 11 153–11 157, 2003.

[4] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, Firstquarter 2016.

[5] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker, "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 72–79, May 2017.

[6] Y. Bartal, R. Gonen, and N. Nisan, "Incentive compatible multi unit combinatorial auctions," in *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, ser. TARK '03. New York, NY, USA: ACM, 2003, pp. 72–87.

[7] N. Buchbinder and R. Gonen, "Incentive Compatible Mulit-Unit Combinatorial Auctions: A Primal Dual Approach," *Algorithmica*, vol. 72, pp. 167–190, 2015.

[8] B. Kalyanasundaram and K. R. Pruhs, "An optimal deterministic algorithm for online b-matching," *Theor. Comput. Sci.*, vol. 233, no. 1-2, pp. 319–325, Feb. 2000.

[9] N. R. Devanur and K. Jain, "Online matching with concave returns," in *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing (STOC)*, New York, NY, USA, 2012.

[10] N. R. Devanur and T. P. Hayes, "The adwords problem: Online keyword matching with budgeted bidders under random permutations," in *Proc. of the 10th ACM Conference on Electronic Commerce*. New York, NY, USA: ACM, 2009, pp. 71–78.

[11] A. Mehta, "Online matching and ad allocation," *Found. Trends Theor. Comput. Sci.*, vol. 8, no. 4, pp. 265–368, Oct. 2013.

[12] Y. Azar, N. Buchbinder, T. H. Chan, S. Chen, I. R. Cohen, A. Gupta, Z. Huang, N. Kang, V. Nagarajan, J. Naor, and D. Panigrahi, "Online algorithms for covering and packing problems with convex objectives," in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, vol. 00, Oct. 2016, pp. 148–157.

[13] N. Buchbinder and J. S. Naor, "Online Primal-Dual Algorithms for Covering and Packing," *Mathematics of Operations Research*, vol. 34, no. 2, pp. 270–286, May 2009.

[14] Y. Zhou, D. Chakrabarty, and R. Lukose, "Budget constrained bidding in keyword auctions and online knapsack problems," in *Internet and Network Economics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 566–576.

[15] A. Blum, A. Gupta, Y. Mansour, and A. Sharma, "Welfare and profit maximization with production costs," in *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, 2011, pp. 77–86.

[16] Z. Huang and A. Kim, "Welfare maximization with production costs: A primal dual approach," *Games and Econ. Behav.*, 2018.

[17] L. Perko, *Differential Equations and Dynamical Systems*. New York, NY: Springer New York, 2001.

[18] V. Arnold, *Ordinary differential equations*. MIT Press, 1973.

[19] A. Wierman, L. L. H. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems," in *IEEE INFOCOM 2009*, April 2009, pp. 2007–2015.

[20] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced cpu energy," in *Proceedings of IEEE 36th Annual Foundations of Computer Science*, Oct 1995, pp. 374–382.

[21] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. C. M. Lau, "Online auctions in iaas clouds: Welfare and profit maximization with server costs," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 1034–1047, April 2017.

[22] B. Sun, X. Tan, and D.H.K. Tsang, "Eliciting multi-dimensional flexibilities from electric vehicles: a mechanism design approach," *IEEE Transactions on Power Systems*, vol. 34, no. 5, p. 4038, 2019.

[23] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec 2016.

[24] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[25] S. Chawla, J. D. Hartline, D. L. Malec, and B. Sivan, "Multi-parameter mechanism design and sequential posted pricing," in *Proceedings of the Forty-second ACM Symposium on Theory of Computing*. New York, NY, USA: ACM, 2010, pp. 311–320.

[26] X. Tan *et al.*, "Online combinatorial auctions for resource allocation with supply costs and capacity limits," Tech. Rep. [Online]. Available: https://bit.ly/2O09z29

[27] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*. Oxford University Press, 1995.

[28] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proceedings of the 3rd ACM Symposium on Cloud Computing (SOCC)*. New York, NY, USA: ACM, 2012.

[29] C. K. D. Economou, S. Rivoire and P. Ranganathan., "Full-system power analysis and modeling for server environments," in *Workshop on Modeling Benchmarking and Simulation (MOBS)*, 2006.