Core-Selecting Auctions for Dynamically Allocating Heterogeneous VMs in Cloud Computing

Haoming Fu, Zongpeng Li
Department of Computer Science
University of Calgary
{hafu, zongpeng}@ucalgary.ca

Chuan Wu
Department of Computer Science
The University of Hong Kong
cwu@cs.hku.hk

Xiaowen Chu
Department of Computer Science
Hong Kong Baptist University
chxw@comp.hkbu.edu.hk

***Abstract*— In a cloud market, the cloud provider provisions heterogeneous virtual machine (VM) instances from its resource pool, for allocation to cloud users. Auction-based allocations are efficient in assigning VMs to users who value them the most. Existing auction design often overlooks the heterogeneity of VMs, and does not consider dynamic, demand-driven VM provisioning. Moreover, the classic VCG auction leads to unsatisfactory seller revenues and vulnerability to a strategic bidding behavior known as *shill bidding*. This work presents a new type of core-selecting VM auctions, which are combinatorial auctions that always select bidder charges from the core of the price vector space, with guaranteed economic efficiency under truthful bidding. These auctions represent a comprehensive three-phase mechanism that instructs the cloud provider to judiciously assemble, allocate, and price VM bundles. They are proof against shills, can improve seller revenue over existing auction mechanisms, and can be tailored to maximize truthfulness.**

I. Introduction

Cloud computing is emerging as a new computing paradigm for flexibly organizing a shared pool of configurable resources (CPU, RAM, storage, *etc.*) in data centers into various types of virtual machines (VMs), for allocation to cloud users anywhere, anytime [1]. Leveraging resource virtualization, such a paradigm abstracts the underlying physical resources from the users by directly providing them with the view of VMs. As a market-driven pricing approach that provides economic incentives for both the *cloud provider* (CP) and *cloud users* (CUs), *auctions* represent a fast and efficient mechanism for VM allocation. For a well known example, Amazon has made its initial effort to implement an auction-based VM allocation mechanism termed *Spot Instances* [2].

Existing literature on VM auctions often treats VMs as identical or substitutable goods [3], [4]. However, CUs have natural demands for combinations of *heterogenous* VMs in practice, given the inherent heterogeneity in real-world computing tasks. For instance, a social gaming application often consists of a front-end web server layer, a load balancing layer and a back-end data storage layer, each best served by a VM that is intended for communication-intensive, computation-intensive, and storage-intensive tasks, respectively [5]. Table I shows an excerpt from the Amazon EC2 Spot Instances. *Multi-unit combinatorial auctions* are natural for such a cloud market, enabling expressive bids for requesting bundles of VM instances belonging to different types.

This work was supported by National Sciences and Engineering Research Council of Canada (NSERC) (10006298 and 10006977) and the University Grants Committee of Hong Kong (HKU 718513 and HKBU 210412).

TABLE I
VM Configurations from Spot Instances [2].

ID	VM Type	CPU	ECU*	Memory	Storage
1	m1.medium	1	2	3.75 GB	410 GB
2	m1.large	2	4	7.5 GB	840 GB
3	c1.xlarge	8	20	7 GB	1680 GB
4	cc2.8xlarge	32	88	60.5 GB	3360 GB
5	m2.xlarge	2	6.5	17.1 GB	420 GB
6	m2.2xlarge	4	13	34.2 GB	850 GB
7	hi1.4xlarge	16	35	60.5 GB	2048 GB

*ECU: EC2 compute units

Prior to the phase of allocating VMs to CUs, the CP needs to assemble the configurable resources from its resource pool into VM instances [1]. The *de facto* standard in the literature of VM auctions is to ignore the phase of resource provisioning, assuming implicitly the strategy of *static resource provisioning* (SRP) [6] — *i.e.*, VMs are pre-configured and already assembled when CUs' bids are received. In contrast, *dynamic resource provisioning* (DRP) is considered in this work. DRP enables the flexibility of tailoring the VM construction process to the specific bids received, improving hardware utilization and overall revenue gleaned. DRP is practically feasible in that online VMs deployment incurs only a small time overhead [7].

The celebrated VCG mechanism [8]–[10] represents essentially the *only* auction mechanism that is both *truthful* and *efficient* [11]. In the context of the cloud market, an auction is truthful if a CU has no incentives to lie about its valuation of its desired VM bundle; and (economic) efficiency holds when winners are chosen to maximize the aggregated valuation of awarded VM bundles. Despite a myriad of interests in theoretical research, the VCG mechanism suffers from two severe economic problems that have essentially prevented its direct application in practice. First, a VCG auction generates a low revenue for the auctioneer, under-exploiting the payment potential of CUs. Second, a VCG auction is susceptible to *shill bidding*, or false-name bidding, in which a single CU impersonates multiple CUs, each bidding for a subset of the desired VM instances [12].

Since the VCG mechanism represents the only truthful and efficient auction mechanism, a relaxation of either efficiency or truthfulness is inevitable in any auction design that aims to address its two inherent problems. Previous research often concentrates on truthfulness by relaxing efficiency, because social welfare maximization leads to NP-hard problems often in the form of 0-1 integer programs [6]. However, latest empirical evidences suggest that 0-1 linear integer programs resulting from such real-world problems can be solved in a reasonable amount of time (*e.g.*, 5000 integer variables in less than 1 second using CPLEX [13] on a laptop computer). A sacrifice of efficiency is therefore less justified. Core-selecting auctions, recently proposed in economics [14]–[16], provide a promising direction for designing combinatorial auctions that are efficient, shill-proof, and generate satisfactory revenues. An auction outcome is *in-core* if no other outcome both is preferred by some subset of CUs and increases the CP's revenue at the same time. The in-core property implies efficiency (social welfare maximization), which in turn ensures effective utilization of cloud resources.

This work makes the following three main contributions. *First*, we are the first to apply the core-selecting auction framework to a cloud market, by formulating a winner determination problem, and a pair of correlated *linear program* (LP) and *quadratic program* (QP) for payment computation. While guaranteeing efficiency, the in-core property is further proven to be sufficient and necessary to avoid shill bidding. Furthermore, our core-selecting auctions are able to achieve a revenue that is at least on par with that of VCG mechanisms, and can be tailored to achieve provable minimization of CUs' incentive to deviate from truthful bidding. Our proposed auctions further represent the first design of multi-unit core-selecting combinatorial auctions.

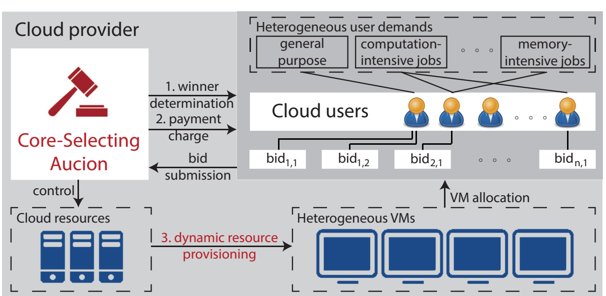


Fig. 1. An illustration of the three-dimension core-selecting auction design.

Second, we propose a new, three-dimension auction framework (Fig. 1) for modeling dynamic resource provisioning. Existing cloud auctions are mostly two dimensional, including winner determination (VM bundle allocation) and payment computation, with static resource provisioning. Our empirical studies reveal that dynamic provisioning can enable substantial improvements over static provisioning in terms of social welfare, seller revenue, and resource utilization.

Third, departing from the previous standard of simplifying VMs into homogeneous and substitutable commodities, our core-selecting auctions give up the restrictive assumption that all VMs are equally powerful, or equally powerful up to an integer scaling factor [6]. Our auctions are expressive enough to model (a) the provisioning of heterogeneous types of VMs for sale at the CP side, and (b) the desire of VMs and VM bundles of different nature by real-world computing jobs, and hence are more readily applicable in realistic cloud systems.

We conducted extensive simulations to examine the performance of the proposed core-selecting VM auctions, driven partially by real world traces from Google Cluster Data [17]. Our auctions run efficiently on a platform with limited computing resources, and generate higher revenues than the VCG mechanism does. DRP is observed to enable a stably high resource utilization and outperforms SRP in terms of revenue by a ratio of at least 20%, because judiciously provisioning resources can accommodate more requests from CUs.

In the rest of the paper, we discuss related work in Sec. II, and present preliminaries in Sec. III. Sec. IV introduces and analyzes core-selecting auctions for a cloud market, while Sec. V studies payment rules. Simulation studies are presented in Sec. VI. Sec. VII concludes the paper.

II. Related Work

The classic VCG mechanism [8]–[10] is both efficient and truthful, but suffers from low seller revenue and vulnerability to shill bidding [12]. Many existing auctions for VM allocation similarly choose to enforce absolute truthfulness and have to settle for a low revenue. Zhang *et al.* [3] proposed an online auction for truthful VM allocation to serve heterogeneous user demands, yet it achieves only 50% of the VCG revenue. Auctions belonging to this realm also appeared in markets formed by federations of clouds, for which two truthful double-auctions were specifically designed [18], [19]. Zhang *et al.* [20] designed a randomized combinatorial auction by utilizing a pair of primal and dual linear programs, and pursue absolute truthfulness at the price of computational and economic efficiencies.

Using auction mechanisms to manage VM allocation has been extensively studied in the past, but many of them simplified VMs into a homogeneous commodity [3], [21]. With an intention to address this problem, Zaman and Grou proposed CA-GREEDY [4] and CA-LP [6], which are combinatorial multi-unit auction mechanisms for VM allocation. However, they still assumed that different types of VMs are interchangeable up to a multiplicative factor, which does not faithfully reflect real-world VM heterogeneity.

Bringing the heterogeneity nature of VMs into consideration, Li *et al.* proposed double auctions for heterogeneous virtual machine trading [18], [19]. Wang *et al.* formulated a combinatorial auction for VM pricing, which is coupled with an algorithm that is computationally efficient [22]. These auctions ingored the stage of resource provisioning, and implicitly relied on static provisioning that is relatively inefficient [1].

III. Problem Formulation and Preliminaries

We consider a *cloud provider* (CP) who periodically pools its idle resources and leases them as *virtual machines* (VMs) to *cloud users* (CUs), through round-by-round auctions. There are t types of resources, such as CPU, RAM and network bandwidth, in the CP's resource pool. The total amount of type k resource is π_k, where $1 \le k \le t$. The CP offers m different types of virtual machines $\mathrm{VM}_1, \ldots, \mathrm{VM}_m$. Based on CUs' demands submitted in their bids, the CP assembles its resources into an appropriate number of VMs for each type j. A VM_j instance consumes an amount α_j^k of type k resource.

A set $\mathcal{N}$ of CUs act as bidders in the auction. Each CU $i \in \mathcal{N}$ is free to bid for one or more bundles of VM instances. For each bundle $\mathcal{S} = (r_1, r_2, \ldots, r_m)$, where r_j is the number of VM_j instances CU i requests, let $v_i(\mathcal{S})$ be i's valuation of $\mathcal{S}$. Let $\mathcal{B}_i$ be the set of all VM bundles CU i bids for. We adopt an *XOR bidding language*, in which a CU can submit multiple bids, but can win a single bid only (a CU's bids are mutually *exclusive*).

Each CU i has a *quasi-linear utility* defined as:

$$u_i = \begin{cases} v_i(\mathcal{S}) - p_i & \text{if CU } i \text{ wins a bundle } \mathcal{S} \in \mathcal{B}_i \\ 0 & \text{otherwise} \end{cases}$$

where p_i is the payment of CU i if it wins $\mathcal{S}$. We assume that the CUs are *individually rational* in that they always prefer a higher utility; consequently $v_i(\mathcal{S})$ is the maximum amount that CU i is willing to pay for $\mathcal{S}$.

Let $b_i(\mathcal{S})$ denote the bid submitted by CU i for VM bundle $\mathcal{S}$. We denote by o the auctioneer (CP) and by $u_o = \sum_{i \in \mathcal{N}} p_i$ the revenue of the auctioneer.

After collecting all bids submitted, the CP computes a resource provision scheme, a VM instance allocation plan as well as a corresponding payment vector. The winner determination problem can be formulated accordingly (WDP):

$$w(\mathcal{N}) = \max \sum_{i \in \mathcal{N}} \sum_{\mathcal{S} \in \mathcal{B}_i} b_i(\mathcal{S}) x_i(\mathcal{S}) \tag{1}$$

subject to:

$$\sum_{j=1}^{m} n_j \alpha_j^k \le \pi_k \qquad \forall 1 \le k \le t; \tag{2}$$

$$\sum_{\mathcal{S} \in \mathcal{B}_i} x_i(\mathcal{S}) \le 1 \qquad \forall i \in \mathcal{N}; \tag{3}$$

$$\sum_{i \in \mathcal{N}} \sum_{\mathcal{S} \in \mathcal{B}_i} x_i(\mathcal{S}) r_j \le n_j \qquad \forall 1 \le j \le m, \quad \mathcal{S} = (r_1, r_2, \ldots, r_m); \tag{4}$$

$$n_j \in \mathbb{N} \qquad \forall 1 \le j \le m; \tag{5}$$

$$x_i(\mathcal{S}) \in \{0, 1\} \qquad \forall i \in \mathcal{N},\ \forall \mathcal{S} \in \mathcal{B}_i. \tag{6}$$

Here n_j is the total number of VM_j instances assembled, and let $\mathbf{n}$ be the vector of n_j values. Constraint (2) states that the amount of resource of each type used for provisioning can not exceed what is available in the resource pool. The total number of VM instances of each type won by all CUs is bounded by the number provisioned, as enforced in Constraint (4). Constraint (3) implements the XOR bidding rule.

Theorem 1. *Relaxing **n** to take fractional values in the WDP does not change the value of* $w(\mathcal{N})$.

Proof: Let WDP$'$ be the problem derived from WDP with constraint (5) removed, and let $w'(\mathcal{N})$ be the objective function of WDP$'$. $w'(\mathcal{N}) \ge w(\mathcal{N})$ since WDP$'$ has fewer constraints. Given a solution $(\mathbf{n}', \mathbf{x}')$ to WDP$'$, we claim that $(\lfloor \mathbf{n}' \rfloor, \mathbf{x}')$ is a solution to WDP, in which case we can conclude that $w'(\mathcal{N}) \le w(\mathcal{N})$. In WDP, $(\lfloor \mathbf{n}' \rfloor, \mathbf{x}')$ satisfies constraints (3, 5, 6). $\sum_{j=1}^{m} \lfloor n_j \rfloor \alpha_j^k \le \sum_{j=1}^{m} n_j \alpha_j^k \le \pi_k$, hence constraint (2) holds. Finally, $\sum_{i \in \mathcal{N}} \sum_{\mathcal{S} \in \mathcal{B}_i} x_i(\mathcal{S}) r_j \le \lfloor n_j \rfloor$ (constraint (4) is feasible) since $\sum_{i \in \mathcal{N}} \sum_{\mathcal{S} \in \mathcal{B}_i} x_i(\mathcal{S}) r_j \le n_j$ and $\mathbf{x}, \mathbf{r}$ are integers. Hence the claim is true, and we have $w'(\mathcal{N}) = w(\mathcal{N})$. ∎

In light of Theorem 1, we can solve WDP in three steps: relax WDP to WDP$'$, solve WDP$'$, then round each n_j to $\lfloor n_j \rfloor$. Note that WDP$'$ is a linear integer program with a moderate number of $\{0,1\}$-variables, which are known to be practically solvable in a reasonable amount of time. Our empirical experiences show that even with up to $5,000$ $\{0,1\}$-variables ($x_i(\mathcal{S})$), the WDP can still be solved in a second, using CPLEX [13] on an off-the-shelf laptop computer. An optimal solution to the WDP maximizes the social welfare of the cloud market, and is required in any economically efficient auction. Notations are summarized in the table below.

$\mathcal{N}$	Set of CUs	p_i	Payment of CU i
$\mathcal{W}$	Set of Winners	u_i	Utility of CU i
α_j^k	Amount of type k resource required by a VM_j instance		
$b_i(\mathcal{S})$	Bid price of CU i for bundle $\mathcal{S}$		
$\mathcal{B}_i$	Set of bundles of VMs bid by CU i		
n_j	Number of VM_j instances assembled		
π_k	Total amount of type k resource in resource pool		
$\mathcal{S}_i$	Bundle of VMs allocated to CU i		
$v_i(\mathcal{S})$	Valuation for bundle $\mathcal{S}$ of CU i		
$x_i(\mathcal{S})$	Indicate whether CU i wins bundle $\mathcal{S}$ or not		

IV. Core-Selecting VM Auctions

A. The Core of Our VM Auction

Let $\mathcal{S}_i$ denote the bundle of VMs allocated to CU i in the auction, and $\mathcal{S}_i = \emptyset$ if i loses. An auction outcome is *blocked* by coalition $\mathcal{C} \subseteq \mathcal{N}$ if there is an alternative outcome with awarded bundles $\{\mathcal{S}_i'\}_{i \in \mathcal{N}}$ and payment vector $\mathbf{p}'$, such that $u_i' \ge u_i$ for all $i \in \mathcal{C}$, and $u_o' = \sum_{i \in \mathcal{N}} p_i' > u_o$. $\mathcal{C}$ is referred to as a *blocking coalition*. The outcomes not blocked by any coalition with respect to the submitted bids $\mathbf{b}$ form the *core*:

$$Core(\mathcal{N}) = \{\mathbf{u} \ge 0 | \sum_{i \in \mathcal{N} \cup \{o\}} u_i = w(\mathcal{N}), \sum_{i \in \mathcal{C} \cup \{o\}} u_i \ge w(\mathcal{C}), \forall \mathcal{C} \subseteq \mathcal{N}\} \tag{7}$$

For example, consider seven CUs numbered 1 through 7, each submitting a single bid for three types of VMs, with the following configuration in CPU units and storage (GB): (1, 1), (1, 3) and (2, 1). The CP has 25 CPUs and 25 GB storage in its resource pool, and the following bids are submitted. The CPU and storage consumptions of each bid are given in the superscript and subscript, respectively.

$$b_1(6,0,1)_7^8 = 4 \qquad b_2(2,3,0)_{11}^5 = 5 \qquad b_3(0,0,6)_6^{12} = 4$$
$$b_4(7,0,0)_7^7 = 27 \qquad b_5(0,4,0)_{12}^4 = 25 \qquad b_6(0,0,6)_6^{12} = 24$$
$$b_7(5,3,7)_{22}^{24} = 33$$

The unique set of winners in any efficient allocation includes CUs 4, 5 and 6, generating a social welfare of 76. The core can be drawn in the payment space, shown in Fig. 2.

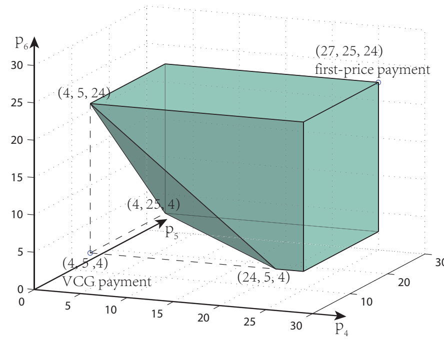


Fig. 2. A geometric illustration of the core.

In this simple example, the constraints defining the core are simply the bids of the losing CUs together with individual rationality of winning CUs. In particular, since CU 1 will always block if CU 4 pays less than 4, we have the constraint $p_4 \geq 4$. Similarly, CU 2 and CU 3 dictate $p_5 \geq 5$ and $p_6 \geq 4$, respectively. CU 7 blocks if CUs 4, 5 and 6 pay less than 33 in total, implying $p_4 + p_5 + p_6 \geq 33$. Upper-bounds are given by winners' bids themselves, consistent with individual rationality. As a result, the intersection of the half spaces defined above formulates the core.

In a general economic problem, the core does not always exist [16]. However, the core is always non-empty in our setting, and in particular contains the first price payment (the winner pays what he bids).

Theorem 2. *The payment vector of first price auction is always in the core of our VM auction.*

Proof: Let $\mathcal{W} \subseteq \mathcal{N}$ be the set of winners. Note that $p_i^{\text{FP}} = b_i(\mathcal{S}_i),\ \forall i \in \mathcal{W}$ and $u_i^{\text{FP}} = 0$ no matter whether CU i wins or not. We have the following:
(1) $\sum_{i \in \mathcal{N} \cup \{o\}} u_i^{\text{FP}} = u_o^{\text{FP}} = \sum_{i \in \mathcal{W}} p_i^{\text{FP}} = \sum_{i \in \mathcal{W}} b_i(\mathcal{S}_i) = w(\mathcal{N})$.
(2) $\sum_{i \in \mathcal{C} \cup \{o\}} u_i^{\text{FP}} = u_o^{\text{FP}} = w(\mathcal{N}) \geq w(\mathcal{C}),\ \forall \mathcal{C} \subseteq \mathcal{N}$.
Hence by definition $\mathbf{u}^{\text{FP}} \in Core(\mathcal{N})$. ∎

B. Necessity of Core-Selecting Auctions

How do we justify the use of a core-selecting auction in a cloud market? We next take VCG revenue as a benchmark, and prove that a core-selecting auction always generates a revenue for the CP at least on par with the VCG revenue, even when bidders are using shills. This is done by showing that a core-selecting auction always leads to a total CU utility no higher than that in the VCG auction, which is combined with the fact that the total utility from both the CP and the CUs is constant in an efficient VM provisioning and allocation scheme.

Informally, the VCG mechanism first solves the WDP to obtain an optimal allocation, and asks each winning CU to pay a price equal to the externality it exerts on the other CUs. More specifically, the VCG payment of a winning CU i is:

$$p_i = b_i(\mathcal{S}_i) - (w(\mathcal{N}) - w(\mathcal{N} \backslash \{i\})) \tag{8}$$

where $w(\mathcal{N}) - w(\mathcal{N} \backslash \{i\})$ is the *marginal contribution* of CU i, *i.e.*, the revenue difference with and without CU i bidding.

We now illustrate the shill bidding problem under the VCG mechanism with the example from section IV-A. Assume that instead of submitting its own bid, CU 5 impersonates four different CUs, each submitting the bid $b(0,1,0)^1_3 = 6.25$. Knowing the rule of the VCG mechanism, CU 5 still wins 4 instances of VM_2, while successfully reducing its payment from 5 to 0 (because each shill wins one VM_2 instance and pays 0 under the VCG mechanism) via disaggregation and impersonation, manifesting the *shill bidding problem* that threatens the CP's revenue.

Now we are ready to show that in our cloud market, core-selecting auctions formulated with WDP as the winner determination problem are essentially robust against shill bidding.

Theorem 3. *In a VM auction formulated with WDP, no CU can earn more than its VCG utility by bidding with shills if and only if the auction is core-selecting.*

Proof: Given a set of CUs $\mathcal{N}$ and an auctioneer o, we show that for any coalition $\mathcal{C} \subseteq \mathcal{N}$, their total utility in a core-selecting auction is no more than if they were to act as a single CU in a VCG auction, in which case the merged entity would bid the same bundle of VMs at the aggregated price. The merged entity of $\mathcal{C}$ enjoys a utility equal to its marginal contribution: $w(\mathcal{N}) - w(\mathcal{N} \backslash \mathcal{C})$. Our restriction is therefore

$$\sum_{i \in \mathcal{C}} u_i \leq w(\mathcal{N}) - w(\mathcal{N} \backslash \mathcal{C}) \tag{9}$$

Since WDP guarantees efficiency, we have

$$w(\mathcal{N}) = \sum_{i \in \mathcal{N} \cup \{o\}} u_i \tag{10}$$

In view of (10), (9) holds if and only if

$$\sum_{i \in (\mathcal{N} \cup \{o\}) \backslash \mathcal{C}} u_i \geq w(\mathcal{N} \backslash \mathcal{C}) \tag{11}$$

Since $\mathcal{C}$ is an arbitrary coalition of CUs, we have that for every coalition $\mathcal{D} = \mathcal{N} \backslash \mathcal{C}$, $\sum_{i \in \mathcal{D} \cup \{o\}} u_i \geq w(\mathcal{D})$, which implies that there is no blocking coalition in the auction. Together with efficiency, we derive $\mathbf{u} \in Core(\mathcal{N})$. ∎

Combining Teorem 3 with the definition of CU utility, we have the following corollaries.

Corollary 1. *Each CU's in-core payment in a VM auction formulated with WDP is at least as high as its VCG payment.*

Corollary 2. *The total revenue in a core-selecting VM auction formulated with WDP is at least as high as that in a VCG auction.*

V. Core-Selecting Payment Rules and Algorithms

We next study several payment rules that can be employed in our core-selecting VM auction, and formulate a linear program and a quadratic program for implementing these rules.

A. Revenue Minimization Rule

Individually rational CUs can try to maximize their utilities through either unilateral or collusive strategic behaviors. In Section IV-B we show that CUs gain no benefits through a form of collusion known as shill bidding. With the relaxation of absolute truthfulness in core-selecting auctions, we are further motivated to minimize CUs' benefits from unilaterally deviating from truth-telling, which in turn, maximizes CUs' incentives towards reporting valuations truthfully.

We start with the notion of bidder-Pareto optimality:

Definition A core-selecting auction is *bidder-Pareto optimal* if it always generates a core outcome such that no other core outcome can improve at least one CU's utility without reducing any other one's in $\mathcal{N}$.

To measure how much CUs are likely to deviate from truthful reporting, we define the incentive profile for a core-selecting auction.

Definition The *incentive profile* of a core-selecting auction M at $\mathbf{v}$ is $\{\theta_i^M(\mathbf{v})\}_{i\in\mathcal{N}}$, where $\theta_i^M(\mathbf{v})$ is i's maximum utility gain by deviating from truthful reporting.

We aim at minimizing CUs' overall economic benefits from deviating from truthful bidding in our core-selecting auction. Consider a core-selecting auction M, it provides *optimal incentives* if there is no core-selecting auction M' such that $\theta_i^{M'}(\mathbf{v}) \leq \theta_i^M(\mathbf{v}), i \in \mathcal{N}$ with strict inequality for some i. Actually, core-selecting auction M provides optimal incentives if and only if M is bidder-Pareto optimal [14].

Suppose that in our efficient VM auction, CU i bids truthfully and wins a bundle of VMs at price p_i. By Corollary 1, p_i is guaranteed to be at least p_i^{VCG}. The following lemma bounds the economic benefit of i [15], [23]:

Lemma 1. *For any efficient auction that produces payments greater than or equal to the VCG payments, the amount that bidder i can benefit by unilaterally deviating from the truthful bidding strategy is no more than $p_i - p_i^{VCG}$.*

Theorem 4. *A core-selecting VM auction formulated with WDP provides optimal incentives for truthful bidding if and only if it is a bidder-Pareto optimal auction.*

Proof: A VM auction formulated with WDP is efficient. From Lemma 1, for a bidder-Pareto optimal, core-selecting VM auction formulated with WDP, the maximum benefit CU i can get is $p_i - p_i^{\text{VCG}}$. Hence the auction is suboptimal exactly when there is another core-selecting auction with higher utilities for all CUs, contradicting the assumption that this auction is bidder-Pareto optimal. ∎

Nonetheless, there may be a broad range of possible bidder-Pareto optimal outcomes in the core. By minimizing the total payments over the core, one can guarantee bidder-Pareto optimality [15], which narrows the space of possible outcomes as well. We further derive the following corollary:

Corollary 3. *A core-selecting VM auction formulated with WDP and employing a revenue-minimization payment rule minimizes CUs' incentives to deviate from truthful bidding.*

Recall the coalitional core constraint, we have

$$\sum_{i\in\mathcal{C}\cup\{o\}} u_i \geq w(\mathcal{C}),\ \forall \mathcal{C} \subseteq \mathcal{N} \tag{12}$$

Assume that CU i receives a bundle $\mathcal{S}_i$. Let $\mathcal{W}$ be the set of winning CUs in WDP, we expand (12) to obtain:

$$\begin{aligned} u_o + \sum_{i\in\mathcal{C}\cap\mathcal{W}} u_i + \sum_{i\in\mathcal{C}\setminus\mathcal{W}} u_i &= \sum_{i\in\mathcal{W}} p_i + \sum_{i\in\mathcal{C}\cap\mathcal{W}} (b_i(\mathcal{S}_i) - p_i) \\ = \sum_{i\in\mathcal{W}\setminus\mathcal{C}} p_i + \sum_{i\in\mathcal{C}\cap\mathcal{W}} b_i(\mathcal{S}_i) &\geq w(\mathcal{C}),\ \forall \mathcal{C} \subseteq \mathcal{N} \end{aligned} \tag{13}$$

Let $\tilde{\mathcal{C}} = \mathcal{C} \cap \mathcal{W}$, then we have $w(\mathcal{C}) \leq w(\tilde{\mathcal{C}} \cup (\mathcal{N}\setminus\mathcal{W}))$ since $\mathcal{C} \subseteq \tilde{\mathcal{C}} \cup (\mathcal{N}\setminus\mathcal{W})$. Since $\mathcal{C}$ is an arbitrary subset of $\mathcal{N}$, (13) is equivalent to

$$\sum_{i\in\mathcal{W}\setminus\tilde{\mathcal{C}}} p_i \geq w(\tilde{\mathcal{C}} \cup (\mathcal{N}\setminus\mathcal{W})) - \sum_{i\in\tilde{\mathcal{C}}} b_i(\mathcal{S}_i),\ \forall \tilde{\mathcal{C}} \subseteq \mathcal{W} \tag{14}$$

Setting $\beta_{\tilde{\mathcal{C}}} = w(\tilde{\mathcal{C}} \cup (\mathcal{N}\setminus\mathcal{W})) - \sum_{i\in\tilde{\mathcal{C}}} b_i(\mathcal{S}_i)$, and denoting the vector of all $\beta_{\tilde{\mathcal{C}}}$ as $\boldsymbol{\beta}$, we can reformulate (14) as

$$A\mathbf{p} \geq \boldsymbol{\beta} \tag{15}$$

where A is a $2^{|\mathcal{W}|-1} \times |\mathcal{W}|$ matrix. In each row $\mathbf{a}_{\tilde{\mathcal{C}}}^T$ of A, the i-th entry equals 0 if CU i is in coalition $\tilde{\mathcal{C}}$ and equals 1 otherwise. The *revenue minimization rule* can be formulated as the following linear program (LP):

$$\delta = \min \mathbf{1}^T \cdot \mathbf{p}$$

subject to:

$$\begin{aligned} A\mathbf{p} &\geq \boldsymbol{\beta} \\ \mathbf{p} &\leq \mathbf{b} \end{aligned}$$

B. Point Reference Rules

The points minimizing the revenue are not unique, hence there is a lack of precision even if we minimize the total payments over the core to ensure bidder-Pareto-optimality. We further define a *point reference rule*, for choosing among these points the one that has the smallest geometric distance from some pre-determined *reference point* $\mathbf{p}'$.

The reference point can be static or dependent on CUs' bids. The point reference rule always leads to a unique revenue-minimizing payment vector, which can be computed by solving the following quadratic program (QP):

$$\min(\mathbf{p} - \mathbf{p}')^T(\mathbf{p} - \mathbf{p}')$$

subject to:

$$\begin{aligned} A\mathbf{p} &\geq \boldsymbol{\beta} \\ \mathbf{p} &\leq \mathbf{b} \\ \mathbf{1}^T \cdot \mathbf{p} &= \delta \end{aligned}$$

We introduce two specific point reference rules. The first is the *VCG-nearest rule*, in which $\mathbf{p}'$ is set to be the VCG point $\mathbf{p}^{\text{VCG}}$. This payment rule is natural in that $\mathbf{p}^{\text{VCG}}$ is known to be the payment point that motivates truthful bidding.

The second is the *constant* $\mathbf{p}'$ *reference rule*. In this rule $\mathbf{p}'$ is some constant payment point pre-defined by the auctioneer. Under this rule, the final payment highly depends on the assumptions of the auctioneer, and the winner with high valuation relative to the auctioneer's expectation shares less of the burden to conquer a coalitional blocking.

We elaborate this with an example. Consider the *origin-nearest rule*, in which $\mathbf{p}' = \mathbf{0}$. There are 18 CPUs and 18 GB storage in the resource pool. Four CUs each submits a single bid for three types of VMs, with the following configuration in CPU units and storage (GB): (1, 1), (1, 3) and (2, 1):

$$b_1(0,0,6)_6^{12} = 100 \qquad b_2(0,4,0)_{12}^4 = 20$$
$$b_3(0,4,6)_{18}^{16} = 60 \qquad b_4(0,0,6)_6^{12} = 50$$

The VCG payments $(50, 0)$ for winners 1 and 2 are not in the core; the pair must raise their combined payment to 60 to keep CU 3 from blocking. If the origin-nearest rule is used, CU 2 will be responsible for this total payment increase with final payments $(50, 10)$. In comparison, the VCG-nearest rule results in a sharing of this burden, with payments $(55, 5)$.

Since the QP is a convex problem, the Karush-Kuhn-Tucker (KKT) conditions are sufficient and necessary for the QP. The KKT conditions indicate that for an optimal solution $\mathbf{p}^*$ to the QP, there exist a vector $\boldsymbol{\lambda} \geq 0$, a vector $\boldsymbol{\omega} \geq 0$, and a scalar $\nu \geq 0$, such that

$$\mathbf{p}^* - \mathbf{p}' - A^T\boldsymbol{\lambda} + I\boldsymbol{\omega} + \mathbf{1}\nu \quad = \mathbf{0} \tag{16}$$

where I is the identity matrix of size $|\mathbf{p}|$. The final payment $\mathbf{p}^*$ can be decomposed as follows for each winning CU i:

$$p_i^* = p_i' + \sum_{\tilde{\mathcal{C}} \in \mathcal{W}\backslash\{i\}} \lambda_{\tilde{\mathcal{C}}} - \omega_i - \nu$$

By utilizing the KKT conditions, it can be shown that the set of constraints $\mathbf{p} \leq \mathbf{b}$ is not necessary under the VCG-nearest rule, which simplifies the QP and improves computational efficiency.

Theorem 5. *Under the VCG-nearest rule, the set of constraints* $\mathbf{p} \leq \mathbf{b}$ *in the QP is unnecessary.*

Proof: By way of contradiction, assume that there is an $i \in \mathcal{W}$, such that the constraint $p_i \leq b_i(\mathcal{S}_i)$ is necessary. Then there exists some $\epsilon > 0$, for which the constraint is still tight when relaxed by ϵ, and for which the solution must change. After relaxation, $b_i(\mathcal{S}_i)$ is increased by ϵ and the set of winners $\mathcal{W}$ due to WDP does not change. Now for the VCG-nearest rule, the KKT necessary and sufficient conditions form the same linear system, since the only affected condition is that of CU i, *i.e.*, $p_i^* = p_i^{\text{VCG}} + \sum_{\tilde{\mathcal{C}} \in \mathcal{W}\backslash\{i\}} \lambda_{\tilde{\mathcal{C}}} - \omega_i - \nu$, which remains unchanged when $b_i(\mathcal{S}_i)$ increases by ϵ. This is because $p_i^{\text{VCG}} = b_i(\mathcal{S}_i) - w(\mathcal{N}) + w(\mathcal{N}\backslash\{i\})$, and the increment ϵ in $b_i(\mathcal{S}_i)$ is cancelled by the increase ϵ in $w(\mathcal{N})$. Now the solution to the linear system does not change when the constraint is relaxed by ϵ, which is a contradiction. ■

Based on Theorem 5, under the VCG-nearest rule, we can solve the following quadratic program instead, which still generates the optimal solution but is easier to solve.

$$\min(\mathbf{p} - \mathbf{p}^{\text{VCG}})^T(\mathbf{p} - \mathbf{p}^{\text{VCG}})$$

subject to:

$$A\mathbf{p} \geq \boldsymbol{\beta}$$
$$\mathbf{1}^T \cdot \mathbf{p} = \delta$$

C. Payment Generation Algorithm

In the payment rules mentioned above, evaluating each $\beta_{\tilde{\mathcal{C}}}$ requires the solution of a WDP, so there will be $2^{|\mathcal{W}|-1}$ non-empty coalitions to consider, prohibitive for a large $|\mathcal{W}|$. However, by adapting the core-constraint generation process given by Day and Raghavan [15], a VCG-nearest in-core payment generation procedure can be employed to reduce the complexity, as shown in Algorithm 1. Instead of enumerating all the possibilities of non-empty coalitions, it finds blocking coalitions effectively by raising payments from the reference point, thereby reducing the complexity. Algorithm 1 can further be easily adapted to solve the in-core payment vectors under other payment rules. The following theorem shows that Algorithm 1 always yields the VCG-nearest in-core payment.

Algorithm 1: VCG-Nearest In-Core Payment Generation

```
1  Set t := 0, payment vector p^t := p^VCG, coefficient matrix
   A^t := ∅, and vector β^t := ∅;
2  while True do
3      t := t + 1;
4      for CU i ∈ N do
5          for bundle S bid by i do
6              b_i^t(S) := b_i(S) − (b_i(S_i) − p_i^{t−1});
7          end
8      end
9      Calculate w^t(N) with b^t, and the set of winning CUs C^t
       is the most violated coalition;
10     if w^t(N) ≤ 1^T p^{t−1} then break;
11     C̃^t := C^t ∩ W;
12     β_{C̃^t} := w^t(C̃^t ∪ (N\W)) − Σ_{i∈C̃^t} b_i^t(S_i);
13     Append the corresponding row a^T_{C̃^t} and new entry β_{C̃^t} to
       A^{t−1} and β^{t−1} to form A^t and β^t, respectively;
14     Solve the LP with A^t and β^t, obtaining δ^t;
15     Solve the QP with A^t, β^t and δ^t, obtaining p^t;
16 end
17 p* := p^{t−1} is the final payment vector.
```

Theorem 6. *Algorithm 1 always yields VCG-nearest in-core payments.*

Proof: We claim that when the algorithm terminates, $\mathbf{p}^{t-1}$ is in the core. Suppose this is not true, then (14) does not hold:

$$\sum_{i \in \mathcal{W}\backslash\tilde{\mathcal{C}}} p_i^{t-1} < w(\tilde{\mathcal{C}} \cup (\mathcal{N}\backslash\mathcal{W})) - \sum_{i \in \tilde{\mathcal{C}}} b_i(\mathcal{S}_i), \ \exists \tilde{\mathcal{C}} \subseteq \mathcal{W} \tag{17}$$

From Algorithm 1 (Line 6), we have

$$\begin{cases} b_i^t(\mathcal{S}_i) = p_i^{t-1}, & i \in \mathcal{W} \\ b_i^t(\mathcal{S}) = b_i(\mathcal{S}), & i \notin \mathcal{W} \end{cases} \tag{18}$$

Eq. (18) and the stopping criterion (Line 10) lead to:

$$w^t(\tilde{\mathcal{C}} \cup (\mathcal{N}\backslash\mathcal{W})) \leq w^t(\mathcal{N}) \leq \sum_{i \in \mathcal{W}\backslash\tilde{\mathcal{C}}} p_i^{t-1} + \sum_{i \in \tilde{\mathcal{C}}} b_i^t(\mathcal{S}_i) \tag{19}$$

We further have the following:

$$\sum_{i \in \tilde{\mathcal{C}}} (b_i(\mathcal{S}_i) - b_i^t(\mathcal{S}_i)) <_1 w(\tilde{\mathcal{C}} \cup (\mathcal{N}\backslash\mathcal{W})) - w^t(\tilde{\mathcal{C}} \cup (\mathcal{N}\backslash\mathcal{W}))$$
$$\leq_2 \sum_{i \in \tilde{\mathcal{W}}} (b_i(\tilde{\mathcal{S}}_i) - b_i^t(\tilde{\mathcal{S}}_i)) =_3 \sum_{i \in \tilde{\mathcal{W}} \cap \tilde{\mathcal{C}}} (b_i(\tilde{\mathcal{S}}_i) - b_i^t(\tilde{\mathcal{S}}_i)) \tag{20}$$

where $\tilde{\mathcal{W}}$ is the set of winners to $w(\tilde{\mathcal{C}} \cup (\mathcal{N}\backslash\mathcal{W}))$ with bundle $\tilde{\mathcal{S}}_i$ allocated to CU i. $<_1$ is due to (17) and (19). Winning set $\tilde{\mathcal{W}}$ with bundle $\tilde{\mathcal{S}}_i$ allocated to i is a feasible solution to $w^t(\tilde{\mathcal{C}} \cup (\mathcal{N}\backslash\mathcal{W}))$, and hence $\sum_{i \in \tilde{\mathcal{W}}} b_i^t(\tilde{\mathcal{S}}_i) \leq w^t(\tilde{\mathcal{C}} \cup (\mathcal{N}\backslash\mathcal{W}))$, which leads to $\leq_2$. $=_3$ is due to Eq. (18) and the fact that $\tilde{\mathcal{W}} \subseteq \tilde{C} \cup (\mathcal{N} \backslash \mathcal{W})$. In view of Line 6, (20) indicates:

$$\sum_{i \in \tilde{\mathcal{C}}} (b_i(\mathcal{S}_i) - p_i^{t-1}) < \sum_{i \in \tilde{\mathcal{W}} \cap \tilde{\mathcal{C}}} (b_i(\mathcal{S}_i) - p_i^{t-1}) \tag{21}$$

(21) does not hold and our claim is hence correct. The in-core payment is VCG-nearest since the correlated LP and QP are solved. The convergence of this algorithm is guaranteed because only a finite number of blocking constraints may be generated, and because the core always contains at least the trivial first-price payment solution (Theorem 2). ■

VI. Simulation Studies

A. Simulation Environment

We consider 7 types of VMs configured from 4 types of resources (Table I). The default amount of resources in the cloud is $R = (5000, 14000, 16000, 700000)$, and scales from $0.5R$ to $4R$ in other cases. The number of bids submitted by a CU is uniform on $[1, 6]$. The bundles of VMs that a CU requests and the bid prices are generated according to uniform and normal distributions, respectively; we refer to this setting as "uniform-normal". For each set of auction parameters, the results shown are averaged over 100 simulation executions.

We adopt the following performance criteria. (a) Social welfare, measured as the sum of reported values from all the winning CUs. (b) Resource utilization, the percentage of resources that are provisioned and sold. (c) User satisfaction, the percentage of CUs winning a VM bundle. (d) CP revenue.

B. VCG-Nearest vs. Origin-Nearest Payment Rules

Using the example in Section V-B, we show that the use of the origin-nearest rule ($\mathbf{p}' = \mathbf{0}$) can lead to a high-valued winner shouldering little of the monetary burden, if any, to conquer a blocking coalition. When the VCG payment is not in the core, we define the *monetary burden* [16] of CU $\bar{i}$ as:

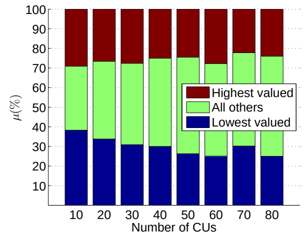
$$\mu_{\bar{i}} = \frac{p_{\bar{i}}^* - p_{\bar{i}}^{\text{VCG}}}{\sum_{i \in \mathcal{N}} (p_i^* - p_i^{\text{VCG}})} \tag{22}$$

where $\bar{i}$ is the index of the highest-valued winning CU or the lowest-valued winning CU.

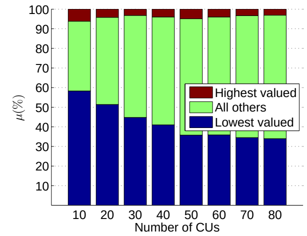
Using this measure, Fig. 3 demonstrates that this phenomenon is not peculiar to a carefully constructed example, but indeed occurs frequently from random data sets. For the VCG-nearest computations, the statistic (22) averages $\sim$27% for the highest-valued winner, while the origin-nearest computations result in a value of $\sim$5% for the highest-valued winner; the use of the origin-nearest rule results in high-valued winners shouldering less of the burden of conquering blocking coalitions. Similarly, the lowest-valued winners pay $\sim$29% of the burden under the VCG-nearest rule, while they pay 42% under the origin-nearest implementation. From these figures, this disparity between these two approaches is most pronounced when the number of CUs is small.

C. Allocation Results

We use "xVyR" to denote the auction settings, where x represents the number of types of VMs auctioned and y represents that an amount yR of resources are available (R is a vector defined in Sec. VI-A). When $x = 4$, we are using the VMs from Table I with odd IDs.



(a) VCG-nearest



(b) Origin-nearest

Fig. 3. The figures show the monetary burden shouldered by the winner(s) under (a) the VCG-nearest rule, and (b) the origin-nearest rule.

Fig. 4 plots the social welfare, resource utilization and user satisfaction under core-selecting auctions, by changing the number of CUs from 10 to 80. Both static and dynamic resource provisioning are tested. When SRP is employed, the resource provisioning scheme, *i.e.*, the vector $(n_1, \ldots, n_m)$, is determined by solving $w(\mathcal{N})$ for 100 times and multiplying their average with some value such that at least one type of resource is depleted. This way we try to improve the performance of SRP through predicting the market demand, for comparison with DRP.

In Fig. 4a, the relative social welfare is at least 10% larger under DRP than under SRP. An interesting observation is that '4V1R static' is always higher than '7V1R static', under an equal amount of resources. This is because in the former scenario, the 4 types of VMs on sale are finer-grained in their configurations, so that a larger number of VM instances can be constructed for sale, leading to a higher social welfare.

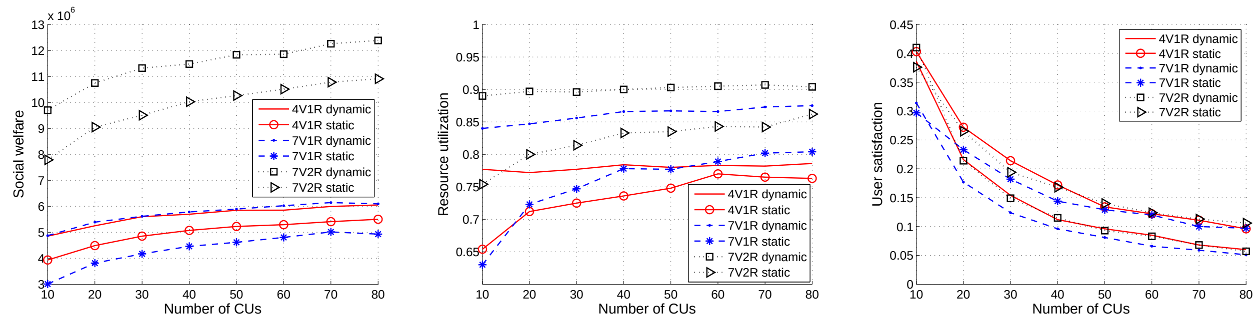
Fig. 4b shows that when our 3-D auction mechanism is employed, with resources dynamically provisioned, resource utilization does not apparently change, and is as high as 90% in '7V2R dynamic'. In contrast, under static provisioning, resource utilization increases with the number of CUs, and is always lower than that under DRP. Despite the fact that we already attempted to improve the performance of SRP, the gap is still as high as 21%.

An interesting phenomenon in Fig. 4c is that user satisfaction decreases and remains rather low when the number of CUs is large. This is due to the fact that when there are enough participants in the auction, there exist a few CUs requesting large bundles of VMs and submitting high bids to exclude other CUs. We can see that SRP performs better than DRP in terms of user satisfaction.

D. The Role of Bid Distribution

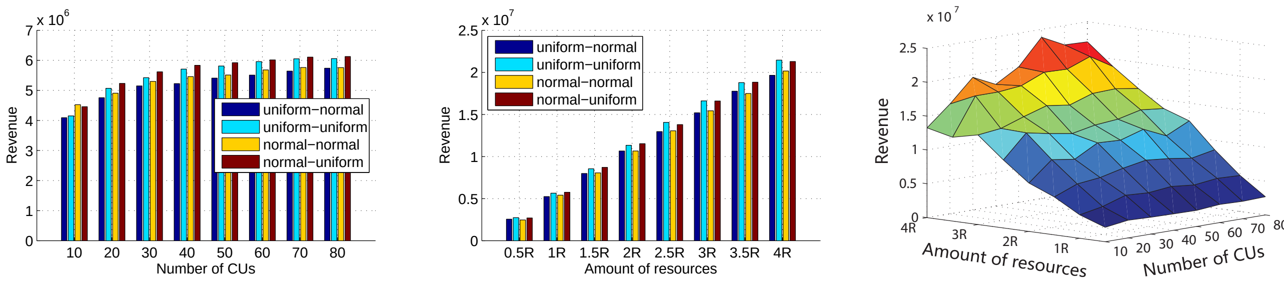
To investigate the influence on revenues by the bid distribution, in addition to "uniform-normal", we further adopt another three settings for comparison, which are "uniform-uniform", "normal-normal" and "normal-uniform". From Fig. 5a and Fig. 5b, we find that the performance of core-selecting auctions is rather stable. Even when bidding instances are generated in different ways, the revenues do not deteriorate.

In Fig. 5c, CUs' bids are synthesized from Google Cluster Data [17] with bid prices generated according to normal



(a) Social welfare (b) Resource utilization (c) User satisfaction

Fig. 4. Performance of the VM allocation result under core-selecting auctions.



(a) Revenue vs. Number of CUs (b) Revenue vs. Amount of resources (c) Revenue against Google Cluster Data

Fig. 5. Influence of bid distribution.

distribution. We find that revenues increase moderately with the number of CUs and increase approximately linearly with increasing amount of resources, which is consistent with what we observe in Fig. 5a and in Fig. 5b.

VII. Conclusions

Core-selecting auctions are emerging as an effective combinatorial auction mechanism for allocating bundles of goods. They guarantee economic efficiency, are proof to shill bidding, and outperform VCG auctions in revenue. This work is the first that designs core-selecting auctions for the cloud computing market, and advances the state-of-the-art of VM auction design by generalizing static resource provisioning to dynamic resource provisioning, and from homogeneous VM instances to heterogeneous VM instances.

References

[1] A. Gulati, A. Holler, M. Ji, G. Shanmuganathan, C. Waldspurger, and X. Zhu, "Vmware distributed resource management: Design, implementation, and lessons learned," *VMware Technical Journal*, vol. 1, no. 1, pp. 45–64, June 2012.

[2] *Amazon EC2*, http://aws.amazon.com/ec2/.

[3] H. Zhang, B. Li, H. Jiang, F. Liu, A. V. Vasilakos, and J. Liu, "A framework for truthful online auctions in cloud computing with heterogeneous user demands," in *Proc. of IEEE INFOCOM*, April 2013.

[4] S. Zaman and D. Grosu, "Combinatorial auction-based mechanisms for vm provisioning and allocation in clouds," in *IEEE/ACM CCGrid*, May 2012, pp. 729–734.

[5] J. M. Pujol, V. Erramilli, G. Siganos, X. Yang, N. Laoutaris, P. Chhabra, and P. Rodriguez, "The little engine(s) that could: Scaling online social networks," in *Proc. of ACM SIGCOMM*, October 2010, pp. 375–386.

[6] S. Zaman and D. Grosu, "Combinatorial auction-based allocation of virtual machine instances in clouds," *Journal of Parallel and Distributed Computing*, vol. 73, no. 4, pp. 495–508, April 2013.

[7] *DigitalOcean*, https://www.digitalocean.com/.

[8] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, March 1961.

[9] E. H. Clarke, "Multipart pricing of public goods," *Public Choice*, vol. 11, no. 1, pp. 17–33, September 1971.

[10] T. Groves, "Incentives in teams," *Econometrica: Journal of the Econometric Society*, vol. 41, no. 4, pp. 617–631, July 1973.

[11] B. Holmstrom, "Groves scheme on restricted domains," *The Econometric Society*, vol. 47, no. 5, pp. 1137–1144, September 1979.

[12] M. Yokoo, Y. Sakurai, and S. Matsubara, "The effect of false-name bids in combinatorial auctions: New fraud in internet auctions," *Games and Economic Behavior*, vol. 46, no. 1, pp. 174–188, January 2004.

[13] *CPLEX Optimizer*, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[14] R. Day and P. Milgrom, "Core-selecting package auctions," *International Journal of Game Theory*, vol. 36, no. 3-4, pp. 393–407, March 2008.

[15] R. W. Day and S. Raghavan, "Fair payments for efficient allocations in public sector combinatorial auctions," *Management Science*, vol. 53, no. 9, pp. 1389–1406, September 2007.

[16] R. W. Day and P. Cramton, "Quadratic core-selecting payment rules for combinatorial auctions," *Operations Research*, vol. 60, no. 3, pp. 588–603, May 2012.

[17] *Google Cluster Data*, https://code.google.com/p/googleclusterdata/.

[18] H. Li, C. Wu, Z. Li, and F. Lau, "Virtual machine trading in a federation of clouds: Individual profit and social welfare maximization," 2013, technical report, arXiv:1304.6491.

[19] ——, "Profit-maximizing virtual machine trading in a federation of selfish clouds," in *Proc. of IEEE INFOCOM (mini-conf.)*, April 2013.

[20] L. Zhang, Z. Li, and C. Wu, "Dynamic resource provisioning in cloud computing: A randomized auction approach," in *Proc. of IEEE INFOCOM*, April 2014.

[21] E. Angel, E. Bampis, and F. Pascual, "Truthful algorithms for scheduling selfish tasks on parallel machines," *Theoretical Computer Science*, vol. 369, no. 1, pp. 157–168, December 2006.

[22] Q. Wang, K. Ren, and X. Meng, "When cloud meets ebay: Towards effective pricing for cloud computing," in *Proc. of IEEE INFOCOM*, March 2012.

[23] Y. Zhu, B. Li, and Z. Li, "Core-selecting combinatorial auction design for secondary spectrum markets," in *Proc. of IEEE INFOCOM*, April 2013.