A Mechanism Design Approach to Resource Procurement in Cloud Computing

Abhinandan S. Prasad, *Member*, *IEEE*, and Shrisha Rao, *Senior Member*, *IEEE*

Abstract—We present a cloud resource procurement approach which not only automates the selection of an appropriate cloud vendor but also implements dynamic pricing. Three possible mechanisms are suggested for cloud resource procurement: cloud-dominant strategy incentive compatible (C-DSIC), cloud-Bayesian incentive compatible (C-BIC), and cloud optimal (C-OPT). C-DSIC is dominant strategy incentive compatible, based on the VCG mechanism, and is a low-bid Vickrey auction. C-BIC is Bayesian incentive compatible, which achieves budget balance. C-BIC does not satisfy individual rationality. In C-DSIC and C-BIC, the cloud vendor who charges the lowest cost per unit QoS is declared the winner. In C-OPT, the cloud vendor with the least virtual cost is declared the winner. C-OPT overcomes the limitations of both C-DSIC and C-BIC. C-OPT is not only Bayesian incentive compatible, but also individually rational. Our experiments indicate that the resource procurement cost decreases with increase in number of cloud vendors irrespective of the mechanisms. We also propose a procurement module for a cloud broker which can implement C-DSIC, C-BIC, or C-OPT to perform resource procurement in a cloud computing context. A cloud broker with such a procurement module enables users to automate the choice of a cloud vendor among many with diverse offerings, and is also an essential first step toward implementing dynamic pricing in the cloud.

Index Terms—Cloud computing, mechanism design, cloud broker, resource procurement, reverse auctions, multiattribute auctions, dynamic pricing

1 Introduction

Cloud computing is an increasingly popular paradigm of offering services over the Internet [1]. It is also an active area of research, and the popularity of this paradigm is growing rapidly. Many companies like Amazon, IBM, Google, salesforce.com, Unisys, and so on, now offer cloud services. The main advantage of cloud computing is the ability to provision IT resources on demand (thus avoiding the problems of over-provisioning and under-provisioning which are commonly seen with organizations that have widely variable requirements due to growth/shrinkage, seasonal peaks, and valleys, etc.). The resources offered may include storage, CPU processing power, IT services, and so on. These resources are often geographically distant from users.

We can say the following:

- A cloud user is a person or an organization (such as an SME—small and medium enterprise) that uses cloud services.
- A cloud vendor is an organization that offers cloud services for use on payment.
- A cloud broker [2] is a middleware that interacts with service providers on behalf of the user. It is responsible for configuring the user's settings suitably and for procuring resources.

• *A.S. Prasad is with Alcatel Lucent India, Manyata Embassy Business Park, Silver Oak - Wing A, Outer Ring Road, Nagavara, Bangalore 560 045, India. E-mail: abhinandansp@ieee.org.*
• *S. Rao is with the International Institute of Information Technology-Bangalore, 26/C Electronics City, Hosur Road, Bangalore 560 100, India. E-mail: shrao@ieee.org.*

Manuscript received 13 Sept. 2012; revised 18 Feb. 2013; accepted 24 Apr. 2013; published online 3 May 2013.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TCSI-2012-09-0659.
Digital Object Identifier no. 10.1109/TC.2013.106.

Resource procurement of cloud resources is an interesting and yet unexplored area in cloud computing. Cloud vendors follow a fixed pricing strategy ("pay as you go") for pricing their resources and do not provide any incentive to their users to adjust consumption patterns according to availability or other factors.

Consider, for example, a user who wants to use a service in the form of an application hosted on a cloud. There are cloud vendors who provide versions of that application at different prices and with varying quality-of-service (QoS) parameters. The user has to go through the specifications of each cloud vendor to select the appropriate one, to obtain the service within budget and of the desired quality. In case of an organization acting as a user, this selection is quite complex and challenging [3]. Also, the companies offering cloud services, and their offerings, change continually. So, given the large and varying multitude of cloud vendors, it is very tedious to select the most appropriate one manually. Hence, there is a need for a scalable and automated method to perform resource procurement in the cloud. Rochwerger *et al.* [4] observe that while cloud vendors do not yet offer standardized services, they will need to do so, and that the "federated cloud has huge potential." In that event, it would become possible to mix and interchange resources offered by different cloud vendors and to automate the procurement of such resources.

If resource procurement is automated, then the challenge would be to find the appropriate location where the solution can be deployed. One manner in which our solution may be deployed is by the use of a cloud broker that implements our approach. Cloud brokerages form an important research area [5], and the cloud brokerage business was expected to be worth \$150 billion by 2013 [6].

Most cloud vendors use the pay-as-you-go model. Many are loath to negotiate contracts as they lack understanding of a sound theoretical basis for dynamic pricing. The default agreement offered by a vendor often contractually benefits the vendor but not the user, resulting in a mismatch with user requirements [7]. Hence, this kind of pricing favors the cloud vendor. Also, there is no clear commitment on SLAs [7]. Dynamic pricing is the solution for these kind of problems [8].

Bichler *et al.* [9] state that uncertainty about the prices of goods and lack of knowledge about market participants are obstacles to dynamic pricing. Auctions are in particular helpful in this kind of situation [9], [10], [11]. If the buyer is an auctioneer and the suppliers are bidders, then the auction is called a *reverse auction*. Reverse auctions are widely used across many industries, and also especially by governments to procure resources. Reverse auctions are preferred over other auctions for procuring resources because competitive bidding in these type of auctions reduces procurement costs and limits the influences of undesirable factors like nepotism and political ties [12].

Resource procurement can be accomplished using conventional methods [13], [14], [15] and economic models [16]. The conventional models assume that resource providers are nonstrategic [13], [14], [15], whereas economic models assume that resource providers are rational and intelligent. In conventional methods, a user pays for the consumed service. In economic models, a user pays based on the value derived from the service [16]. Hence, economic models are more appropriate in the context of cloud services.

An important feature of economic models is the distribution of incentives to bidders, which are cloud vendors in our domain. However, this means that cloud vendors may not act truthfully and may seek to maximize their incentives using improper behavior. Game-theoretic models cannot enforce the structure in games. Mechanism design enables the social planner to design the game according to his wish. So the social planner can implement strategies to motivate participants to act truthfully.

The important contributions of this work are:

- procurement mechanisms for implementing dynamic pricing, and
- novel procurement module based on mechanism design for a cloud broker.

Dynamic pricing increases user welfare, facilitates healthy competition among vendors, and increases the efficiency of cloud resource usage [17]. Auctions are one way of implementing dynamic pricing [9]. Dynamic pricing is not only advantageous for cloud users but also maximizes the profit for vendors [18]. The mechanisms proposed in this paper are based on reverse auctions and are more appropriate for implementing dynamic pricing.

The procurement module enables the cloud broker to automate resource procurement. In our procurement module, the user sends the specifications to the cloud broker and requests for resources. The cloud broker sends the user specification to all cloud vendors. The cloud vendors respond with cost and QoS parameters of their services. We do not consider implementation issues like caching, refresh, and so on, of cost and QoS by the broker. The cloud broker assigns weights for different QoS parameters using analytic hierarchy process (AHP), which are scaled before computing a weighted QoS score. This step is called normalization. If normalization is not done, then it is not possible to compare different QoS specifications. The cloud broker implements one of cloud-dominant strategy incentive compatible (C-DSIC), cloud-Bayesian incentive compatible (C-BIC), or cloud-optimal (C-OPT) mechanisms. The winner is determined based on the mechanism implemented. The cloud broker notifies both winner and user. Finally, the cloud broker pays money to the cloud vendors according to the payment function of the mechanism. This is called the *procurement cost*.

We assume that cloud vendors are *selfish* (interested in maximizing their own profits) and *rational* (able to appropriately calculate values and derive choices based on available information, rather than relying merely on past experience). Hence, there is a possibility of overbidding and underbidding [19]. As is standard, our assumption of selfishness is limited only to the pricing aspect, but we assume that vendors are truthful in technical matters; for example, if a cloud vendor claims 99.99 percent uptime for a service, then we assume that this is true information.

Incentives are offered to the cloud vendors to make truth revelation the best strategy. The right amount of incentive offered to induce truth is called *incentive compatibility*. There are two types of incentive compatibility: *dominant strategy incentive compatibility* (DSIC) and *Bayesian incentive compatibility* (BIC). These are the only ways of implementing incentive compatibility. In DSIC, the optimal strategy for each cloud vendor is to report true valuation irrespective of other cloud vendor's valuation. In BIC, the optimal strategy is to report true valuation only if all the other cloud vendor's report true valuation.

In this work, we present three possible mechanisms:

- C-DSIC: This is a dominant strategy incentive compatible mechanism. It is based on the VCG mechanism (see [20] for an explanation of the VCG mechanism). In C-DSIC, the best strategy for a cloud vendor is to bid truthfully. The ratio of cost and QoS is computed for each cloud vendor. The cloud vendor with the lowest ratio of cost to QoS is the winner. The payment rule is based on the VCG mechanism. The user pays the price as per the next lowest bid. C-DSIC is a low-bid Vickrey auction. C-DSIC achieves *allocative efficiency* (objects are allocated to the cloud vendors who value them most) and *individual rationality* (cloud vendors get negative payoff if they withdraw from the auction) but it is not *budget balanced* (there is no external funding in the system). If all cloud vendors use the same probability distribution of price and QoS, then C-DSIC is to be preferred.
- C-BIC: This mechanism is based on the dAGVA mechanism [21]. In C-BIC, each cloud vendor contributes a participation fee. This money is used for paying other cloud vendors. Hence, C-BIC is *budget balanced* and *allocative efficient*. In this mechanism also, the vendor with lowest cost and QoS ratio is declared the winner. The procurement cost for the user is less here compared with C-DSIC. C-BIC does not satisfy *individual rationality* but achieves allocative efficiency and budget balance. C-BIC is

suitable for government organizations. Generally, the participants in government-sponsored procurement auctions pay a participation fee and this is the accepted practice in them. The loss of a cloud vendor's money in the C-BIC can be viewed as the fee for participating in procurement auction.

- C-OPT: This mechanism is proposed to overcome the limitations of both C-DSIC and C-BIC. The winner determination and payment rule are different compared to C-DSIC and C-BIC. We compute *virtual cost* [12], [22] for every cloud vendor. This virtual cost is used to determine the winner. In our model, virtual cost is a function of cost and QoS. We rank the cloud vendors based on their virtual costs. The cloud vendor with lowest virtual cost is declared the winner. The payment is computed based on the quoted cost and the expectation of the allocation.

The simulation of price and QoS are not supported in popular cloud environments like Eucalyptus, CloudSim, and so on. Cloudsim [23] does not support auction protocols [24]. Hence, we develop a simulation tool using Java without compromising on cloud properties.

We performed experiments in two scenarios to address the lack of standard QoS distributions in the context of cloud computing. In the first scenario, prices are lognormally distributed and QoS parameters are uniformly distributed. In the second scenario, price is lognormally distributed, whereas QoS values are distributed normally. (Prices of stocks and other goods are frequently lognormally distributed [25], and we find this distribution also fits published cloud service price data. See Limpert *et al.* [26] for other examples of the same distribution in many domains.) The procurement cost for each mechanism in every scenario is calculated in the presence of different cloud vendors.

We observed that the procurement cost decreases as the number of cloud vendors increases. Also, the procurement cost is lower in C-BIC compared to the other mechanisms. The procurement cost in C-OPT is slightly higher than C-DSIC, except in a very few cases. The procurement cost in C-OPT depends on the cost valuation of the user.

According to the Gibbard-Satterthwaite impossibility theorem [27], only dictatorial social choice functions are dominant strategy incentive compatible. In dictatorial social choice function, only certain agents are preferred every time. Hence, these types of functions cannot be used in real-world scenarios. So there are some cases where the impossibility theorem does not hold (like quasilinear environments, etc.).

There is no single mechanism which can be applied to all the scenarios. C-DISC is to be preferred as long as cloud vendors have quasilinear utility function. C-BIC is to be preferred when the social planner/cloud broker wants Bayesian Nash equilibrium. C-OPT satisfies all the properties except allocative efficiency.

The rest of the paper is organized as follows. Section 2 presents similar work in grid and cloud computing, and touches on some important concepts in mechanism design. Section 3 presents the QoS scaling and cloud resource procurement models. Section 4 presents the C-DSIC, C-BIC, and C-OPT mechanisms. Section 5 describes the experiments and analyses of the proposed mechanisms. Section 6 proposes the architecture of the procurement module for a cloud broker using these approaches. Section 7 concludes this paper.

2 Related Work

2.1 Resource Allocation in Grid and Cloud

Resource allocation is an important challenge in today's Internet, especially in large distributed systems like Grid, cloud, and so on. Resource allocation is a very active area of research in Grid [16], [28], [29], [30]. These resources are owned by the companies and are mostly distributed geographically. Resource allocation algorithms are generally based on one of these types of models:

1. conventional models, and
2. economic and game-theoretic models.

Conventional models [13], [14], [15] require global knowledge and complete information. These algorithms are mostly centralized in nature. The cost models of the centralized algorithms derive cost based on the usage of the resources.

Economic models for resource allocation are very popular. Economic models of resource management are not only decentralized but also offer incentives to participants. These models derive cost based on the value the user derives from the services [16]. Most resource allocation algorithms based on economic models rely on single market mechanisms. Vilajosana *et al.* [31] develop a configurable auction server that gives the ability to configure markets dynamically. Buyya *et al.* [16] use economic models like commodity market, posted price, and so on, for developing a grid resource broker for resource management.

Generally, an Internet Service Provider (ISP) sets the price without consulting the consumers. This pricing scheme is not Pareto optimal. Hence, Cao *et al.* [32] use game theory to determine the pricing based on quality of service. They model the pricing as a cooperative bargaining game. Also, they extend the work for two competitive ISPs and compute a Nash equilibrium point so that the ISPs and the user cannot decide the price arbitrarily. Narahari *et al.* [19] describe mechanisms based on dominant strategy and Bayesian incentive compatibility. Sometimes, economic models are ineffective with respect to sharing. Mingbiao and Shengli [33] overcomes this problem. Subramoniam et al. [34] use commodity market models to perform resource allocation in Grid. Xhafa and Kolodziej [35] not only survey game-theoretic-based resource allocation models, but also propose their solution based on metaheuristic methods. Parsa *et al.* [36] use a double auction mechanism for performing resource allocation.

Ismail *et al.* [37] propose a formal model for evaluating resource allocation algorithms in Grids. Shu [38] uses a "quantum chromosomes" genetic algorithm to solve the problem of resource allocation in Grid. Li *et al.* [39] use particle swarm optimization to perform resource allocation. Shah *et al.* [40] present a linear programming formulation of the resource allocation problem and compare the efficiency of existing algorithms which solve this problem. Li and Qi [41] present a grid resource allocation algorithm based on fuzzy clustering. This algorithm assigns resources based on task need and also performs reservation of the resources.

Cloud computing is evolved from the Grid. Lin *et al.* [42] use dynamic auctions (based on the Vickrey auction) to perform resource allocation. Cloud users bid for resources and the highest bidder wins the auction. The winner pays the second highest price. Lin *et al.* [42] also introduce the concept of off-peak and peak pricing periods. They assume that all users behave truthfully, which is not always the case in the real world. They also do not discuss the enforcement of the truthfulness property. Narahari *et al.* [19] propose mechanisms for procurement of resources for sweep type jobs in Grid. These mechanisms cannot be applied directly to the cloud. In cloud, the resources are not limited to sweep type jobs. They can be SaaS, PaaS, IaaS, and so on. This work makes use of existing mathematical models like Vickrey auction, and so on. We apply these models to design reverse auctions to procure cloud resources.

2.2 Mechanism Design

The main goal of mechanism design is to implement system-wide solutions to problems that involve multiple self-interested agents, given private information about their preferences [19], [20]. It can also be viewed as the design of a framework of protocols that would foster particular ways of interaction among agents with known behavioral characteristics, to bring about a globally desirable outcome [19].

The work of Nisan and Ronen [43] is considered seminal in the field of algorithmic mechanism design. They successfully use the concepts of mechanism design [20] for solving scheduling problems.

In nonstrategic social choice theory, agents have preferences but they do not try to obfuscate them to maximize their utility [21]. Mechanism design is a strategic version of social choice theory where agents try to maximize their individual payoffs [21]. The goal of mechanism design is to design social choice and payment functions. We refer to standard texts [19], [20], [21] for an in-depth treatment of mechanism design.

Narahari *et al.* [19] apply concepts of mechanism design to solve sponsored search auctions and resource procurement in grid computing. They design three mechanisms for procuring resources in Grid. The mechanisms presented are incentive compatible and optimal. They also design incentive compatible broadcast protocols for ad hoc networks.

2.3 Optimal Multiattribute Auctions

We refer to [11] for a comprehensive introduction to auction theory, including the various types of auctions, their characteristics, and their applications in computing.

In traditional auctions, only price is considered. It is difficult to account for nonnumerical attributes like quality, and so on, which are important in the real world. On the other hand, multiattribute auctions take attributes like quality, and so on, into account. Hence, multiattribute auctions are interesting and challenging. Che [44] proposes a scoring rule (weights for each attribute) to compute a final score. Once the final score is computed, then the traditional auction is performed. Branco [45] describes the properties of optimal multiattribute auctions and proposes a two-stage procurement mechanism. In the first stage, the winner is determined. In the next stage, bargaining is performed for desired quality. Bichler and Kalagnanam [46] analyze the problem of winner determination in the case of multiple sourcing. They also extend multiattribute auctions for configurable offers. According to the authors, multiattribute auctions achieve higher market efficiency compared to traditional single attribute auctions. Ronen and Saberi [47] prove that the minimum approximation ratio achieved in deterministic polynomial time ascending auctions is $\frac{3}{4}$. They also prove that if the dependence between the agents' valuations is bounded, then the approximation ratio achieved is close to 1. Chandrashekar *et al.* [48] present the state of the art in the area of auction mechanisms for electronic procurement.

Ronen and Lehmann [49] present a generic method to construct optimal multiattribute auctions, a method that can be applied to various multiattribute auction designs.

Wang *et al.* [50] present two kinds of multiattribute auction models based on the scoring rule and bidding objective functions.

3 System Model

Our system model is based on Narahari *et al.* [19]. In game theory, we assume that players are rational and have common knowledge and private information. Rationality implies that goal is to maximize payoff. In our model, cloud vendors are rational. Hence, cloud vendors are risk neutral. The concepts of risk neutral and quasilinear are described in detail elsewhere [21].

Each cloud user has resource requirements. The users perform reverse auctions for procuring resources (which are also called *procurement auctions*). Cloud vendors offer resources, but with varying costs and quality metrics. The goal of the cloud user is to minimize the total cost of procuring resources without compromising quality of service. To minimize the procurement cost, it is necessary for the cloud user to know the real costs of cloud vendors.

A user announces its specifications for desired resources and quality of service to all cloud vendors, with the broker acting as a middleman. The cloud vendors decide whether to participate in the auction based on the user information and submit their bids to the broker. The broker aggregates the bidding information and selects the appropriate cloud vendor. Cloud vendors are rational and intelligent. Hence, one of them might bid with a false valuation to maximize its utility. The goal of providing incentives is to encourage truthful bidding.

3.1 QoS Scaling

Cloud vendors provide different resources with different quality-of-service levels. Hence, the QoS parameters are not the same for all cloud vendors. For example, one cloud vendor may guarantee 99 percent uptime and another 99.9 percent uptime, and so on. Also, the QoS parameters can be either *positive* or *negative*. A QoS parameter is called positive if a higher value of that parameter denotes a higher quality of service (i.e., a higher value of the parameter is more desirable than a lower one—for example, network bandwidth), and it is called negative if a lower value of the parameter denotes a higher quality of service (i.e., a lower value of the parameter is more desirable than a lower one—for example, network latency).

The comparison of different QoS parameters is a common problem in multiple criteria decision making. Zeng *et al.* [51] use a well-known method called simple additive weighting (SAW) [52] to perform comparison of quality attributes of web services. Also, Mohabey *et al.* [53]

TABLE 1
AHP Score Table

QoS Specification	Score
100 <Bandwidth < 300kbps	2
300kbps < Bandwidth < 1kbps	4
5 <Latency $\leq$ 10 ms	5
10 < Latency > 20 ms	3
Turnaround time > 24 hrs	2
Turnaround time < 10 hrs	4

suggest use of the SAW method to scale and normalize QoS to perform combinatorial procurement auctions of web services. We use the SAW method to perform QoS normalization in cloud services.

The QoS parameters given by cloud vendors may be textual, like "99 percent uptime," and so on. The SAW technique works only if the values of QoS parameters are integers. Hence, it is necessary to assign a suitable value for every QoS attribute. The analytic hierarchy process [54] is a well-known technique used in these kinds of situations.

The AHP is commonly applied in vendor selection [55], [56]. Generally, it is very difficult for any customer to compare vendors unless the criteria for cloud vendor selection are defined, and the AHP defines clear criteria and helps to arrive at a consensus decision [55].

In our work, scores are assigned to QoS parameters using the AHP, based on user criteria. Let $\gamma_{i,k}$ be the value of the QoS parameter k of cloud vendor i. This $\gamma_{i,k}$ is determined using the AHP.

We construct a matrix $\Gamma = \{\gamma_{i,j}; 1 \leq i \leq n, i \leq j \leq k\}$. Each row of matrix Γ corresponds to the QoS parameters of the cloud vendor i. Let $\alpha_i^{\max}$ and $\alpha_i^{\min}$ be the maximum and minimum values of the QoS of cloud vendor i. Also, $\alpha_i^{\min} \leq \gamma_{i,j} \leq \alpha_i^{\max}$.

Let $B = \{\beta_{i,j}; 1 \leq i \leq n, i \leq j \leq k\}$ be the matrix and $\beta_{i,j}$ the normalized value of the QoS parameter $\gamma_{i,j}$.

The SAW method involves two stages. They are:

1. *Scaling.* The QoS parameters can be either positive or negative. Hence, they are scaled differently. Negative parameters are scaled using the following equation:

$$\beta_{i,j} = \begin{cases} \frac{\alpha_i^{\max} - \gamma_{i,j}}{\alpha_i^{\max} - \alpha_i^{\min}}, & \text{if } \alpha_i^{\max} \neq \alpha_i^{\min}, \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

   Positive parameters are scaled using the following equation:

$$\beta_{i,j} = \begin{cases} \frac{\gamma_{i,j} - \alpha_i^{\min}}{\alpha_i^{\max} - \alpha_i^{\min}}, & \text{if } \alpha_i^{\max} \neq \alpha_i^{\min}, \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

2. *Weighting.* In this stage, the final score q_i is computed using the formula $score = \sum_{j=1}^{k} \beta_{i,j} \cdot w_k$, where $w_k \in [0, 1]$ and $\sum_{j=1}^{k} w_j = 1$.

   Let s_f be the scaling factor and the final $q_i = score \times s_f$.

TABLE 2
QoS Parameters and Weights

QoS Parameter	Weight
Bandwidth	0.2
Latency	0.3
Turnaround time	0.5

In this way, QoS parameters are boiled down to a number. This number is used along with the cost in the procurement auction.

We illustrate QoS scaling with a numerical example. In our example, we consider a few common QoS parameters and assign scores arbitrarily for illustration. Table 1 shows some QoS parameters and their associated scores. Table 2 shows the weights associated with these QoS parameters.

Suppose a cloud vendor responds with specifications: Latency 30 ms, Turnaround time = 6 hours, and 200-kbps bandwidth. Let $s_f = 10$. Therefore, the final score may be computed as $v_{11} = 0.5, v_{12} = 1$, and $v_{13} = 1$. $score = (0.5 \cdot 0.3 + 1 \cdot 0.5 + 1 \cdot 0.2) \cdot 10 = 8.5$.

3.2 Cloud Resource Procurement Model

Cloud vendors are represented by $N = \{1, 2, \ldots, n\}$. In this procurement auction, each cloud vendor responds by bidding with total cost c_i and promised QoS parameters. These parameters are converted into numbers q_i using the technique presented in the previous section. Hence, the bid is an ordered pair (c_i, q_i).

Each cloud vendor $i \in N$ has execution cost c_i, where $0 < c_i$ and QoS q_i, where $0 < q_i$. Let $\underline{c}$ be the lowest cost valuation and $\bar{c}$ the highest cost valuation. The cloud vendor's cost is always in the interval $[\underline{c}, \bar{c}]$, i.e., $\underline{c} \leq c_i \leq \bar{c}$. Similarly, let $\underline{q}$ be the lowest QoS value and $\bar{q}$ the highest QoS value. The cloud vendor's QoS is always in the interval $[\underline{q}, \bar{q}]$, i.e., $\underline{q} \leq q_i \leq \bar{q}$. This information is private to the cloud vendor.

Let Θ_i be the set of all possible true types of the cloud vendor and $\Theta_i = [\underline{c}, \bar{c}] \times [\underline{q}, \bar{q}]$. Let $\Theta = \Theta_1 \times \Theta_2 \times \cdots \times \Theta_n$.

We assume that cost and QoS are correlated. Hence, there is a joint distribution function of cost and QoS represented by Φ. We assume that the joint distribution function Φ is the same for all n cloud vendors, i.e., $\Phi_1 = \Phi_2 = \cdots = \Phi_n$. Hence, all cloud vendors are symmetric. This assumption does not hold in case of an optimal mechanism like C-OPT, where different vendors have different joint distributions, as in Section 4.3.

In this mechanism, the *true type* [20] of each cloud vendor is represented by $b_i = (c_i, q_i)$, and the reported bid is represented by $\hat{b}_i = (\hat{c}_i, \hat{q}_i)$. Let $b = (b_1, b_2, \ldots, b_n)$ be the tuple of bids of all agents, called the *type profile*. Let b_{-i} be the cloud vendor tuple without cloud vendor i, i.e., $b_{-i} = (b_1, b_2, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$. Also $b = (b_{-i}, b_i)$. The goal of a mechanism is to design the following functions:

- Allocation function $g : \Theta \rightarrow \{0, 1\}$ specifies the winner.
- Payment function $h : \Theta \rightarrow \mathbb{R}$ determines the amount paid by the user to the cloud vendor.

TABLE 3
Table of Major Notation

Symbol	Description
n	Number of cloud vendors
N	A set of cloud vendors, $\{1, 2, ..., n\}$
X	Outcome set
θ_i	Preference or type of cloud vendor i
b_i	Cloud vendor i's true bid
$\hat{b_i}$	Cloud vendor i's reported bid
c_i	True cost of the resource provided by i
$\hat{c_i}$	Reported cost of the resource provided by i
q_i	True QoS supported by i
$\hat{q_i}$	Reported QoS supported by i
Θ_i	Set of types of cloud vendor i
u_i	Utility function of cloud vendor i
Φ_i	Joint distribution of cloud vendor i
v_i	Valuation of cloud vendor i
b	Bid vector $b = (b_1, b_1, \cdots, b_n)$
x	outcome and $x \in X$
$f(b)$	Social choice function
g	Allocation function
h	Payment function
h_i	Payment received by i
ξ_i	Expected social welfare of agent i
T_i	Expected payment of cloud vendor i
ρ_i	Offered expected surplus of cloud vendor i
π_i	Expected surplus of cloud vendor i

The goal of the cloud user is to minimize the procurement cost. This can be achieved only if all the cloud vendors quote true costs (bid truthfully). This is also called *truth elicitation*, which can be done in two ways [19]:

- *Dominant strategy incentive compatibility.* This corresponds to dominant strategy equilibrium.
- *Bayesian incentive compatibility.* This corresponds to Bayesian Nash equilibrium.

Major notations are summarized in Table 3.

4 PROPOSED MECHANISMS

4.1 Cloud-Dominant Strategy Incentive Compatible Mechanism

Dominant strategy incentive compatibility is one of the methods for truth elicitation. In this, truth telling is the best response of the participants, irrespective of other participants' strategies [20]. Let $f(\hat{b}) = (g(\hat{b}), h_1(\hat{b}), \ldots, h_n(\hat{b}))$ be the social choice function in the C-DSIC mechanism which implements dominant strategy incentive compatibility. $g(\hat{b})$ is the allocation rule that represents the winner. $h_i(\hat{b})$ is the payment received by the cloud vendor i. If $h_i(\hat{b}) > 0$, then payment is received by the cloud vendor i; otherwise, i pays money to the user.

We design a dominant strategy incentive compatible mechanism. According to the Gibbard-Satterthwaite impossibility theorem [57], [27], only dictatorial social choice functions are DSIC. Hence, in a general environment, DSIC cannot be implemented. But social choice functions are nondictatorial in a quasilinear environment [20]. The VCG mechanism [20] is DSIC in a quasilinear environment. We assume that cloud vendor's aim is to maximize profit. Hence, cloud vendors are *risk neutral*, and this implies quasilinearity [21, p. 269]. A similar assumption is also made by others in case of Grid [19].

The allocation rule is given by

$$g_i(\hat{b}) = \begin{cases} 1, & \text{if } \frac{\hat{c}_i}{\hat{q}_i} = \min\left(\frac{\hat{c}_1}{\hat{q}_1}, \frac{\hat{c}_2}{\hat{q}_2}, \ldots, \frac{\hat{c}_n}{\hat{q}_n}\right), \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

In the above allocation rule, the cloud vendor whose ratio of cost over QoS is minimum is the winner. The payment function—which is also called the *pivot rule* or Clarke's mechanism [19]—is given as follows:

$$h_i(\hat{b}) = g_i(\hat{b})\hat{c}_i + \sum_{j \neq i} \hat{c}_j g_j^{-i}(\hat{b}) - \sum_{j \neq i} \hat{c}_j g_j(\hat{b}). \quad (4)$$

The payment received by cloud vendor i is the sum of quoted cost and the difference between optimal cost without cloud vendor i and optimal cost with cloud vendor i.

Assume i is the winner and let k be the winner in the absence of i. The payment received by i according to (4) is $h_i(\hat{b}) = \hat{c}_i + \hat{c}_k - \hat{c}_i = \hat{c}_k$.

Let us assume there is a cloud vendor $l \in N$ and l is not a winner. Let m be the winner. The payment received by l is $h_l(\hat{b}) = \hat{c}_l \times 0 + \hat{c}_m - \hat{c}_m = 0$. Hence, in the above mechanism, the winner quotes the lowest cost per QoS but receives second lowest amount as payment. Other participants do not receive any payment. Therefore, this is a low-bid Vickrey auction [58].

The C-DSIC is presented in Algorithm 1.

Algorithm 1: C-DSIC

```
Input  : Set of bids $\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_n$
Output: Winner and payments for participants
        $(h_1, h_2, \cdots, h_n)$
1  $min \leftarrow \infty$;
2  $winner \leftarrow 0$;
3  for $i \leftarrow 1$ to $n$ do
4      if $(\frac{\hat{c}_i}{\hat{q}_i}) < min$ then $min \leftarrow \frac{\hat{c}_i}{\hat{q}_i}$;
5      $winner \leftarrow i$;
6  end
7  for $i \leftarrow 1$ to $n$ do
8      // The payment for each cloud vendor
9      // $i$ as per (4)
10     $h_i(\hat{b}) \leftarrow g_i(\hat{b})\hat{c}_i + \sum_{j \neq i} \hat{c}_j g_j^{-i}(\hat{b}) - \sum_{j \neq i} \hat{c}_j g_j(\hat{b})$;
11 end
```

The time complexity of this algorithm is $\mathcal{O}(n)$. In the first *for* loop, the minimum cost per QoS is calculated. The participant with lowest cost over QoS is the winner. The payment function h_i is computed based on the VCG mechanism according to (4). The C-DSIC algorithm is a low-bid Vickrey auction, and hence, only the winner gets a payment. The other participants do not receive any remuneration.

The properties satisfied by C-DSIC are:

- *Dominant strategy incentive compatibility.* The C-DSIC mechanism is based on VCG mechanism. VCG is DSIC and, hence, the proposed mechanism is DSIC.
- *Individual rationality.* The payments received by the cloud vendors are greater than or equal to zero. In

this mechanism, cloud vendors never pay the user and have a nonnegative payoff.

- *Allocative efficiency.* The winner is the cloud vendor with lowest cost over QoS. Hence, C-DSIC is allocative efficient.

4.2 Cloud-Bayesian Incentive Compatible Mechanism

The VCG mechanism is not budget balanced [20]. Hence, the C-DSIC mechanism is not budget balanced. If the mechanism is not budget balanced, then an external agent should pay money to perform auctions. C-BIC is proposed to overcome this limitation.

Nondictatorial social choice functions can be implemented in either of two ways: we can restrict the environment to be quasilinear, or we can implement Bayesian incentive compatibility which is weaker than VCG. The dAGVA mechanism [59] is Bayesian incentive compatible. In the dAGVA mechanism, each agent contributes money, and a payment is made to the agents using the contributed money [21, p. 289].

Let $f(\hat{b}) = (g(\hat{b}), h_1(\hat{b}), \dots, h_n(\hat{b}))$ be the social choice function in the C-BIC mechanism. Let $g_i(\hat{b})$ be the allocation rule and $h_i(\hat{b})$ be the payment received by cloud vendor i. If $h_i(\hat{b}) > 0$, then payment is received by i; otherwise, i pays the cloud user.

We design a Bayesian incentive compatible (BIC) mechanism. The allocation rule is the same as C-DSIC but the payment function is different from C-DSIC. The allocation rule is given as follows:

$$g_i(\hat{b}) = \begin{cases} 1, & \text{if } \frac{\hat{c}_i}{\hat{q}_i} = \min\left(\frac{\hat{c}_1}{\hat{q}_1}, \frac{\hat{c}_2}{\hat{q}_2}, \dots, \frac{\hat{c}_n}{\hat{q}_n}\right), \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The payment rule is based on the dAGVA mechanism. Each cloud vendor contributes money and this contributed money is used for paying all the cloud vendors. In other words, cloud vendors pay a participation fee. Let ξ_i be the expected social welfare [21] of agent i and is calculated using the following equation:

$$\xi_i(\hat{b}_i) = \mathbb{E}_{\hat{b}_{-i}}\left[\sum_{i \neq j} c_j(g_j(\hat{b}_i, b_j))\right]. \tag{6}$$

The payment rule [59] is given by

$$h_i(\hat{b}_i) = \xi_i(\hat{b}_i) - \left(\frac{1}{n-1}\sum_{j \neq i} \xi_j(\hat{b}_j)\right). \tag{7}$$

Each cloud vendor i receives payment $\xi(\hat{b}_i)$. Now, each cloud vendor contributes an equal $\frac{1}{n-1}$ of its share. Hence, cloud vendor i's net transfer is $\xi_i(\hat{b}_i) - (\frac{1}{n-1})\sum_{j \neq i} \xi_j(\hat{b}_i)$.

Since cloud vendors themselves pay money for the payment, this mechanism is budget balanced. The proof of budget balance in the dAGVA mechanism can be found in [19].

The C-BIC is presented in Algorithm 2 below.

Algorithm 2: C-BIC

```
Input  : Set of bids $\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_n$
Output: Winner and payments for participants
        $(h_1, h_2, \cdots, h_n)$
1  $min \leftarrow \infty$;
2  $winner \leftarrow 0$;
3  for $i \leftarrow 1$ to $n$ do
4      if $(\frac{\hat{c}_i}{\hat{q}_i}) < min$ then $min \leftarrow \frac{\hat{c}_i}{\hat{q}_i}$;
5      $winner \leftarrow i$;
6  end
7  for $i \leftarrow 1$ to $n$ do
8      // Pay each cloud vendor i
9      // based on (6) and (7)
10     $\xi_i(\hat{b}_i) \leftarrow \mathbb{E}_{\hat{b}_{-i}}[\sum_{i \neq j} c_j(g_j(\hat{b}_i, b_j))]$;
11     $h_i(\hat{b}_i) \leftarrow \xi_i(\hat{b_j}) - \left(\frac{1}{n-1}\sum_{j \neq i} \xi_j(\hat{b}_j)\right)$;
12 end
```

The time complexity of this algorithm is $\mathcal{O}(n)$. In the first *for* loop, the minimum cost per QoS is calculated. The participant with lowest cost per QoS is the winner. The payment function h_i is computed based on the dAGVA mechanism. The C-DSIC algorithm is a low-bid Vickrey auction and hence only the winner gets paid. The other cloud vendors do not get any money.

In C-BIC, every cloud vendor contributes a participation fee, but only the winner gets paid. Hence, the procurement cost is less than C-DSIC. Therefore, the other cloud vendors suffer a loss. This loss is regarded as the participation fee. Since the allocation rules of C-DSIC and C-BIC are the same, C-BIC is also allocative efficient. The C-BIC mechanism cannot guarantee individual rationality. This is an important property—even though *ex ante* individual rationality is preserved (there is no loss to the cloud vendor if it withdraws from the auction before it submits a bid), *interim* individual rationality is not preserved. This implies that the cloud vendors suffer a loss if they withdraw from the auction after they submit bids.

4.3 Cloud-Optimal Mechanism

The C-DSIC mechanism is not budget balanced. On the other hand, even though the C-BSIC mechanism is budget balanced, it is not individually rational. Hence, we propose the C-OPT mechanism to address the limitations of both the C-DSIC and C-BIC mechanisms.

According to Myerson [22], if a mechanism is Bayesian incentive compatible and individually rational, then the mechanism is optimal. Myerson's optimal auction can be applied only to single items with unit demand. In our model, both cost and QoS are correlated. Hence, the design of an optimal auction is not trivial [19], [60].

Iyengar and Kumar [12] propose an optimal mechanism for procurement auctions for suppliers who have finite production capacity (capacitated suppliers). Practically, it is not possible for cloud service providers to guarantee infinite QoS for every cloud user. Hence, we assume that cloud vendors have finite QoS, i.e., $0 < q_i < \infty$.

Iyengar and Kumar's [12] is an important work with respect to building an optimal mechanism. They prove a set of theorems to prove a mechanism as optimal. We use these theorems and prove that our C-OPT mechanism is optimal.

Let X_i and T_i be the expected allocation and payment, respectively. Iyengar and Kumar [12] give the following definitions:

Definition 4.1. *The offered expected surplus for a procurement mechanism (g,h) is defined as* $\rho_i(\hat{c}_i, \hat{q}_i) = T_i(\hat{c}_i, \hat{q}_i) - \hat{c}_i X_i(\hat{c}_i, \hat{q}_i)$. *It is the expected transfer payment when the vendor i bids* $(\hat{c}_i, \hat{q}_i)$.

Definition 4.2. *The* expected surplus *of a vendor i when the bid is* $\hat{b}_i = (\hat{c}_i, \hat{q}_i)$ *is defined as* $\pi_i(\hat{c}_i, \hat{q}_i) = T_i(\hat{c}_i, \hat{q}_i) - c_i X_i(\hat{c}_i, \hat{q}_i)$. *Also,*

$$\pi_i(\hat{c}_i, \hat{q}_i) = \pi_i(\hat{c}_i, \hat{q}_i) + (c_i - \hat{c}_i) X_i(\hat{c}_i, \hat{q}_i)$$

In an incentive compatible mechanism, the true surplus π_i *is equal to offered surplus.*

In simple terms, *expected surplus* is the difference between what the cloud vendor is willing to accept and what it actually obtains. Myerson [22] defines a virtual parameter for ranking the buyers. This virtual parameter is modified [12], [19], [60] to capture the correlation between cost and quantity, and is called *virtual cost*.

Definition 4.3. *The* virtual cost *is defined as*

$$H_i(c_i, q_i) = c_i + \frac{F_i\left(\frac{c_i}{q_i}\right)}{f_i\left(\frac{c_i}{q_i}\right)}.$$

This virtual cost is similar to [12], [19], [60], except that we consider QoS instead of quantity.

To develop an optimal mechanism, we assume the following [12]:

- The joint distribution function $\Phi_i(c_i, q_i)$ is completely defined.
- The virtual cost function H_i is nondecreasing in both c_i and q_i.

Let $\pi(g, h)$ be the total expected profit of the user. The goal of an optimal mechanism is to maximize $\pi(g, h) = \mathbb{E}[\mathbb{R} - \sum_{i=1}^{n} h_i(b)]$ subject to

1. *individual rationality*. The expected interim surplus for each cloud vendor is nonnegative, i.e., $\pi_i(b_i) \geq 0$; and
2. *Bayesian incentive compatibility*. The truth elicitation should be weakly dominant strategy for all cloud vendors, i.e.,

$$\begin{aligned}\mathbb{E}_{b_{-i}}[h_i(b_i, b_{-i}) - c_i g_i(b_i, b_{-i})] \geq \mathbb{E}_{b_{-i}}[h_i(\hat{b}_i, b_{-i}) \\ - c_i g_i(\hat{b}_i, b_{-i})], \forall i \in N, \forall b_i, \hat{b}_i \in \Theta_i.\end{aligned}$$

By Myerson [22], a mechanism that satisfies the above constraints and maximizes cloud user profit is optimal.

The optimal auction presented in [22] assumes unit demand of the item and does not take QoS into account. In our model, the cloud user has QoS requirement and QoS plays an important role in the selection of cloud vendor. This multidimensional attribute of cloud vendors makes this a nontrivial problem. In particular, we cannot apply traditional optimizing schemes in this case [12], [19], [60].

The properties of optimal procurement mechanism [12] with capacitated suppliers are:

1. The expected allocation $X_i(c_i, q_i)$ is nonincreasing in the cost parameter $c_i \forall$ suppliers.
2. The offered surplus $\rho_i(\hat{c}_i, \hat{q}_i)$ is of the form $\rho_i(\hat{c}_i, \hat{q}_i) = \rho_i(\bar{c}, \hat{q}) + \int_{\hat{c}_i}^{\bar{c}} X_i(y, \hat{q}_i)$.

The proofs of the above properties can be found in [12].

We prove that the proposed C-OPT mechanism satisfies the above properties and hence is optimal.

Lemma 4.4. *The expected allocation* $X_i(c_i, q_i)$ *of the C-OPT mechanism is nonincreasing in the cost parameter* c_i *and the QoS parameter* q_i.

Proof. The expected allocation $X_i(c_i, q_i) = \int_{\underline{c}}^{\hat{c}_i} g_i(\hat{b}_i)$.

Assume q_i is fixed. The expected allocation X_i is 0 for the losers. Let c_{ia} and c_{ib} be the quoted cost of the cloud vendor c_i. Also, we assume i is the winner in both the cases and $c_{ia} < c_{ib}$.

The expected allocation X_i when the cost c_{ia} is

$$\begin{aligned} X_i &= \int_{\underline{c}}^{\hat{c_{ia}}} g_i(\hat{c}_i, \hat{q}_i) dc \\ &= \hat{q}_i(\hat{c_{ia}} - \underline{c}). \end{aligned} \tag{8}$$

Similarly, when the cost is c_{ib}, the expected allocation is

$$\begin{aligned} X_i &= \int_{\underline{c}}^{\hat{c_{ib}}} g_i(\hat{c}_i, \hat{q}_i) dc \\ &= \hat{q}_i(\hat{c_{ib}} - \underline{c}). \end{aligned} \tag{9}$$

As is clear from (8) and (9), $X_i(\hat{c_{ia}}, \hat{q}_i) < X_i(\hat{c_{ib}}, \hat{q}_i)$ when $c_{ia} < c_{ib}$.

Hence, X_i is nonincreasing in cost when QoS is fixed. Similarly, we can prove that X_i is nonincreasing in QoS when cost is fixed. □

The expected surplus of the winning vendor is called the *information rent* of the vendor [61]. Classically, surpluses like supplier surplus and consumer surplus are examples of information rent.

Lemma 4.5. *The offered surplus* $\rho_i(\hat{c}_i, \hat{q}_i)$ *in C-OPT mechanism is of the form* $\rho_i(\hat{c}_i, \hat{q}_i) = \rho_i(\bar{c}, \hat{q}) + \int_{\hat{c}_i}^{\bar{c}} X_i(y, \hat{q}_i)$

Proof. The payment function in C-OPT is given by $h_i(\hat{b}_i) = c_i g_i(\hat{b}) + \int_{c_i}^{\bar{c}} X_i(y, \hat{q}_i) dy$, where $\int_{c_i}^{\bar{c}} X_i(y, \hat{q}_i) dy$ is the offered surplus to the cloud vendors.

In this case, the information rent of the cloud vendor i is $\rho_i(\bar{c}, \hat{q}) = 0$. When $\hat{c}_i = \bar{c}$, then $\rho_i = 0$. Hence, the information rent paid for the cloud vendor with highest cost valuation is 0. Hence, the offered surplus is in the form of $\rho_i(\hat{c}_i, \hat{q}_i) = \rho_i(\bar{c}, \hat{q}) + \int_{\hat{c}_i}^{\bar{c}} X_i(y, \hat{q}_i)$ □

The allocation rule is given by

$$g_i(\hat{b}) = \begin{cases} 1, & \text{if } H_i = \min(H_1, H_2, \ldots, H_n,) \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

By (10), the cloud vendor i whose virtual cost H is minimum is declared the winner.

The payment rule is given by

$$h_i(\hat{b_i}) = c_i g_i(\hat{b}) + \int_{c_i}^{\bar{c}} X_i(y, \hat{q}_i)dy. \quad (11)$$

The C-OPT is presented in Algorithm 3.

Algorithm 3: C-OPT

```
Input  : Set of bids $\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_n$
Output: Winner and payments for participants
        $(h_1, h_2, \cdots, h_n)$
1  $min \leftarrow \infty$;
2  $winner \leftarrow 0$;
3  for $i \leftarrow 1$ to $n$ do
4      Compute $H_i$;
5      if $(H_i < min)$ then  $min \leftarrow H_i$;
6      $winner \leftarrow i$;
7  end
8  for $i \leftarrow 1$ to $n$ do
9      // Pay each cloud vendor i
10     // based on (11)
11     $h_i(\hat{b_i}) \leftarrow c_i g_i(\hat{b}) + \int_{c_i}^{\bar{c}} X_i(y, \hat{q}_i)dy$
12 end
```

Theorem 4.6. *The C-OPT mechanism with allocation rule (10) and payment rule (11) is* Bayesian incentive compatible, individually rational *and* revenue maximizing.

Proof. According to Lemma 4.4, $X_i(c_i, q_i)$ is nonincreasing in c_i and q_i. Also, the offered surplus ρ_i is in the form specified by Lemma 4.5. The offered surplus $\rho_i = \int_{c_i}^{\bar{c}} X_i(y, \hat{q}_i)dy$.

If $\pi_i(c_i, q_i)$ is nondecreasing in q_i, $\forall c_i$, then the mechanism is incentive compatible [12]. It can easily be seen from the offered surplus ρ_i that $\pi_i(c_i, q_i)$ is nondecreasing in q_i, $\forall c_i$.

Also, the $\rho_i \geq 0$, $\forall i$. Hence, C-OPT is individually rational.

Since the payment rule is of form $h_i(\hat{b_i}) = c_i g_i(\hat{b}) + \int_{c_i}^{\bar{c}} X_i(y, \hat{q}_i)dy$, the mechanism is revenue maximizing [12]. □

C-OPT is an optimal mechanism and is more general compared to both C-DSIC and C-BIC. We assume that cloud vendors are symmetric in C-DSIC and C-BIC. But in realistic scenarios, different cloud vendors may have different price distributions. On the other hand, C-OPT can be applied when $\Phi_1 \neq \Phi_2 \neq \cdots \neq \Phi_n$. By the discussion of Narahari *et al.* [19], C-OPT reduces to C-DSIC under the following conditions:

- Cloud vendors are symmetric.
- The joint distribution function Φ is regular.

C-DSIC is prone to bidder collusion and is not budget balanced. In C-BIC, losing cloud vendors lose their money. In C-OPT, the cloud vendor can neither overbid nor underbid. If the cloud vendor overbids, then incentive is not paid. On the other hand, if it underbids, then it will not be the winner. Hence, C-OPT is suitable in a larger set of real-world contexts than C-DSIC and C-BIC.

The mechanisms presented in this paper have linear time complexity. Hence, they are appropriate for implementing procurement auctions.

5 Experimental Results

Currently, the selection of a cloud vendor is manual. The selection of a cloud vendor with low cost is also popularly called the *first price auction*. Similarly, a user can perform a Vickrey auction and pay the second-lowest cost to the winner [42].

In the real world, cloud vendors follow different price distributions. In this kind of scenario, the winner determination and procurement cost computation using first-price and Vickrey auctions is not optimal [20]. Hence, this approach should not be followed in the real world.

It may be noted that if we do not use mechanism design, we would need to use standard auctions like first bid, second bid, and so on. However, we cannot enforce truthfulness simply using auctions. Truthfulness cannot be measured. Hence, there is no other baseline to compare our models.

Narahari *et al.* [19] design procurement mechanisms that can be used by grid users to procure resources in a computational Grid with rational resource providers. They simulate the proposed mechanisms and compare the procurement costs of the mechanisms. The following approach is adopted by them to evaluate their proposed mechanisms.

- The mechanisms proposed are decentralized in nature. To determine the lower bound on the procurement cost, they use a naïve centralized algorithm. This centralized algorithm sorts the bids in the ascending order and allocate jobs according to the order. This algorithm assumes that resource providers are nonstrategic.
- They do not use a standard grid toolkit because their goal is to evaluate mechanisms and not to simulate low-level grid tasks. They build a customized simulation environment with an appropriate level of abstraction.
- Costs and tasks are uniformly distributed. The average procurement cost is calculated in every mechanism and compared.

We follow a similar methodology to simulate C-DSIC, C-BIC, and C-OPT. Our simulation approach is as follows:

1. The main challenge we faced is the simulation of cloud resource prices. Currently, different cloud vendors use different prices depending on the resource. Amazon EC2 [62] is a popular web service that provides resizable compute capacity in the cloud. We performed distribution fitting of the Amazon EC2 prices to find the probability distribution of the price.
2. The EC2 yearly contract price is in the interval of [57,8000] and is lognormally distributed. In our simulation, we generate costs in this interval, distributed lognormally as earlier noted.
3. QoS is an emerging topic in cloud computing. There is yet a lack of standard work about the QoS and its properties in the context of cloud computing. Hence, we perform simulations with different possible distributions of QoS. In Scenario 1, QoS is uniformly distributed in the interval $[1, 10]$, and in Scenario 2, it is normally distributed with mean $(\mu = 5.0)$ and variance $\sigma^2 = 5.0$. The mean and variance values are chosen to facilitate QoS rating on the scale of 10.

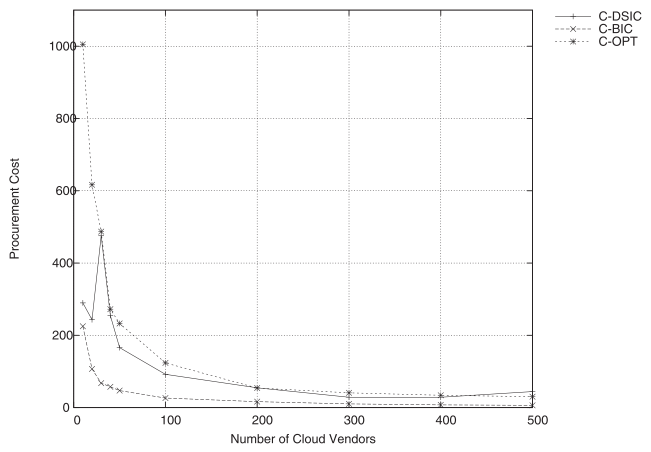

Fig. 1. Comparison of procurement costs of C-DSIC, CBIC, and C-OPT in Scenario 1.

4. In our simulation, we do not take the cloud resource type (SaaS, IaaS, etc.) into account, since our aim is to evaluate the proposed mechanism.
5. Currently, there is a lack of any standard toolkit for evaluating mechanisms in cloud. CloudSim is a popular simulation software for cloud applications; however, it supports neither auction protocols nor price generation [24]. Hence, Belalem *et al.* [24] use a custom, fixed formula to generate prices, which is not possible for us as price and QoS generation are quite essential to validate our proposed mechanisms. Similarly, other popular cloud environments like Eucalyptus, and so on, do not provide for price simulation. Hence, we implemented our simulation using Java based on the equations presented in this work, without compromising on cloud properties.

In Fig. 1, the x-axis scale is with one unit length representing 100 cloud vendors. The minimum number of cloud vendors is taken to be 10. Similarly, also is the case with Fig. 2.

Table 4 shows the procurement cost to the user in C-DSIC, C-BIC, and C-OPT for different number of cloud vendors in the Scenario 1. In this scenario, QoS is uniformly distributed. Table 5 shows the procurement cost in Scenario 2, where QoS is normally distributed.

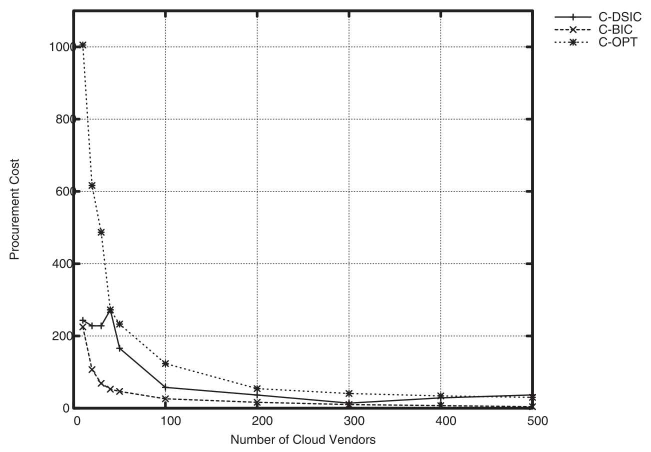

Fig. 2. Comparison of procurement costs of C-DSIC, CBIC, and C-OPT in Scenario 2.

TABLE 4
Procurement Costs in Scenario 1

	Procurement Cost ($)		
Cloud vendors	C-DSIC	C-BIC	C-OPT
10	289.62	225.12	1005.16
20	243.32	107.02	616.56
30	475.42	67.65	487.03
40	254.71	57.44	272.45
50	165.87	46.97	232.82
100	91.95	26.27	123.71
200	54.79	16.36	54.16
300	28.65	10.29	40.85
400	28.67	7.35	34.19
500	44.16	6.17	30.2

Figs. 1 and 2 show the graphs of procurement costs in C-DSIC, C-BIC, and C-OPT for different numbers of cloud vendors in Scenarios 1 and 2, respectively.

Fig. 2 shows the graph of procurement costs in C-DSIC, C-BIC, and C-OPT for different number of cloud vendors in Scenario 2.

The main observations from these experiments are:

1. The payment made by the cloud user decreases with the increase of number of cloud vendors, irrespective of the mechanism implemented.
2. The behavior of procurement costs is similar in both Figs. 1 and 2.
3. The payment in C-BIC decreases more rapidly compared to other methods with an increase in the number of cloud vendors, in both the scenarios. As the number of cloud vendors increases, the additional surplus generated also increases. Hence, the user has to pay less money to the winner. Therefore, C-BIC outperforms both C-DSIC and C-OPT.
4. In Table 4, the C-DSIC procurement cost is greater than C-OPT when the number of cloud vendors is 500. In C-DSIC, the marginal contribution is the difference between the lowest and second lowest costs. If the marginal contribution is low, then the procurement cost becomes high. In C-OPT, incentive is calculated as the difference between quoted cost and highest valuation. Hence, the incentive decreases as the quoted cost approaches highest cost valuation. Therefore, the procurement cost in C-DSIC

TABLE 5
Procurement Costs in Scenario 2

	Procurement Cost ($)		
Cloud vendors	C-DSIC	C-BIC	C-OPT
10	243.32	225.12	1005.17
20	227.96	107.03	616.57
30	227.96	68.86	487.03
40	272.39	52.82	272.45
50	165.87	46.47	232.82
100	57.59	26.16	123.71
200	36.51	16.36	54.16
300	14.23	10.24	40.85
400	28.65	7.3	34.19
500	36.77	4.16	30.12

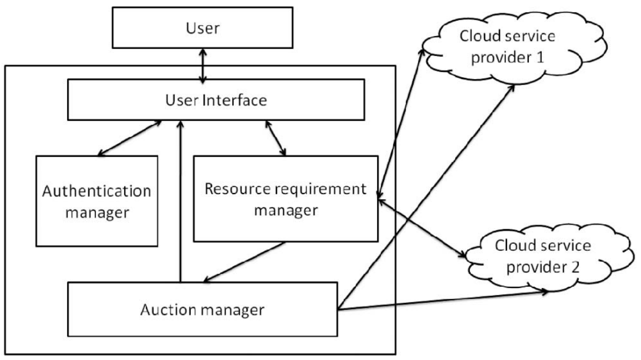

Fig. 3. Cloud broker procurement module architecture.

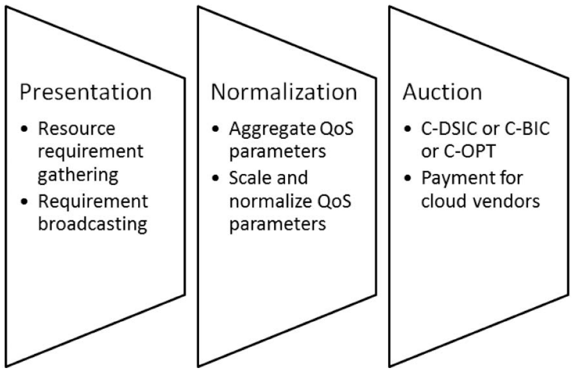
Presentation
• Resource requirement gathering
• Requirement broadcasting
Normalization
• Aggregate QoS parameters
• Scale and normalize QoS parameters
Auction
• C-DSIC or C-BIC or C-OPT
• Payment for cloud vendors

Fig. 4. Procurement module phases.

is greater than C-OPT. This is the case where marginal contribution is less but the quoted cost nears the highest cost valuation.

5. In Table 5, the C-DSIC procurement cost is less than that for C-OPT. Also, the C-OPT procurement cost steadily decreases in both scenarios when cloud vendors increase in number (see Tables 4 and 5).
6. In most cases, the procurement cost of C-OPT is slightly greater than C-DSIC (see Table 4) because in our setting, the cost valuation is high (8,000). When we reduce the highest cost valuation, the C-OPT procurement cost is less than with C-DSIC. Hence, the C-OPT procurement cost depends on the interval of the cost.

6 Cloud Broker Procurement Module

The cloud broker is responsible for management of a cloud environment. Grivas *et al.* [2] extend the functionality of a cloud broker. In their model, a cloud broker automatically reconfigures the cloud environment based on changes in the user's business process.

Consider a scenario where an organization wants to use an application hosted on a cloud. There are a lot of cloud vendors like Google, Yahoo, and so on. The specifications of these cloud vendors are not uniform and comparing these specifications manually is very tedious. So it is very challenging for an organization to select an appropriate cloud vendor. This manual selection is impractical when there are large number of cloud vendors. Hence, the selection of cloud vendors must be automated. Fig. 3 shows the architecture diagram of the procurement module of a cloud broker.

The main components of the procurement module are:

- *User interface*: This component is responsible for interacting with the user. The cloud user indicates his resource requirements through this component.
- *Authentication manager*: This component is responsible for authenticating users and cloud vendors.
- *Resource requirement manager*: This component validates the user resource requirements. The validated requirements are broadcasted to all the cloud vendors. The cloud vendors respond with the assumed QoS parameters and cost. This information is validated and sent to the auction manager.
- *Auction manager*: This component is the heart of the procurement module. The different QoS parameters received are normalized to enable QoS comparisons. C-DSIC, C-BIC, or C-OPT mechanism presented in this work is used to perform procurement auction. The winner is determined and the payment is calculated based on the mechanism. Finally, both the user and winner are notified.

Fig. 4 represents the proposed procurement module phases.

The procurement module performs three functions:

1. *Presentation*. Resource requirements are gathered from the user and broadcast to all cloud vendors. The user specifies the requirements necessary to complete a job. Some of the examples of the requirements are: SLA for the job is 1 day, and so on. The cloud vendor goes through the requirements and responds with cost and QoS parameters.
2. *Normalization*. Cloud vendors provide different QoS. To compare them, it is necessary to normalize them. We use the method presented in Section 3.1. If the QoS of the cloud vendor is less than a specified minimum, then the bid is excluded from the auction.
3. *Auction*. In this phase, the mechanisms presented in this work are used for performing procurement auction. The user can configure this module to use C-DSIC, C-BIC, or C-OPT based on its preference. The payments are computed according to the mechanism implemented.

Fig. 5 represents the cloud broker activity.

The cloud user sends required resource specifications to the cloud broker. Cloud broker in turn broadcasts the user specifications to all the cloud vendors. Cloud vendors respond with their cost and assumed QoS. The different QoS parameters are normalized by the broker. This normalization is important because it enables the comparison of different QoS specifications. Cost and assumed QoS form a bid vector in this scenario. The cloud broker collects these bid vectors and implements C-DSIC or C-BIC or C-OPT mechanism. Cloud vendor with lowest ratio of cost over QoS is declared the winner. In Fig. 5, vendor i is the winner. Cloud broker notifies both the winner and the user. Finally, cloud broker pays all the cloud vendors based on implemented mechanism (C-DSIC, C-BIC, or C-OPT).

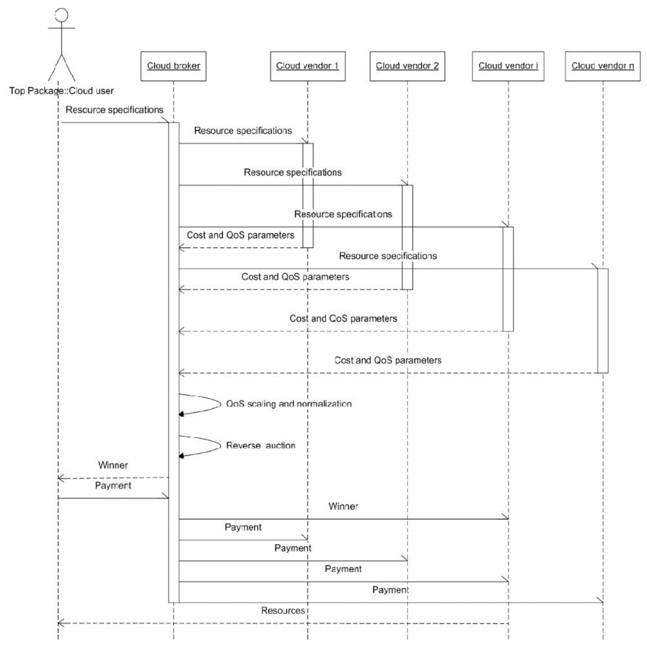

Fig. 5. Cloud broker interactions.

7 Conclusion and Future Work

Currently, the cloud user pays a fixed price for resources or services. This type of pricing is called fixed pricing. Fixed pricing is very popular with telecom providers. On the flip side, there is no provision for incentives for users in the fixed strategy. Resource procurement is not only an important problem in cloud computing but is also an unexplored area. Currently, resource procurement is done manually and there is a pressing need to automate it.

To automate procurement, we have presented three mechanisms: C-DSIC, C-BIC, and C-OPT. C-DSIC is a low-bid Vickrey auction. It is allocative efficient and individual rational but not budget balanced. If the mechanism is not budget balanced, then an external agency has to provide money to perform procurement.

C-BIC is a weaker strategy compared to C-DSIC and it is Bayesian incentive compatible. In C-BIC, vendorss reveal the truth only if other vendors reveal the truth, unlike C-DISC where vendors reveal the truth irrespective of others' choices. C-BIC achieves budget balance and allocative efficiency but not individual rationality.

C-OPT achieves both Bayesian incentive compatibility and individual rationality, which the other two mechanisms cannot achieve. This mechanism is immune to both overbidding and underbidding. If a cloud vendor overbids, then the incentive is reduced. If it underbids, then it may not be a winner. C-OPT is more general compared to both C-DSIC and C-BIC—even if cloud vendors use different distributions for cost and QoS, we can safely use C-OPT. Hence, C-OPT is the preferred mechanism in more cases in the real world.

The experiments reveal an interesting pattern. The resource procurement cost reduces as the number of cloud vendors increase, irrespective of the mechanism implemented. The cost in C-BIC reduces more significantly, compared to the other two mechanisms.

The procurement module for a cloud broker based on C-DSIC, C-BIC, or C-OPT is able to automate the selection of cloud vendors. The mechanisms presented assume that cloud vendors are rational and intelligent, which is true in the real-world scenario. This work enables the user to select the appropriate cloud vendor, and the mechanism chosen also decides the price for the resource. This user-centric pricing is a step toward implementing dynamic pricing in the cloud.

References

[1] P. Mell and T. Grance, *The NIST Definition of Cloud Computing,* NIST Special Publication 800-145, Nat'l Inst. of Standards and Technology, US Dept. of Commerce, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, Sept. 2011.

[2] S. Grivas, T.U. Kumar, and H. Wache, "Cloud Broker: Bringing Intelligence into the Cloud," *Proc. IEEE Third Int'l Conf. Cloud Computing (CLOUD),* pp. 544-545, July 2010.

[3] M.F. Mithani, M. Salsburg, and S. Rao, "A Decision Support System for Moving Workloads to Public Clouds," *GSTF Int'l J. Computing,* vol. 1, no. 1, pp. 150-157, Aug. 2010, doi:10.5176_2010-2283_1.1.25.

[4] B. Rochwerger, J. Tordsson, C. Ragusa, D. Breitgand, S. Clayman, A. Epstein, D. Hadas, E. Levy, I. Loy, A. Maraschini, P. Massonet, H. Muñoz, K. Nagin, G. Toffetti, and M. Villari, "RESERVOIR—-When One Cloud is Not Enough," *Computer,* vol. 44, no. 3, pp. 44-51, Mar. 2011.

[5] Y. Yang, Y. Zhou, L. Liang, D. He, and Z. Sun, "A Sevice-Oriented Broker for Bulk Data Transfer in Cloud Computing," *Proc. Ninth Int'l Conf. Grid and Cooperative Computing (GCC),* pp. 264-269, Nov. 2010.

[6] B.J. Lheureux and D.C. Plummer, "Cloud Services Brokerages: The Dawn of the Next Intermediation Age," Research Report G00208731, Gartner, http://www.gartner.com/DisplayDocument?doc_cd=208731, Nov. 2010.

[7] F. Ridder and A. Bona, "Four Risky Issues When Contracting for Cloud Services," Research Report G00210385, Gartner, http://bit.ly/S6L4Zx, Feb. 2011.

[8] R. Weiss and A. Mehrotra, "Online Dynamic Pricing: Efficiency, Equity and the Future of E-Commerce," *Virginia J. Law and Technology,* vol. 6, no. 2, 2001.

[9] M. Bichler, J. Kalagnanam, K. Katircioglu, A.J. King, R.D. Lawrence, H.S. Lee, G.Y. Lin, and Y. Lu, "Applications of Flexible Pricing in Business-to-Business Electronic Commerce," *IBM Systems J.,* vol. 41, no. 2, pp. 287-302, 2002.

[10] Y. Narahari, C. Raju, K. Ravikumar, and S. Shah, "Dynamic Pricing Models for Electronic Business," *Sadhana,* vol. 30, pp. 231-256, 2005.

[11] S. Parsons, J.A. Rodriguez-Aguilar, and M. Klein, "Auctions and Bidding: A Guide for Computer Scientists," *ACM Computing Surveys,* vol. 43, no. 2, article 10, Jan. 2011, doi:10.1145/1883612.1883617.

[12] G. Iyengar and A. Kumar, "Optimal Procurement Mechanisms for Divisible Goods with Capacitated Suppliers," *Rev. Economic Design,* vol. 12, no. 2, pp. 129-154, June 2008.

[13] I. Foster, C. Kesselman, C. Lee, B. Lindell, K. Nahrstedt, and A. Roy, "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation," *Proc. Int'l Workshop Quality of Service,* pp. 27-36, 1999.

[14] H. Casanova and J. Dongarra, "NetSolve: A Network Server for Solving Computational Science Problems," *The Int'l J. Supercomputer Applications and High Performance Computing,* vol. 11, pp. 212-223, 1995.

[15] S.J. Chapin, D. Katramatos, J.F. Karpovich, and A.S. Grimshaw, "The Legion Resource Management System," *Proc. Job Scheduling Strategies for Parallel Processing,* pp. 162-178, 1999.

[16] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic Models for Resource Management and Scheduling in Grid Computing," *Concurrency and Computation: Practice and Experience,* vol. 14, nos. 13-15, pp. 1507-1542, 2002.

[17] M. Mihailescu and Y.M. Teo, "Dynamic Resource Pricing on Federated Clouds," *Proc. IEEE/ACM 10th Int'l Conf. Cluster, Cloud and Grid Computing (CCGRID '10),* pp. 513-517, 2010.

[18] B. Kalyanasundaram, M. Velauthapillai, and J. Waclawsky, "Unlocking the Advantages of Dynamic Service Selection and Pricing," *Theory Computing System*, vol. 38, pp. 393-410, July 2005.

[19] Y. Narahari, D. Garg, R. Narayanam, and H. Prakash, *Game Theoretic Problems in Network Economics and Mechanism Design Solutions.* Springer, 2009.

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

[21] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge Univ. Press, Dec. 2008.

[22] R.B. Myerson, "Optimal Auction Design," *Math. Operations Research*, vol. 6, no. 1, pp. 58-73, 1981.

[23] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software—Practice and Experience*, vol. 41, no. 1, pp. 23-50, Jan. 2011.

[24] G. Belalem, S. Bouamama, and L. Sekhri, "An Effective Economic Management of Resources in Cloud Computing," *J. Computers*, vol. 6, pp. 404-411, 2011.

[25] M. Sharpe, "Lognormal Model for Stock Prices," http://math.ucsd.edu/~msharpe/stockgrowth.pdf, 2004.

[26] E. Limpert, W.A. Stahl, and M. Abbt, "Log-Normal Distributions across the Sciences: Keys and Clues," *BioScience*, vol. 51, no. 5, pp. 341-352, May 2001.

[27] A. Gibbard, "Manipulation of Voting Schemes: A General Result," *Econometrica*, vol. 41, no. 4, pp. 587-601, 1973, doi:10.2307/1914083.

[28] P. Ghosh, N. Roy, S.K. Das, and K. Basu, "A Pricing Strategy for Job Allocation in Mobile Grids Using a Non-Cooperative Bargaining Theory Framework," *J. Parallel and Distributed Computing*, vol. 65, no. 11, pp. 1366-1383, 2005.

[29] S. Penmatsa and A. Chronopoulos, "Price-Based User-Optimal Job Allocation Scheme for Grid Systems," *Proc. Int'l Symp. Parallel and Distributed Processing*, p. 396, Apr. 2006.

[30] X. Xie, J. Huang, H. Jin, S. Wu, M. Koh, J. Song, and S. See, "Pricing Strategies in Grid Market: Simulation and Analysis," *Proc. Seventh Int'l Conf. Grid and Cooperative Computing*, pp. 532-538, Oct. 2008.

[31] X. Vilajosana, R. Krishnaswamy, and J. Marques, "Design of a Configurable Auction Server for Resource Allocation in Grid," *Proc. Int'l Conf. Complex, Intelligent and Software Intensive Systems (CISIS '09)*, pp. 396-401, Mar. 2009.

[32] X.-R. Cao, H.-X. Shen, R. Milito, and P. Wirth, "Internet Pricing with a Game Theoretical Approach: Concepts and Examples," *IEEE/ACM Trans. Networking*, vol. 10, no. 2, pp. 208-216, Apr. 2002.

[33] L.J.L. Mingbiao and X. Shengli, "Posted Price Model Based on GRS and Its Optimization Using in Grid Resource Allocation," *Proc. Int'l Conf. Wireless Comm., Networking and Mobile Computing (WiCom '07)*, pp. 3172-3175, Sept. 2007.

[34] K. Subramoniam, M. Maheswaran, and M. Toulouse, "Towards a Micro-Economic Model for Resource Allocation in Grid Computing Systems," *Proc. IEEE Canadian Conf. Electrical and Computer Eng. (CCECE '02)*, vol. 2, pp. 782-785, 2002.

[35] F. Xhafa and J. Kolodziej, "Game-Theoretic, Market and Meta-Heuristics Approaches for Modelling Scheduling and Resource Allocation in Grid Systems," *Proc. Int'l Conf. P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pp. 235-242, Nov. 2010.

[36] S. Parsa, A. Shokri, and S. Nourossana, "A Novel Market Based Grid Resource Allocation Algorithm," *Proc. First Int'l Conf. Networked Digital Technologies (NDT '09)*, pp. 146-152, July. 2009.

[37] L. Ismail, B. Mills, and A. Hennebelle, "A Formal Model of Dynamic Resource Allocation in Grid Computing Environment," *Proc. Ninth ACIS Int'l Conf. Software Eng., Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '08)*, pp. 685-693, Aug. 2008.

[38] W. Shu, "Optimal Resource Allocation on Grid Computing Using a Quantum Chromosomes Genetic Algorithm," *Proc. Second Workshop Digital Media and Its Application in Museum and Heritages*, pp. 303-306, Dec. 2007.

[39] Z.-J. Li, X.-D. Liu, X.-D. Duan, and C.-R. Wang, "Optimal Solution for Grid Resource Allocation Using Particle Swarm Optimization," *Proc. Third Int'l Conf. Multimedia and Ubiquitous Eng. (MUE '09)*, pp. 339-346, June. 2009.

[40] S. Shah, A. Mahmood, and A. Oxley, "Modified Least Cost Method for Grid Resource Allocation," *Proc. Int'l Conf. Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 218-225, Oct. 2010.

[41] F. Li and D. Qi, "Research on Grid Resource Allocation Algorithm Based on Fuzzy Clustering," *Proc. Second Int'l Conf. Future Generation Comm. and Networking (FGCN '08)*, vol. 2, pp. 162-166, Dec. 2008.

[42] W.-Y. Lin, G.-Y. Lin, and H.-Y. Wei, "Dynamic Auction Mechanism for Cloud Resource Allocation," *Proc. IEEE/ACM 10th Int'l Conf. Cluster, Cloud and Grid Computing (CCGRID '10)*, pp. 591-592, 2010.

[43] N. Nisan and A. Ronen, "Algorithmic Mechanism Design," *Games and Economic Behavior*, vol. 35, nos. 1/2, pp. 166-196, Apr. 2001.

[44] Y.-K. Che, "Design Competition through Multidimensional Auctions," *RAND J. Economics*, vol. 24, no. 4, pp. 668-680, 1993.

[45] F. Branco, "The Design of Multidimensional Auctions," *RAND J. Economics*, vol. 28, no. 1, pp. 63-81, 1997.

[46] M. Bichler and J. Kalagnanam, "Configurable Offers and Winner Determination in Multi-Attribute Auctions," *European J. Operational Research*, vol. 160, pp. 380-394, 2005.

[47] A. Ronen and A. Saberi, "On the Hardness of Optimal Auctions," *Proc. 43rd Symp. Foundations of Computer Science (FOCS '02)*, pp. 396-405, 2002.

[48] T.S. Chandrashekar, Y. Narahari, C.H. Rosa, D.M. Kulkarni, J.D. Tew, and P. Dayama, "Auction-Based Mechanisms for Electronic Procurement," *IEEE Trans. Automation Science and Eng.*, vol. 4, no. 3, pp. 297-321, July 2007.

[49] A. Ronen and D. Lehmann, "Nearly Optimal Multi Attribute Auctions," *Proc. Sixth ACM Conf. Electronic Commerce (EC '05)*, pp. 279-285, 2005.

[50] M. Wang, S. Liu, S. Wang, and K.K. Lai, "A Weighted Product Method for Bidding Strategies in Multi-Attribute Auctions," *J. Systems Science and Complexity*, vol. 23, pp. 194-208, 2010.

[51] L. Zeng, B. Benatallah, A.H.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-Aware Middleware for Web Services Composition," *IEEE Trans. Software Eng.*, vol. 30, no. 5, pp. 311-327, May 2004.

[52] C. Hwang and K. Yoon, *Multiple Attribute Decision Making : Methods and Applications.* Springer-Verlag, 1981.

[53] M. Mohabey, Y. Narahari, S. Mallick, P. Suresh, and S.V. Subrahmanya, "A Combinatorial Procurement Auction for QoS-Aware Web Services Composition," *Proc. IEEE Int'l Conf. Automation Science and Eng. (CASE '07)*, pp. 716-721, 2007.

[54] T. Saaty, *The Analytic Hierarchy Process, Planning, Piority Setting, Resource Allocation.* McGraw-Hill, 1980.

[55] M.C.Y. Tam and V.M.R. Tummala, "An Application of the AHP in Vendor Selection of a Telecommunications System," *Omega*, vol. 29, no. 2, pp. 171-182, 2001.

[56] R.L. Nydick and R.P. Hill, "Using the Analytic Hierarchy Process to Structure the Supplier Selection Procedure," *Int'l J. Purchasing and Materials Management*, vol. 28, no. 2, pp. 31-36, 1992.

[57] M.A. Satterthwaite, "Strategy-Proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions," *J. Economic Theory*, vol. 10, no. 2, pp. 187-217, Apr. 1975.

[58] N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani, *Algorithmic Game Theory.* Cambridge Univ. Press, 2007.

[59] C. d'Aspremont and L.-A. Grard-Varet, "Incentives and Incomplete Information," *J. Public Economics*, vol. 11, no. 1, pp. 25-45, 1979.

[60] R. Gautam, N. Hemachandra, Y. Narahari, H. Prakash, D. Kulkarni, and J. Tew, "Optimal Auctions for Multi-Unit Procurement with Volume Discount Bids," *Int'l J. Operational Research*, vol. 6, no. 1, pp. 70-91, 2009.

[61] B. Katzman, J. Reif, and J.A. Schwartz, "The Relation between Variance and Information Rent in Auctions," *Int'l J. Industrial Organization*, vol. 28, no. 2, pp. 127-130, 2010.

[62] Amazon Web Services LLC, "Amazon EC2 Pricing," http://aws.amazon.com/ec2/pricing/, Jan. 2011.

Abhinandan S. Prasad received the MTech degree in information technology from IIIT-Bangalore, a graduate school of information technology in Bangalore, India. He is currently at Alcatel Lucent India, Bangalore. His research interests include resource allocation and its application in cloud computing, and computational linguistics applied to Sanskrit. He is a member of the IEEE Computer Society and the IEEE.

Shrisha Rao received the MS degree in logic and computation from Carnegie Mellon University and the PhD degree in computer science from the University of Iowa. He is currently an associate professor at IIIT-Bangalore. His research interests are in distributed computing, specifically algorithms and approaches for concurrent and distributed systems, and include solar energy and microgrids, cloud computing, energy-aware computing ("green IT"), and demand side resource management. He is a member of the ACM, the American Mathematical Society, the Computer Society of India, and the IEEE Computer Society. He is a senior member of the IEEE.